

Nº30 diciembre 2016



Comparti MOSS

REVISTA ESPECIALIZADA EN TECNOLOGÍAS MICROSOFT

Entrevista
Yina Arenas

Desarrollo con
SharePoint
Framework

SharePoint
FrameWork: El
Despliegue

Enterprise
Mobility Suite
Parte 1 – Intro-
ducción

Staff

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

DIRECCIÓN GENERAL

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

DISEÑO Y DIAGRAMACIÓN

- Santiago Porras Rodríguez

Contacte con nosotros

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

BLOGS

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

REDES SOCIALES

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

Contenido

03
 Editorial

08
 Big Data como solución PaaS en Azure: HDInsight y HBase

18
 Como realizar una plantilla en Azure en formato ARM

23
 SharePoint y Azure – Azure Functions

33
 Integración continua en SharePoint con Team Services y PnP

39
 Prepara la migración desde tus granjas OnPremises a SharePoint Online con SMAT

44
 Enterprise Mobility Suite Parte 1 – Introducción

47
 Guía práctica sobre límites de uso y tamaños de SharePoint

57
 Integración de SharePoint y Wordpress mediante webhooks, Azure y el API de Wordpress.com (I)

04
 Microsoft Teams: mejorando aún más la colaboración en Office 365

13
 Desarrollo con SharePoint Framework

21
 Entrevista Yina Arenas

27
 Usando R-script en Power BI

37
 Power BI Embedded

42
 SharePoint Framework: El Despliegue

46
 Límite de tamaño y uso de SharePoint - Infografía

52
 DocumentDB Query's



03

Editorial

Para los que llevamos muchos años trabajando con tecnología Microsoft, estamos en un nuevo momento de cambio hacia una nueva estrategia más “Open” y lo podemos ver en las últimas noticias de Microsoft o en las que hemos recibido desde que Satya Nadella ha tomado las riendas del gigante de Redmond.

Este nuevo terreno Open Source es donde las empresas tenemos que aprender a jugar y aprovechar todo el potencial que nos puede ofrecer. Pensemos en Visual Studio for Mac, .NET Core y, como no podía ser menos, el nuevo SharePoint Framework que se apoya en este tipo de tecnología para revolucionar el actual modelo de desarrollo para SharePoint.

Por su puesto, Microsoft no se olvida que el objetivo es que todos hagamos negocios haciendo un mundo mejor y por eso no se ha olvidado de nuestro SharePoint 2016 recibiendo su primer Feature Pack con funcionalidades nuevas que, como ya sabemos, nos vienen de SharePoint Online. Tampoco se olvida de mejorar la productividad de los usuarios de Office 365 creando aplicaciones como Teams o añadiendo nuevas capacidades a Azure para desplegar Bots como servicio.

Los dejamos con este nuevo número de la revista que día a día la siguen alimentando magníficos autores que nos cuentan estas y otras novedades del mundo de la tecnología Microsoft.

El equipo editorial de CompartiMOSS



Microsoft Teams: mejorando aún más la colaboración en Office 365

Nuevas necesidades de colaboración

La colaboración y el trabajo en equipo son dos factores claves en el día a día de gran parte de las empresas. La tecnología nos proporciona herramientas fantásticas, ajustadas a nuestras necesidades, con las que poder colaborar con nuestros compañeros de trabajo y conseguir ser extremadamente productivos.

No obstante, un problema muy importante es el conseguir ser eficiente trabajando con multitud de servicios y aplicaciones, desarrollados por multitud de fabricantes, con diferentes tecnologías y formas de usar. Es por ello que, en los últimos años, han surgido aplicaciones que nos permiten aglutinar en un único punto multitud de servicios y orientarlos al trabajo en grupo.

Cuando Microsoft quiso pagar 8.000 millones de dólares por comprar Slack y, finalmente, se echó atrás argumentando que preferían invertir esa cantidad en potenciar Skype para hacerlo más útil y amigable para el mundo empresarial, comenzamos a sospechar que estaba trabajando en su propia herramienta colaborativa. Durante meses se rumoreó sobre un supuesto “Skype Teams” que permitiría acceder a todos los servicios de Microsoft desde un único punto.

Finalmente, el pasado mes de noviembre se presentaba Microsoft Teams, una herramienta de trabajo en grupo, basada en chat, desde la que se puede acceder no sólo a los servicios de Office 365, sino a otros servicios externos a la plataforma y externos a Microsoft.

¿Qué es Microsoft Teams?

Microsoft Teams llega como un nuevo servicio de Office 365 orientado al trabajo en grupo y a la colaboración. El objetivo de Microsoft Teams es que podamos utilizar todos los servicios de nuestro día a día desde una única aplicación donde el chat, las conversaciones y el trabajo en equipo sean el centro del trabajo.

Si pensamos en los servicios de Office 365, Microsoft Teams nos permite tener en un único punto nuestras conversaciones, videollamadas, archivos del Team, blocs de OneNote, sitios de SharePoint, documentos de Office, planificaciones de Planner y paneles de Power BI. No será necesario ir sal-

tando de servicio en servicio para poder consumir toda la información de cada uno de los servicios.

Adicionalmente, Microsoft Teams permite o va a permitir conectar a muchos otros servicios como Visual Studio Team Services, Zendesk, Asana, Hootsuite, los servicios de Intercom, Jira, Twitter, Wunderlist, Trello, GitHub, etc.

¿Cómo funciona?

Microsoft Teams puede utilizarse desde el navegador integrado en Office 365 (teams.microsoft.com), desde aplicaciones de escritorio (Windows y Mac) y desde apps para teléfonos móviles (Windows, iOS y Android).

La interfaz de trabajo contiene distintas secciones sobre las que trabajar (Imagen 1):

- **Actividad:** en ella veremos notificaciones sobre nuestras interacciones con los compañeros de trabajo, menciones, respuestas y avisos de servicios a los que hayamos conectado.
- **Chat:** podremos establecer conversaciones individuales con usuarios, conversaciones de grupo y conversaciones con bots.
- **Equipos:** el centro neurálgico de la aplicación, en la que podremos crear nuestros equipos y trabajar con ellos en los distintos servicios.
- **Reuniones:** nos permitirá ver las próximas entradas en nuestro calendario y en los calendarios de todos los equipos de los que formamos parte.
- **Archivos:** permite acceder a los archivos de nuestro OneDrive y a los archivos del resto de equipos de los que formamos parte.

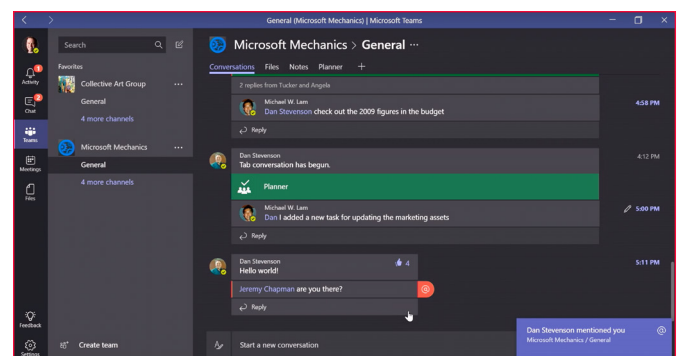


Imagen 1.- El área de trabajo de Microsoft Teams.

La pantalla de Equipos es, sin duda, la zona principal de trabajo de Microsoft Teams (Imagen 2). En ella, los usuarios

podrán crear nuevos equipos de trabajo. Los equipos se pueden dividir en canales, que serán distintos espacios de trabajo dentro de un equipo.

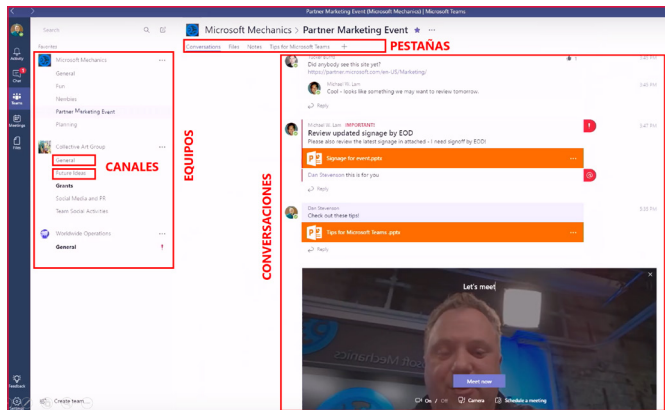


Imagen 2.- Distribución de la pantalla de equipos.

Cada canal tiene un espacio de chat y una serie de pestañas. En el chat se podrá iniciar nuevas conversaciones, responder, hacer likes, programar videoconferencias y comentar sobre contenido interactivo que se añada al canal (como documentos, imágenes, vídeos y contenido generado desde conectores a servicios externos).

Microsoft Teams llega como un nuevo servicio de Office 365 orientado al trabajo en grupo y a la colaboración

Las pestañas nos permiten acceder a contenido adicional del equipo. Por defecto veremos las pestañas de conversaciones, archivos y notas. La pestaña “Archivos” (Imagen 3) nos permite visualizar los archivos del canal, cargar nuevos archivos y editarlos desde Office Online o desde Office cliente. La pestaña “Notas” permitirá acceder, visualizar y editar un bloc de notas compartido para todo el canal.

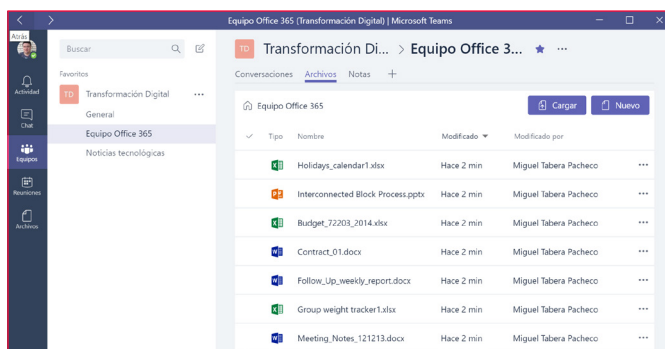


Imagen 3.- Pestaña de archivos del equipo.

Además de las tres pestañas por defecto, es posible insertar nuevas pestañas. Con el tiempo se irán añadiendo nuevas pestañas, de momento se dispone de las siguientes:

- Excel, PowerPoint y Word: permiten anclar un documento importante directamente en una pestaña para que los miembros del equipo tengan acceso a él.
- OneNote: permite anclar blocs de notas adicionales al canal.

- SharePoint: permite visualizar los documentos de un sitio concreto de SharePoint, además de los documentos propios que ya existen en el canal.
- Planner: permite crear planes con tareas del servicio Planner de Office 365, lo que permitirá organizar el trabajo en equipo de una forma más eficiente (Imagen 4).
- Power BI: permite añadir uno o varios informes del servicio Power BI a la pestaña.
- Team Services: sin duda una gran utilidad para desarrolladores. Consiste en un panel de Visual Studio Team Services para que el equipo pueda colaborar en la supervisión y el seguimiento del trabajo.

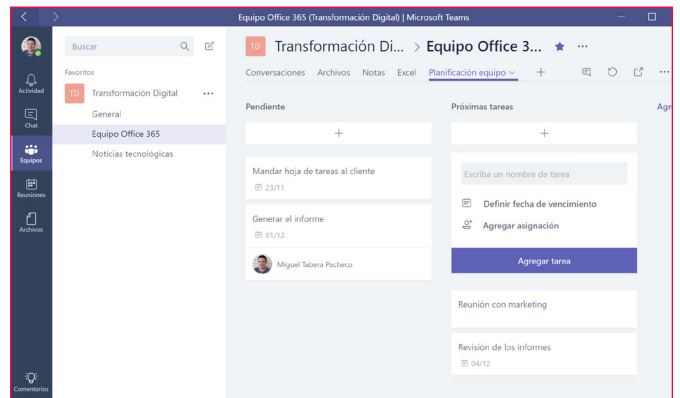


Imagen 4.- Vista de la pestaña de Planner.

Los canales admiten el uso de conectores, que permitirán conectar a multitud de servicios externos como Twitter, Trello, Wunderlist, RSS, Asana, GitHub, Jira, Zendesk, Dynamics 365, MailChimp, BitBucket, Google Analytics, Jenkins, Salesforce, Yammer, etc. Estos conectores se pueden desarrollar y publicar y permiten integrar notificaciones de los distintos servicios dentro del área de conversaciones para que el equipo pueda trabajar con esta información. La lista cada vez es más grande (Imagen 5).

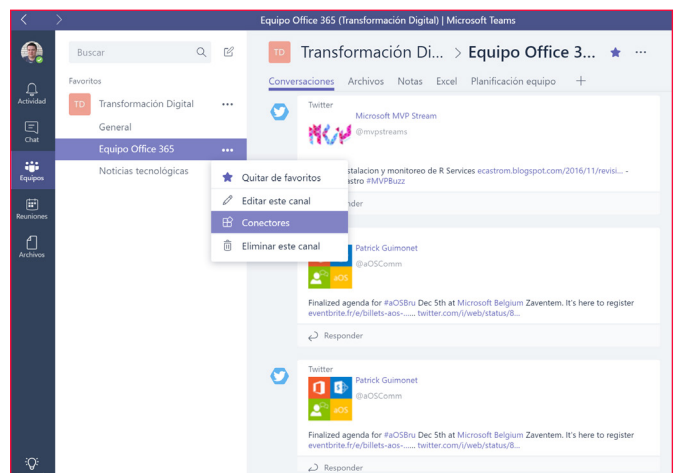


Imagen 5.- Cómo añadir conectores y ejemplo de publicaciones del conector de Twitter.

Los chats están adaptados a las necesidades que tienen los equipos actualmente. Se organizan por hilos de conversación y permiten insertar textos enriquecidos, contenido dinámico, Emojis y GIFs, texto resaltado y adjuntar imágenes y archivos. Es posible participar en los chats del equipo y sus canales o acceder a la sección “Chats” en la que se puede hacer chats privados con una persona o con varias.

Además de chats con personas, es posible hacer chats con

bots inteligentes. Los desarrolladores podrán programar y publicar sus propios bots en Microsoft Teams. Para probar el chat con bots, es posible, de momento, iniciar una conversación con T-Bot, el chat de ayuda de Microsoft Teams (Imagen 6). T-Bot contiene opciones de conversación y pestañas con ayuda, vídeos y notas adicionales.

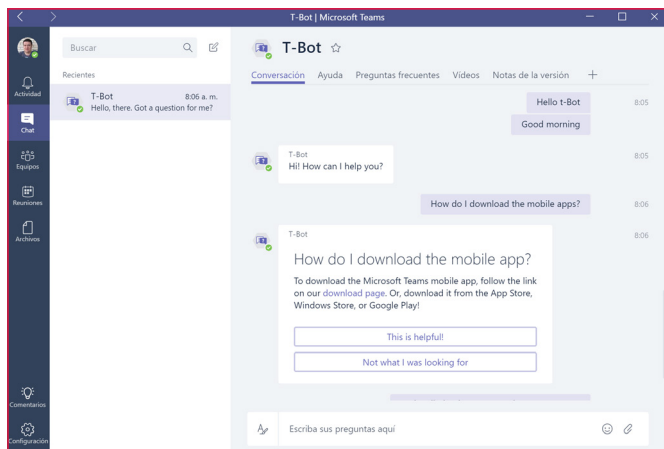


Imagen 6.- El bot T-Bot nos dará ayuda y consejos sobre el uso del servicio.

También es posible realizar videollamadas a varias personas. Las videollamadas se pueden iniciar desde las conversaciones del canal y los demás usuarios podrán verlas y unirse al momento. También se puede añadir manualmente a nuevos miembros y se podrá ver a la vez el vídeo de hasta cuatro conversaciones (Imagen 7). Cabe destacar que, una vez finalizada, la videollamada quedará grabada en el canal y los usuarios podrán verla y comentar sobre ella en el chat del canal.

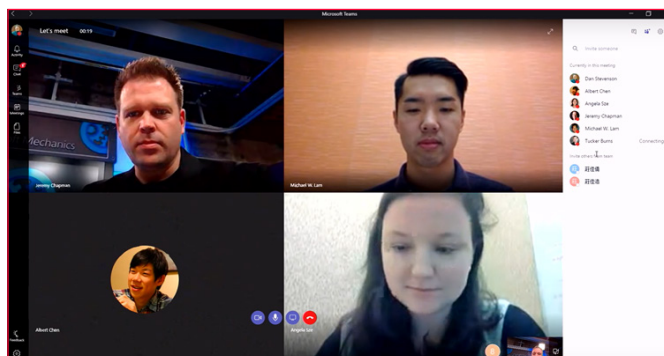


Imagen 7.- Videollamadas grupales en Microsoft Teams.

Microsoft Teams vs Office Groups

Cuando uno comience a utilizar Microsoft Teams, seguramente se preguntará qué va a pasar con los Grupos de Office 365 ya que muchas funcionalidades son propias de los grupos (conversaciones, archivos, OneNote, sitio de SharePoint...).

En realidad, los equipos de Microsoft Teams y los grupos de Office 365 están mucho más unidos de lo que puede parecer, ya que Microsoft Teams se basa internamente en los grupos de Office 365. Es bueno saber que cada equipo de Microsoft Teams es un grupo de Office 365.

Cuando creamos un nuevo equipo en Microsoft Teams, se aprovisionará internamente un grupo de Office 365 con el mismo nombre, descripción y miembros. Varios de los ser-

vicios que ofrece Microsoft Teams utilizan servicios de los grupos. Algunos ejemplos interesantes son:

- Archivos: los archivos de un equipo de Microsoft Teams se almacenan en la biblioteca del sitio del Grupo de Office 365. En la biblioteca principal del sitio se creará una carpeta por cada canal del equipo. En Microsoft Teams, veremos en cada canal el contenido de la carpeta.
- OneNote: los espacios de notas de un equipo de Microsoft Teams se almacenan en el bloc de notas OneNote del grupo de Office 365. En el OneNote se creará una sección por cada canal del equipo. En Microsoft Teams, veremos en cada canal el contenido de su sección.
- Planner: el contenido de Planner que podemos añadir al equipo en Microsoft Teams serán planes del grupo de Office 365. A diferencia que el uso clásico de Planner (1 grupo = 1 plan), en Microsoft Teams se nos permite añadir distintos planes por canal (1 grupo = múltiples planes).

Aunque el tener grupos y equipos y que ambas cosas puedan ser lo mismo puede prestar a confusión, puede ser muy útil a la hora de trabajar. En Microsoft Teams tendremos todo el contenido en un único lugar, lo que es práctico para trabajar de forma mucho más efectiva. En cambio, cuando nos movamos por los distintos servicios de Office 365, tendremos mucho más a mano los grupos para trabajar con nuestro contenido y hacer cosas como sincronizar los archivos con el cliente de OneDrive, conectar blocs de notas a OneNote de escritorio o acceder rápidamente a archivos.

Opciones de administración

Los administradores de IT de Office 365 podrán activar la preview de Microsoft Teams y controlar el uso de determinadas características del servicio. Aunque las opciones son limitadas, se irá disponiendo de más control a medida que el servicio evolucione y salga de la preview.

De momento, es posible hacer las siguientes tareas (Imagen 8):

- Activar/desactivar el servicio.
- Controlar determinados aspectos que se muestran en el perfil.
- Permitir notificaciones por correo.
- Activar el envío de información de diagnóstico a Microsoft.
- Permitir vídeo y compartir pantalla en las conversaciones.
- Permitir el uso de imágenes animadas informales en las conversaciones.
- Asignar un nivel de moderación del contenido que se publica.
- Añadir imágenes editables en las conversaciones.
- Activar/desactivar la posibilidad de añadir nuevas pestañas en los equipos.
- Activar/desactivar el uso de bots en el servicio.

Adicionalmente, como un equipo es un grupo de Office 365, se dispone de las mismas opciones de administración y control de grupos para un equipo.

También es bueno que los administradores de IT conozcan que Microsoft Teams cumple con las siguientes características de seguridad:

- Los datos están cifrados tanto en reposo como en tránsito.
- Se cumple con todas las características de cumplimiento que el resto de servicios (EUMC, HIPAA, etc.).
- Se soporta la autenticación multi-factor, al igual que el resto de Office 365.
- El servicio se sirve desde los mismos centros de datos de Office 365 que el resto de la suscripción.
- La administración está y estará centralizada en Office 365 y se irá ampliando y dotando de más opciones de seguridad, control y administración.

Posibilidades de desarrollo

Actualmente, se ha abierto un programa “Developer Preview” para que los desarrolladores puedan ampliar el servicio de Microsoft Teams con su propio contenido o contenido de otros servicios. Microsoft ha invitado a desarrolladores de todo el mundo a unirse a este programa y comenzar a desarrollar contenido adicional para el servicio mediante la creación de pestañas, bots y conectores.

Por un lado, es posible crear pestañas para Microsoft Teams en las que los usuarios podrán visualizar una web o servicio desde su equipo o canal y podrán trabajar con la información en el contexto adecuado.

El desarrollador tendrá un lienzo sobre el que exponer su servicio, sobre el que se podrá interactuar. Este servicio se podrá añadir en un canal como una nueva pestaña. Para más información sobre desarrollo y publicación de pestañas, se puede consultar <https://msdn.microsoft.com/en-us/microsoft-teams/tabs>

Servicios como Asana ya están creando sus propias pestañas y muestran sus ejemplos en Office Dev Center (Img 8)

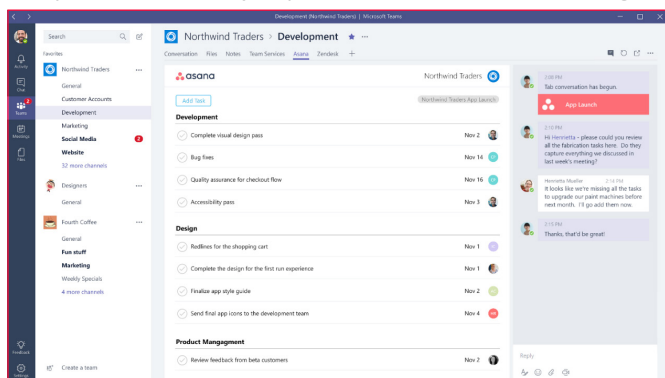


Imagen 8.- Ejemplo de diseño de una pestaña personalizada para Asana.

También es posible escribir bots que puedan interactuar con usuarios y equipos proporcionándoles ayuda, servicios y acciones. Estos bots se podrán desarrollar mediante el

Microsoft Bot Framework, también usado para desarrollar bots en otras plataformas de Microsoft como Skype o el correo de Office 365. Los bots se pueden usar para dar consejos a los miembros del equipo, contestar con lenguaje natural a determinadas preguntas o a gestionar tareas y planificaciones. También se pueden usar para retirar información inteligente del Office Graph y mostrarla a los usuarios.

La colaboración y el trabajo en equipo son dos factores claves en el día a día de gran parte de las empresas

Para más información sobre desarrollo y publicación de bots, se puede consultar <https://msdn.microsoft.com/en-us/microsoft-teams/bots>

Y si ya tienes un bot creado y publicado para servicios como Slack, es posible migrarlo con poco esfuerzo a Microsoft Teams gracias a Message.io.

Finalmente, es posible crear conectores para enviar notificaciones de nuestros propios servicios a los canales de los equipos y publicarlos como conversaciones. Si quieres ver ejemplos de cómo funcionan los conectores, puedes añadir a un canal alguno de los conectores ya existentes (Twitter, Trello, Wunderlist, RSS, Asana, GitHub, Jira, Zendesk, Dynamics 365, Google Analytics, Salesforce, Yammer, etc.).

Para más información sobre desarrollo y publicación de conectores, se puede consultar <https://msdn.microsoft.com/en-us/microsoft-teams/connectors>

Y si quieres más información sobre el programa Developer Preview de Microsoft Teams, puedes visitar <https://dev.office.com/microsoft-teams>

Comenzar a probar Microsoft Teams

Microsoft Teams ya es una realidad y cualquier cliente empresarial de Office 365 puede comenzar a usar la preview. Para ello, un administrador debe activar el servicio desde el Centro de Administración de Office 365 (en la sección Configuración -> Aplicaciones).

Hasta que aparezca un icono oficial en el lanzador de aplicaciones de Office 365, se puede utilizar accediendo a teams.microsoft.com o descargando las aplicaciones de escritorio (Windows y Mac) o móviles (Windows, iOS y Android) desde teams.microsoft.com/downloads

MIGUEL TABERA PACHECO

Office 365 Lead en NECSIA

MVP de Office Servers and Services

miguel.tabera@outlook.com

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

<http://docs.com/migueltabera>

Big Data como solución PaaS en Azure: HDInsight y HBase

Es una evidencia que cada vez más tendemos a acumular cantidades muy grandes de datos, y que casi cualquier corporación media está pasando por este proceso. Si bien es muy normal que casi cualquier empresa haya abandonado ya los modelos relacionales, invirtiendo en otros modelos más dinámicos como modelos documentales o analíticos, se hace más complejo ver instaurado un modelo Big Data, bien por desconocimiento, por coste o por cultura.

¿Y si este escenario se convirtiere en un servicio de pago por uso?

¿Qué es HDInsight?

Aunque no es el objetivo de este artículo, si sería bueno tener nociones básicas de que es HDInsight y sobretodo adentrarnos un poco en la solución Hadoop que nos proporciona este servicio.

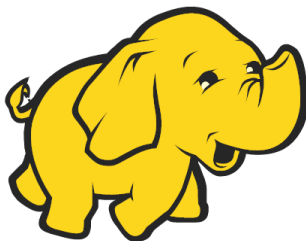


Imagen 1.- Logo HDInsight.

HDInsight no deja de ser un servicio PaaS que nos proporciona Microsoft desde Azure para instalar soluciones de Apache Hadoop que nos permiten entre otras cosas disfrutar de Spark, Hive, MapReduce, R o HBase entre otras utilidades.

HDInsight no deja de ser un servicio PaaS que nos proporciona Microsoft desde Azure

Para el que nunca haya trabajado con Hadoop el párrafo anterior le puede haber sonado a chino como es lógico, y siento anticipar que en este artículo no voy a profundizar en esta herramienta, aunque sí puedo referenciar que Apache Hadoop es un proyecto Open Source que se define como un Framework que permite el procesamiento de grandes cantidades de datos en sistemas distribuidos. O lo que es lo mismo está pensado para en base a un sistema distribuido en un cluster (pensado para un crecimiento horizontal y vertical), podamos manejar cantidades eleva-

das de datos, por lo que podemos empezar a pensar en Gigabytes o TeraBytes de información.



Imagen 2.- Logo HBase.

Dentro de toda esta gran herramienta nos vamos a centrar en como instalar y manejar un cluster con el objetivo de utilizar HBase que por definición es una base de datos distribuida no relacional modelada a partir de Google BigTable y escrita en Java.

HBase: datos orientados a columnas

Apache HBase no es más que una base de datos NoSQL de código abierto que encontramos dentro del proyecto Hadoop y que está pensado para almacenar gran cantidad de datos no estructurados y semiestructurados. HBase es un ejemplo de base de datos que tiene una organización de datos Orientado a Columnas. Esto quiere decir que almacena la información por columnas o familia de columnas permitiendo almacenar una gran cantidad de datos ya que la búsqueda de un valor de una columna en concreto evita tener que ir recorriendo cada una de las filas.

Por ejemplo, para guardar la información de un usuario que podría ser de la siguiente forma:

ID	NOMBRE	APELLIDOS	TELÉFONO	CARGO	CENTRO DE TRABAJO	RANGO SALARIAL
10000	Sergio	Hernández		Team Leader	Madrid	A1

En HBase dejaríamos a un lado este modelo tabular y tendría una forma parecida a la siguiente:

FAMILIA DE COLUMNAS: PERSONAL				FAMILIA DE COLUMNAS: PROFESIONAL		
ID	NOMBRE	APELLIDOS	TELÉFONO	CARGO	CENTRO DE TRABAJO	RANGO SALARIAL
10000	Sergio	Hernández		Team Leader	Madrid	A1

HBase está optimizado para que crezca el número de columnas frente el crecimiento por filas, pudiendo agrupar

estas columnas por familias agrupando así la información que necesitemos filtrar.

Cada fila es única y se asignará una única RowKey por lo que toda la información de un usuario se almacenará en una única fila. Este modelo de datos está pensado para escenarios con una cantidad de datos elevada en escenarios como blogs, foros o logging.

Primer Paso, crear nuestro cluster de Hadoop

Para empezar a trabajar con Hadoop o en nuestro caso con HBase lo primero que tendremos que hacer es acceder al portal de Azure y crear un servicio de HDInsight.

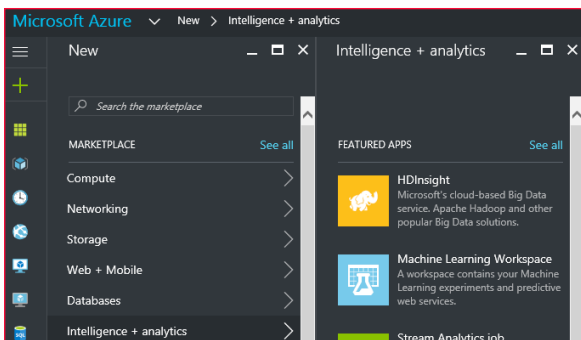


Imagen 3.- Creando el servicio.

Lo vamos a encontrar en la categoría de "Intelligence & analytics" y para configurarlo correctamente seguiremos los siguientes pasos:

- Nombre del cluster. Lo primero es dar un nombre "reconocible" y que recordemos a nuestro nuevo cluster. Es importante recordar este paso ya que lo vamos a utilizar a futuro para conectarnos. El nombre del ejemplo será CompartimossCluster.azurehdinsight.net .
- Configuración del cluster. Este paso posiblemente sea uno de los más importantes ya que estamos dándole forma a nuestro cluster, deberemos decidir el tipo de cluster (Hadoop, Hbase, R Server), si lo queremos en base Linux o Windows y la versión del Hbase. Este paso es importante ya que no podremos modificarlo una vez creado.

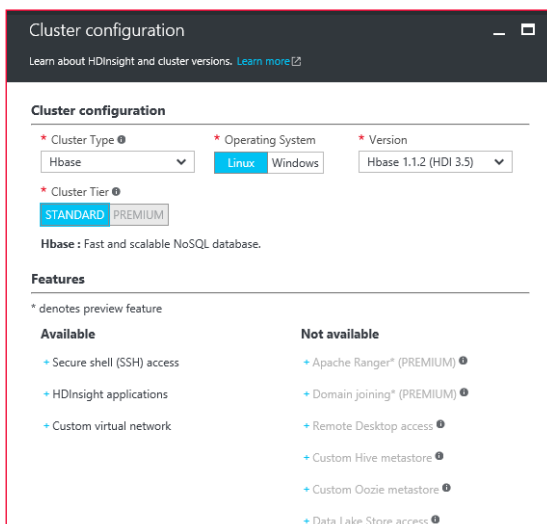


Imagen 4.- Configuración cluster.

- Credenciales de acceso. Este paso nos permitirá configurar tanto las credenciales de acceso a nuestro Cluster y por tanto al portal del cluster, como el acceso remoto por SSH (Secure Shell) como veremos más adelante.

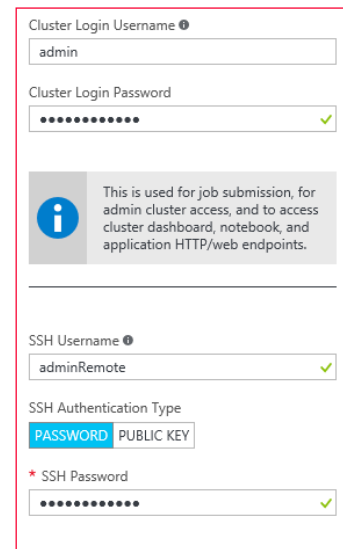


Imagen 5.- Credenciales cluster.

- Origen de datos. Podremos crear una nueva cuenta de almacenamiento donde se dejarán nuestros datos, o por el contrario utilizar una ya existente en nuestra suscripción.
- Plan de precios. Posiblemente el paso más importante ya que nos va a marcar por un lado la potencia de nuestra instalación, y por otro cuando nos vamos a gastar (importante en un modelo de pago por uso). Aclarar que una instalación de Hadoop o HBase es costosa ya que implica la instalación de un nº elevado de máquinas. Deberemos escoger el nº de regiones siendo el mínimo una región que implicará la creación de 6 máquinas para distribuir nuestra instalación. Cada máquina es configurable pudiendo añadir "Cores", o discos sólidos. En nuestro caso vamos a usar el plan mínimo tanto de regiones como de máquinas, y podréis observar que el coste ya es de 1,47€ por hora.

Number of Region nodes	1	✓
* Region node size	>	
D3 v2 (1 node, 4 cores)		
* Head node size	>	
D3 v2 (2 nodes, 8 cores)		
* Zookeeper node sizes	>	
A3 (3 nodes, 12 cores)		
REGION NODES	0.29 x 1 =	0.29
HEAD NODES	0.29 x 2 =	0.58
ZOOKEEPER NODES	0.20 x 3 =	0.61
TOTAL COST		1.47
		EUR/HOUR (ESTIMATED)

Imagen 6.- Plan de precios.

Es cierto que la instalación requiere de más indicaciones ya que se nos permite añadir aplicaciones extra o elegir el grupo de recursos. Una vez que terminemos este proceso, la creación es automática, pero se deben saber dos

cosas. La primera es que es relativamente lenta dado que el caso mínimo es aprovisionar 6 máquinas en un cluster, y por otro lado a día de hoy no se puede detener el cluster, por lo que se debe borrar si no queremos gastar más de lo debido ya que el pago es por uso y hora de encendido.

Diseña y trabaja con bases de datos sobre HBase

Hasta aquí ya podemos decir que tenemos una idea general de que es HBase, de que es HDInsight y tenemos un cluster distribuido de datos listo para empezar a trabajar con él.

¿Pero cómo empiezo a trabajar con mi HBase?

Para empezar, deberemos diseñar un modelo de datos ajustado a columnas como vimos en la explicación de HBase, y en nuestro caso usaremos el ejemplo de los datos de un Usuario que vimos en el primer ejemplo.

HBase Shell con Secure Shell (SSH)

La primera forma que vamos a ver para comunicarnos con HBASE es hacer uso de HBase Shell y para ello lo primero es conectarnos al cluster vía SSH, para poder aplicar de forma remota las distintas sentencias que necesitamos. Para comunicarnos vía SSH vamos hacer uso de una herramienta gratuita como Putty, que podemos descargar desde su página oficial <http://www.putty.org/>.

Una vez descargada la aplicación, necesitaremos configurar la conexión contra nuestro cluster de HDInsight. El nombre de la conexión es fácil de deducir ya que es <Nombre_Cluster-ssh.azurehdinsight.net>.

Ahora debemos recuperar las credenciales que escogimos en la instalación del cluster y en concreto el nombre de usuario SSH y contraseña que escogimos.

Si no recordamos el nombre de que escogimos tenemos todos estos datos en el portal de Azure y en concreto en la configuración de nuestro servicio de HDInsight.

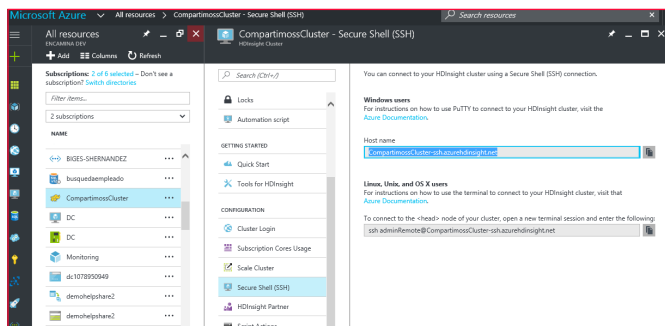


Imagen 7.- Configuración del servicio.

No es necesario cambiar el puerto que por defecto viene configurado en el 22. Seleccionaremos en tipo de conexión, la opción SSH.

Siguiendo todos estos pasos deberíamos tener todo configurado, y bastaría con seleccionar “Open” para que de forma casi instantánea se nos pidan las credenciales de SSH que configuramos.

Si todo ha ido de forma correcta deberíamos tener acceso a la consola tal cual se ve en la siguiente imagen.

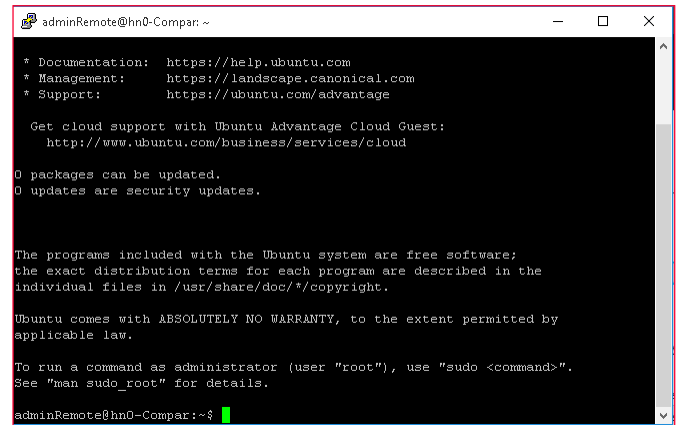


Imagen 8.- Empezando con hbase Shell.

Una vez aquí debemos instanciar HBase Shell con la instrucción “hbase Shell”, y desde aquí ya podemos hacer uso de las distintas instrucciones que encontramos en este lenguaje, con el que podemos hacer cualquier operación que necesitemos.

Crear Tabla Usuario

Deberemos ejecutar la instrucción “create ‘Nombre Tabla,’Nombre Familia Columnas,’Nombre Familia de Columnas”.

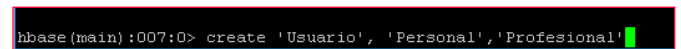


Imagen 9.- Creando una tabla con Shell.

Introducir usuario y definir columnas

Una vez tenemos definida la tabla vamos a introducir un primer dato de prueba para definir el contenido de la familia de columnas y las columnas con las que vamos a organizar la información. Para ir añadiendo a las columnas ejecutaremos el comando “put ‘tabla’, ‘rowKey’ familia:columna’, ‘valor’ ”



Imagen 10.- Definiendo el contenido.

Deberemos ir ejecutando esta instrucción por cada una de las columnas que queremos ir definiendo. Decir que el rowKey es único, no pueden existir dos filas con un mismo RowKey en HBase.

Recuperar los datos de una tabla

Tenemos dos instrucciones para ver el contenido o la estructura de una tabla. La primera es “describe ‘tabla””, que nos expone el contenido y la estructura completa de una

tabla. Y por otro lado podemos ejecutar tanto la instrucción “get ‘tabla’, ‘rowKey’”, que nos devuelve el contenido de una fila por clave de fila o bien ejecutar una instrucción “scan ‘tabla’, ‘{familia=>[columnas]}”.

```
hbase(main):023:0> get 'Usuario','1'
COLUMN                                CELL
Personal:Apellidos                    timestamp=1479677788704, value=Hernandez
Personal:Nombre                        timestamp=1479677528736, value=Sergio
2 row(s) in 0.0210 seconds
```

Imagen 11.- Resultado instrucción Get.

```
hbase(main):031:0> scan 'Usuario',{'Personal'=>['Apellidos','Nombre']}
ROW                                    COLUMN+CELL
1                                       column=Personal:Apellidos, timestamp=1479677788704, value=Hernandez
1                                       column=Personal:Nombre, timestamp=1479677528736, value=Sergio
1 row(s) in 0.0080 seconds
```

Imagen 12.- Resultado de un Scan.

La diferencia de una consulta u otra, es que hacer un get por clave de fila recupera toda la información de una fila, y realizar un scan nos permite no filtrar por fila sino por “columna”, sacando el máximo partido al modelo orientado a columnas que nos aporta HBase.

C# y HBase

Aunque Hadoop está desarrollado sobre Java, nos es posible comunicarnos y hacer todas las operaciones que hemos visto con HBase Shell desde código en C#. Para ello debemos hacer uso de un paquete NuGet que podemos descargar desde el Visual Studio y añadirlo a nuestras soluciones, y que se llama Microsoft HBase Client.

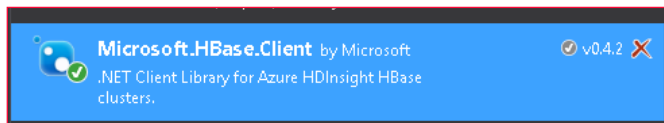


Imagen 13.- Paquete Hbase.Client.

Para poder manejar esta librería necesitamos manejar los siguientes conceptos, tal y como vemos en los siguientes fragmentos de código.

Inicializar la conexión:

```
private void InitServer()
{
    cluster = new ClusterCredentials(new Uri(server), user, pass);
    client = new HBaseClient(cluster);
}
```

Imagen 14.- Code Inicializar cluster.

Lo primer que necesitamos es por un lado instanciar “ClusterCredentials”, para poder instanciar una nueva instancia de HBaseClient que nos dará acceso a la base de datos.

Crear Tabla:

```
public async void CreateTable(string tableName, List<string> cellFamily)
{
    if (cluster == null || client == null)
    {
        InitServer();
    }
    var schema = new TableSchema();
    schema.name = tableName;
    cellFamily.ForEach(i => schema.columns.Add(new ColumnSchema() { name = i }));
    await client.CreateTableAsync(schema);
}
```

Imagen 15.- Código crear tabla.

Si revisamos este código podemos ver como se parece bastante a como lo creamos vía Shell, ya que debemos por un

lado especificar el nombre de la tabla que vamos a crear, como un listado de familia de columnas. Con esto conseguimos definir el schema de la tabla e invocar la creación de la misma. Un ejemplo de uso de este método puede ser CreateTable(«Usuario», new List<string>() {«Profesional»,«Personal»}).

HBase es un ejemplo de base de datos que tiene una organización de datos orientado a columnas

Get fila:

```
public async Task<List<Cell>> GetData(string tableName, string key)
{
    if (cluster == null || client == null)
    {
        InitServer();
    }
    var queryResult = await client.GetCellsAsync(tableName, key);
    var value = queryResult.rows.FirstOrDefault().values;
    return value;
}
```

Imagen 16 Código GetData.

Como vemos la librería de HBase.Client nos expone un método GetCellsAsync, que, especificando el nombre de la tabla y la clave de fila, nos devuelve el contenido de toda la fila, de forma muy similar a como vimos en el método Get de Hbase Shell.

Insertar:

```
public async void InsertData(string tableName, CellSet cellSet)
{
    if (cluster == null || client == null)
    {
        InitServer();
    }
    var result = client.StoreCellsAsync(tableName, cellSet);
    await result;
}
```

Imagen 17.- Código InsertData.

Este método es muy sencillo de implementar, ya que basta con instanciar el método StoreCellAsync que nos proporciona la libería.

Lo interesante es construir “los datos” que queremos insertar dentro de HBase y que son del tipo CellSet. Para ello podemos utilizar el siguiente método que construye un CellSet en base a un listado de valores, una lista de columnas y la clave de fila.

```
private CellSet GeneratedColumns(List<string> columns, List<string> values, string rowkey)
{
    var set = new CellSet();
    var row = new CellSet.Row { key = Encoding.UTF8.GetBytes(rowkey)};
    for (int i = 0; i < columns.Count; i++)
    {
        var cell = new Cell { column = Encoding.UTF8.GetBytes(columns[i]), data = Encoding.UTF8.GetBytes(values[i]) };
        row.values.Add(cell);
    }
    set.rows.Add(row);
    return set;
}
```

Imagen 18.- Código construir columnas.

Si analizamos el código debemos setear CellSet.Row con la clave escogida, y por otro lado ir construyendo el objeto CellSet, añadiendo a la fila cada una de las columnas que hemos ido definiendo. Una forma de generar nuestras columnas sería muy parecida a esta List<string> columnaUsuario = new List<string>() { «personal:Nombre», «personal:Apellidos», «profesional:Sede», «profesional:An-



tiguedad}}, y que deberemos pasar a nuestro método GeneratedColumns.

Scanner:

```
public async Task<List<Cell>> GetDataByScanner(int firstKey, string tableName, string columnFamilyFilter, string keyColumn)
{
    if (cluster == null || client == null)
    {
        InitServer();
    }
    var comparer = new SubStringComparator(firstKey.ToString());
    SingleColumnValueFilter filtro = new SingleColumnValueFilter(Encoding.UTF8.GetBytes(columnFamilyFilter),
        Encoding.UTF8.GetBytes(keyColumn), CompareFilter.CompareOp.Equal, comparer);
    var scanSettings = new Scanner()
    {
        filter = filtro.ToEncodedString(),
    };
    RequestOptions options = RequestOptions.DefaultOptions();
    options.AlternativeEndpoint = "hbase-rest/";
    ScannerInformation info;
    var scannerExecute = client.CreateScannerAsync(tableName, scanSettings, options);
    await scannerExecute;
    info = scannerExecute.Result;
    var task = client.ScannerGetNextAsync(info, options);
    List<Cell> data = new List<Cell>();
    await task;
    foreach (var row in task.Result.rows)
    {
        data.AddRange(row.values);
    }
    return data;
}
```

Imagen 19.- Código Scanner

Posiblemente este sea el método más complejo de instanciar de esta librería, ya que se nos permite hacer infinidad de filtros.

Lo primero que necesitamos es definir el filtro o filtros que vamos a aplicar, y en este caso queremos filtrar por nombre de la familia de columnas.

```
SingleColumnValueFilter filtro = new SingleColumnValueFilter(
    Encoding.UTF8.GetBytes(columnFamilyFilter),
    Encoding.UTF8.GetBytes(keyColumn), CompareFilter.CompareOp.Equal, comparer);
```

Una vez se tiene el filtro definido se puede crear el scanner que ejecutaremos para obtener los datos.

```
var scanSettings = new Scanner()
{
    filter = filtro.ToEncodedString(),
};
```

Una vez definido el Scanner, basta con ejecutarlo con la instrucción ScannerGetNextAsync, que devuelve una lista de filas que cumplen los criterios. Si observamos bien los

métodos que se han implementado son asíncronos, y deberemos instanciarlos de esta forma u obtendremos error al ejecutarlos.

Conclusiones

Una vez más los servicios PaaS de Azure nos demuestran que si no lo es ya, este modelo de consumo por uso va a ser muy útil en el futuro. Por ejemplo, lanzarnos a usar Hadoop sin saber si es nuestra solución indicada, o aun sabiéndolo implementar toda esta infraestructura es muy costoso, cuando en pocos pasos podemos crear un servicio de HDInsight y empezar a trabajar.

Por otro lado, usar HBase es muy recomendable para casos en los cuales vayamos a tener infinidad de datos, y podamos tener información organizada en columnas, sin duda el rendimiento de esta base de datos es muy superior a otras como puede ser una base documental en MongoDB. Eso sí tenemos que analizar bien nuestro modelo de datos, ya que en algunas ocasiones no es sencillo adaptarnos a un modelo orientado a columnas, o simplemente no nos va a ser útil.

Sin duda Hadoop es una herramienta enorme, que podemos empezar a saborear con HDInsight, y que nos permitirá ir introduciendo el Big Data en nuestro día a día sin suponer una gran inversión de inicio tanto en conocimiento como en costes, sin duda muy recomendable empezar a darle uso.

SERGIO HERNÁNDEZ MANCEBO

Team Leader de Soluciones Microsoft en Encamina
@shmancebo

feed your brain®

SharePoint 2016

de principio a fin

Gustavo Velez
Juan Carlos González
Fabián Imaz
Alberto Díaz Martín

SharePoint Server 2016 y la versión Online disponible en Office 365, constituyen la plataforma de colaboración, comunicación y productividad de Microsoft.

Para SharePoint va mucho más allá: es, sobre todo, una plataforma de desarrollo debido a sus Modelos de Objetos abiertos, que permiten extender el sistema según los requerimientos explícitos de los usuarios.

Este libro no sólo abarca todos los aspectos de SharePoint relativos a su instalación, configuración y administración, hasta cómo programar sus infraestructuras sino también su utilización por usuarios casuales y avanzados, y el uso de los herramientas proporcionados por Microsoft para extenderlo y modificarlo: SharePoint Designer, Visual Studio, Office y PowerShell.

Como no podía ser de esta forma, en esta nueva revisión de la obra de referencia en castellano de SharePoint, se ha adaptado por completo todo el contenido a la última versión además de añadir todas las novedades incorporadas por Microsoft en 2016, entre otras:

- La nueva arquitectura para despliegue SharePoint basada en Minikole
- Los nuevos escenarios de instalación disponibles
- Los cambios en la interfaz de usuario
- Las novedades clave de la plataforma en Inteligencia de Negocios con la integración de Office Online Server
- El cumplimiento de Directivas de Prevención de Datos

En esta obra imprescindible encontrarás todo lo necesario para dominar SharePoint Server 2016, sea cual sea tu rol en la empresa y las necesidades que tengas.

Nivel: Iniciación - Intermedia - Avanzado

Desarrollo con SharePoint Framework

En el pasado tuvimos “SharePoint Solutions”, para aquellos de nosotros que somos lo suficientemente mayores y comenzamos con WSS 3.0, SharePoint 2003 o 2007, esta era la única manera de desarrollar soluciones en ese momento. Tuvimos que crear nuestras soluciones con código C #, código que crea páginas, WebParts, listas, Features, Timer Jobs, etc. Hasta el año 2012 fue la manera más popular para desarrollar en SharePoint, pero después de SharePoint 2013 Microsoft comenzó a cambiar su visión, Office 365 es un producto muy importante en la visión de Microsoft, en ese momento se introdujo los SharePoint Add-Ins (anteriormente conocido como SharePoint Apps), con este modelo de desarrollo podemos crear Add-In con C #, pero el código estaba alojado en otro servidor, IIS o Azure, con las API CSOM que puede interactuar con datos de SharePoint, e integrar SharePoint con miles de otras API #. Con SharePoint Hosted Add-Ins, sólo podemos desarrollar con javascript, el código se ejecuta en el navegador, pero con las API REST también podemos interactuar con los datos de SharePoint y combinar varios plugins javascript para llegar a soluciones bien interesantes.

Estas 2 últimas piezas están todavía en funcionamiento, y es probable que sobrevivan durante mucho tiempo, son muy potentes y probablemente tendrán espacio en muchos blogs durante los próximos años. Sin embargo desde mayo de 2016, Microsoft ha anunciado la introducción del SharePoint Framework, también conocido como SPF, este nuevo framework y cito es un:

Modelo de desarrollo de WebParts que permite y soporta plenamente el desarrollo del lado del cliente, fácil integración con el Microsoft Graph y soporte para herramientas de código abierto.

Esta nuevo Framework es espectacular, es de código abierto, por lo que cualquiera de ustedes puede colaborar con el repositorio GitHub, y una de las cosas que me gustan de él, que Microsoft sí lo están utilizando para construir las nuevas experiencias de SharePoint, por ejemplo, el look and feel de la nueva Biblioteca de documentos, entre otras cosas que se han dado a conocer los últimos meses.

Con cada nueva tecnología, llega un reto, un desafío para aprender algo nuevo, debido a que SPF esta basado en las últimas tecnologías de código abierto voy a mencionar lo que se necesita para aprender:

- Obligatorio:
 - TypeScript.
 - NodeJS.
 - NPM (gesto de paquetes como nuget).
 - Gulp.
- No es obligatorio :
 - React.
 - Angular
 - Cualquier otro framework de JS.

A pesar de que menciono que no son obligatorios, hoy en día es importante tener el conocimiento en alguno de estos para entregar mejores soluciones. En este post vamos a aprender como configurar el ambiente para el desarrollo de WebParts con SharePoint Framework.

Cómo empezar

PASO 1:

1.- Obtener un developer tenant de Office 365

Con el fin de construir y desplegar elementos Web en SharePoint durante la vista previa inicial, se necesita un Tenant de Desarrollo Office 365. Tenga en cuenta que, por ahora, el tenant tiene que ser un tenant desarrollador, no sólo la colección de sitios.

Si está recibiendo siguiente excepción cuando se mueve a workbench.aspx página, significa que no está utilizando el tenant de desarrollo: “La operación solicitada es parte de una característica experimental que no se admite en el entorno actual . la operación solicitada es parte de una característica experimental que no se admite en el entorno actual

2.- Crear un sitio catálogo de aplicaciones.

Si usted es como yo y ya ha utilizado el modelo de complementos, entonces usted ya debe saber cómo crear una, si no seguir leyendo:

- Navegar a SharePoint Centro de administración escribiendo el siguiente URL en su navegador (Sustituir yourtenantprefix con su prefijo de Office 36)

<https://yourtenantprefix-admin.SharePoint.com>

- En la barra lateral izquierda, haga clic en el elemento de menú de aplicaciones y haga clic en App

Catalog.

- Haga clic en el botón Aceptar para crear un nuevo sitio catálogo de aplicaciones.
- En la página siguiente, introduzca los siguientes datos:

Título: Aplicación catálogo

Sitio Web Dirección sufijo: Llene sufijo preferido para catálogo de aplicaciones, por ejemplo: aplicaciones

Administrador: Introduzca su nombre de usuario y haga clic en el botón de decisión para resolver el nombre de usuario

- Haga clic en el botón OK para crear el sitio catálogo de aplicaciones. SharePoint en este momento va a crear el sitio catálogo de aplicaciones y usted será capaz de ver su progreso en el centro de administración de SharePoint

- 3.- Crear una nueva colección de sitios desarrollador.
- Navegar a SharePoint Centro de administración escribiendo el siguiente URL en su navegador.

Sustituir yourtenantprefix con su prefijo de Office 365:

<https://yourtenantprefix-admin.SharePoint.com>

- En la cinta de SharePoint, haga clic nuevo-> Colección privada de sitios.
- En el cuadro de diálogo, introduzca lo siguiente:

Título: escriba un título para su colección de sitio de desarrolladores, por ejemplo: desarrollador del sitio

Sitio Web Dirección sufijo común: Introduzca un sufijo para la colección de sitios desarrollador, por ejemplo: dev

Selección de plantilla: Seleccione Sitio desarrollador como la plantilla colección de sitios

Administrador: Introduzca su nombre de usuario y haga clic en el botón de decisión para resolver el nombre de usuario

- Haga clic en el botón Aceptar para crear la colección de sitios.
- 4.- Configuración de biblioteca de documentos
- Con el fin de depurar y tener vista previa de los WebParts necesitamos cargar la página workbench.aspx, también tenemos que crear una nueva columna, las instrucciones son las siguientes:
- Crear la nueva columna.
 - Siga los pasos de abajo para agregar el nuevo campo a la Biblioteca Documentos:
 - Haga clic en el icono de engranajes en la parte superior derecha y haga clic en Configuración del sitio para abrir la página de configuración.
 - Haga clic en las bibliotecas y las listas del sitio en la categoría Administración de sitios.
 - Haga clic en Personalizar Documentos
 - Ahora, haga clic en Crear columna en Columnas:
 - Tipo ClientSideApplicationId como el nombre de la columna y dejar otros campos como son.
 - Haga clic en el botón Aceptar para crear la co-

lumna.

- Luego sube el workbench.aspx, última versión se puede encontrar aquí: <https://github.com/SharePoint/sp-dev-docs/wiki/Setup-SharePoint-Tenant>

5.- Configuración del entorno de desarrollo

Algo nuevo aquí, ahora podemos desarrollar con un sistema operativo MAC, ¿no es genial? primera vez que las herramientas son tan ampliamente disponibles a través de sistemas operativos para construir las nuevas experiencias de SharePoint.

- Instalar nodejs LTS (Long Term Support) versión. <https://nodejs.org/en/>
- Instalar NPM. Abra un símbolo del sistema y pegue el siguiente código:

```
npm -g install npm@next
```

- Si está utilizando Windows instalar este paquete:

```
npm install --global --production windows-build-tools
```

- Instalar VS CODE. <https://code.visualstudio.com/download>
- Instalar Yeoman y gulp. Abrir y ejecutar CMD.

```
npm i -g yo gulp
```

- Instalar Yeoman SharePoint generador.

```
npm i -g @microsoft/generator-SharePoint nt
```

Las herramientas anteriores son de código abierto y ampliamente utilizados en la comunidad por otro tipo de desarrollos, también son ampliamente utilizados en el nuevo ASP.NET core, así que aprender un poco acerca de ellos es un deber.

Nuestro Primer WebPart

Ahora, vamos a hacer un WebPart fácil, un WebPart que tenga una propiedad sencilla y se pueda ver el valor de esa propiedad dinámicamente. En primer lugar lo que tenemos que saber que lo que estamos construyendo aquí es puro WebPart del lado del cliente. Los WebParts del lado del cliente ya son soportados en SharePoint y en Office 365, pero de una manera diferente, se puede añadir un editor de código, y pegar el código JavaScript y HTML, no tendrán ningún problema, pero el problema con este enfoque es el mantenimiento, el código reside en una página, en la sección de código de un elemento Web, no hay manera de controlar el código fuente con TFS o Git.

En este punto, podemos usar CMD para algunos de los comandos que necesitamos para ejecutar, pero quiero in-



roducir una herramienta que algunos de ustedes pueden no conocer, CMDER, un gran reemplazo con una mejor apariencia y características avanzadas. Puede descargarlo aquí.

Office 365 es un producto muy importante en la vision de Microsoft

Paso 1

Vamos a crear nuestro directorio donde almacenaremos nuestra solución.

```
mkdir OurFirstSPFApp
cd OurFirstSPFApp
```

Paso 2

Crear el proyecto WebPart, para ello vamos a utilizar el generador Yeoman que construimos en el paso anterior

```
yo @microsoft/SharePoint
```

Se le pedirá una serie de preguntas:

- Aceptar el nombre por defecto como su nombre de la solución y pulse Enter. Seleccione Usar la carpeta actual para colocar los archivos.
- El siguiente conjunto de instrucciones le pedirá información específica acerca de su WebPart:
 - Aceptar el nombre por defecto como nombre de elemento Web y pulse Enter. Acepte la descripción predeterminada como su descripción de elementos Web y pulse Enter.
 - Aceptar el valor predeterminado No JavaScript Framework web como el Framework que desea utilizar y pulse Enter.

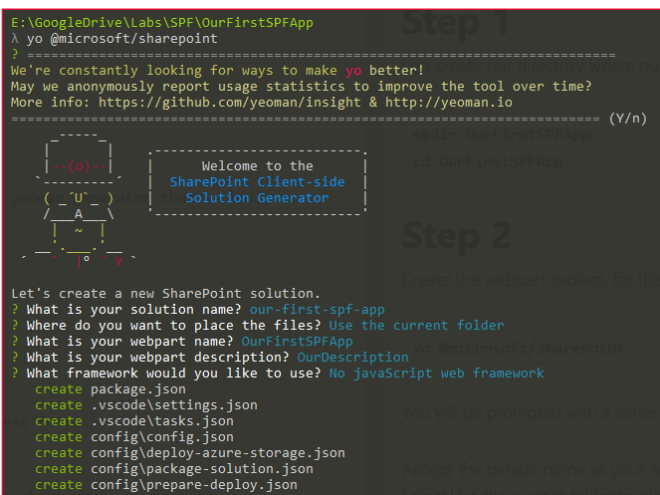


Imagen 1.- Configuración de Yeoman.

- Yeoman instalará todas las dependencias necesarias y la estructura del proyecto, como se muestra aquí:

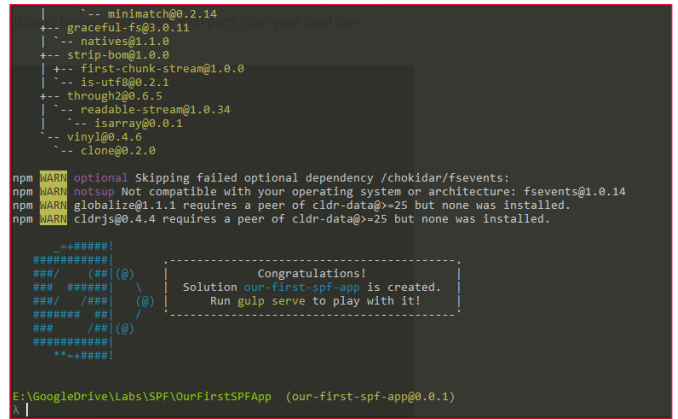


Imagen 2.- Instalación de las dependencias necesarias para Yeoman.

Paso 3

Se puede utilizar cualquier editor de código que te gusta, a mi me gusta VS Code, y eso es lo que veras en las capturas de pantalla. Para abrir la parte web en VS Code sólo tiene que escribir:

```
code .
```

Paso 4

Ahora necesitamos una vista previa de nuestra web part, para eso tenemos que depender de gulp, gulp es un administrador de tareas que se encarga de construir procesos tales como:

- Combinar y minificar JavaScript y CSS.
- Ejecutar herramientas para llamar a las funciones de empaquetamiento y minificación antes de cada compilación.
- Compilar archivos SASS en CSS.
- Compilar los archivos de Typescript a JavaScript. En un futuro post vamos a personalizar el proceso de construcción con gulp. Por ahora, vamos a utilizar el siguiente comando:

```
gulp serve
```

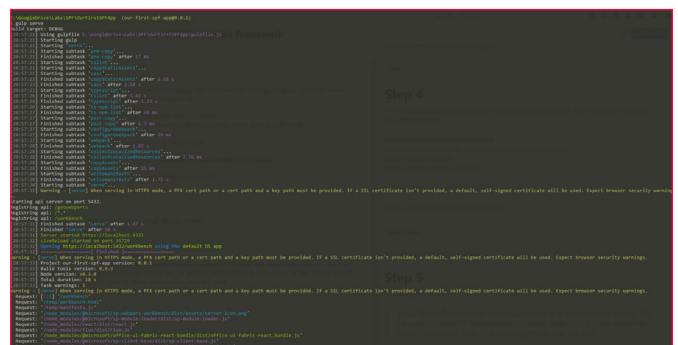


Imagen 3.- Creación del servidor de NodeJS.

- Esto creará un servidor NodeJS local. Una vez que creamos gulp serve un navegador se abrirá con nuestro workbench de trabajo local, este es el lugar donde vamos a editar nuestra página y agregar el elemento Web para la prueba.





Imagen 4.- Workbench local.

Modelo de desarrollo de WebParts que permite y soporta plenamente el desarrollo del lado del cliente

Paso 5

Vamos a compilar nuestro proyecto. Visual Studio Code proporciona soporte incorporado para el Gulp y otros administradores de tareas. Puede presionar **Ctrl + Shift + B** si está en Windows o **Cmd + Shift + B** si se encuentra en Mac para compilar, depurar y tener vista previa del WebPart.

Una vez que haga esto, usted será capaz de ver en la ventana de resultados como el proyecto se está construyendo.

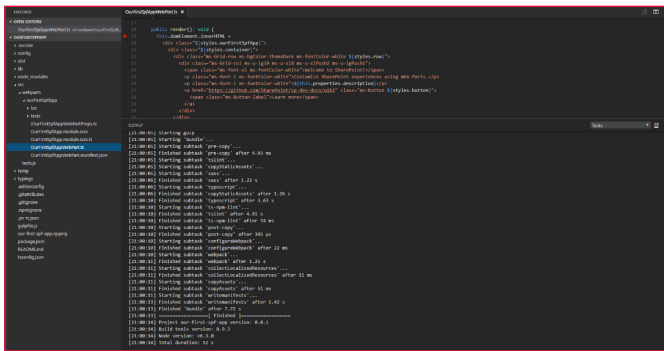


Imagen 5.- Resultados de la compilación del proyecto.

Ahora puede agregar el elemento Web a su página:

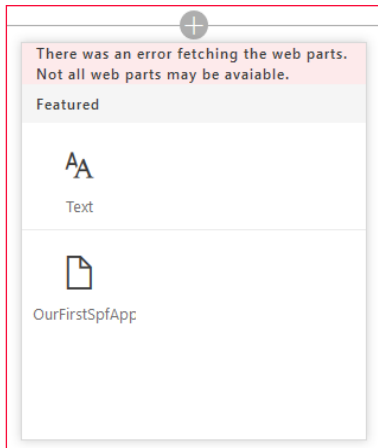


Imagen 6.- Agregando un WebPart en una página.

Y verlo en acción,

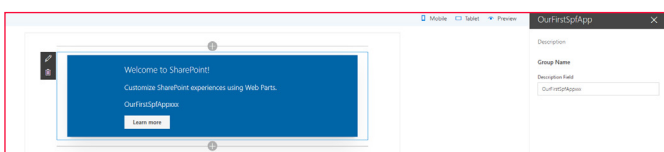


Imagen 7.- WebPart añadida.

Paso 6

Vamos a aprender acerca de la estructura del proyecto:

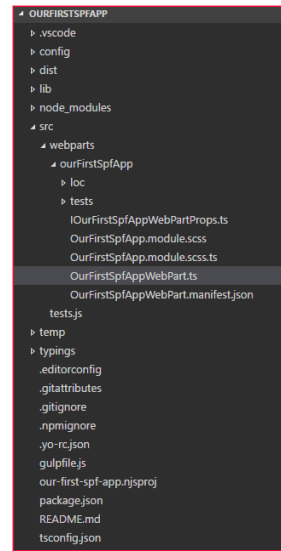


Imagen 8.- Estructura del proyecto.

El SRC es donde toda la magia sucede, esto es la carpeta que realmente importa, y en la que se codifica.

TypeScript es el idioma principal para construir WebParts de SharePoint del lado del cliente. TypeScript es un superconjunto escrito de JavaScript que se compila en pleno JavaScript. Las herramientas de desarrollo del lado del cliente de SharePoint se construyen utilizando clases de TypeScript, módulos e interfaces para ayudar a los desarrolladores a crear piezas solidas para los Web parts del lado del cliente.

Paso 7

Vamos a dar un paseo a través del código y los archivos más importantes:

- OurFirstSpAppWebPart.ts La clase de elemento Web.
- OurFirstSpAppWebPart.ts define el punto de entrada principal para su WebPart en la que la clase OurFirstSpAppWebPart extiende la clase BaseClientSideWebPart. Cualquier Web Part del lado del cliente se debe extender esta clase.

```
public constructor(context: IWebPartContext) {
    super(context);
}
```

- BaseClientSideWebPart implementa la funcionalidad mínima que se requiere para construir una WebPart. Esta clase también ofrece un montón de parámetros para validar y acceder a las propiedades de sólo lectura, como displayMode, propiedades de elementos web, el contexto, el instanceId del WebPart, el DOMElement y mucho más. Nótese que también estamos definiendo nuestra clase de Web Part para aceptar un tipo de propiedad IOurFirstSpfAppWebPartProps.
- Las propiedades se definen en una interfaz en un IOurFirstSpfAppWebPartProps.ts

```
export interface IOurFirstSpfAppWebPartProps {
    description: string;
}
```


Esta definición de la propiedad se utiliza para definir los tipos de propiedades personalizados para su web part, lo que vamos a ver en el panel de propiedades.

- Parte de Web render (). El elemento DOM donde del WebPart se mostrara es en el método render. Este método se utiliza para representar la web part dentro de ese elemento DOM. En nuestra web part OurFirstSpfA-ppWebPart, se pasa en el DIV para el elemento DOM. Los parámetros del método incluyen el modo de visualización (ya sea de lectura o de edición) y las propiedades de la web part configurados si los hubiere:

```
import {
  BaseClientSideWebPart,
  IPropertyPaneSettings,
  IWebPartContext,
  PropertyPaneTextField
} from '@microsoft/sp-client-preview';
import styles from './OurFirstSpfWebPart.module.scss';
import * as strings from 'ourFirstSpfWebPartstrings';
import { IOurFirstSpfWebPartWebPartProps } from './IOurFirstSpfWebPartWebPartProps';

export default class OurFirstSpfWebPartWebPart extends BaseClientSideWebPart<IOurFirstSpfWebPartWebPartProps> {
  public constructor(context: IWebPartContext) {
    super(context);
  }
  public render(): void {
    this.domElement.innerHTML = `
    <div class="${styles.helloWorld}">
    <div class="${styles.container}">
    <div class="ms-Grid-row ms-bgColor-themeDark ms-fontColor-white ${styles.row}">
    <div class="ms-Grid-col ms-u-lg10 ms-u-xl8 ms-u-xlPush2 ms-u-lgPush1">
    <span class="ms-font-xl ms-fontColor-white">Welcome to SharePoint!</span>
    <p class="ms-font-l ms-fontColor-white">Customize SharePoint experiences using Web Parts.</p>
    <p class="ms-font-l ms-fontColor-white">${this.properties.description}</p>
    <a href="https://github.com/SharePoint/sp-dev-docs/wiki" class="ms-Button ${styles.button}">
    <span class="ms-Button-label">Learn more</span>
    </a>
    </div>
    </div>
    </div>
    </div>`;
  }
}
```

Analizamos este código, el más importante, como se puede ver que importamos los estilos de un CSS, pero lo que es más sorprendente de TypeScript es que podemos reemplazar estilos dinámicamente usando variables, como por ejemplo:

```
$(styles.helloWorld)
También aquí se puede ver que podemos utilizar:
${this.properties.description}
```

Para utilizar las propiedades del panel de propiedades, pero ¿qué pasa si queremos hacer algo más interesante, lo que yo quiero que se fijen es algo llamado el panel de propiedades reactiva, podemos explicar sobre esto más adelante, pero básicamente lo que se ingrese o seleccione en sus propiedades se refleja inmediatamente en su web part en tiempo real, como llamados servicios web, actualización de valores, etc.

Para el ejemplo, he agregado 2 propiedades en el archivo: IOurFirstSpfWebPartWebPartProps.ts

```
export interface IOurFirstSpfWebPartWebPartProps {
  description: string;
  n1: string;
  n2: string;
}
```

Luego he añadido estas propiedades en el manifest.json:

```
{
  "$schema": "../../../../node_modules/@microsoft/sp-module-interfaces/lib/manifestSchemas/jsonSchemas/clientSideComponentManifestSchema.json",
  "id": "1ac1bfca-4ac6-4bb8-906e-6715ab383a2b",
  "componentType": "WebPart",
  "version": "0.0.1",
  "manifestVersion": 2,
  "preconfiguredEntries": [{
    "groupId": "1ac1bfca-4ac6-4bb8-906e-6715ab383a2b",
    "group": { "default": "Under Development" },
    "title": { "default": "OurFirstSpfWebPart" },
    "description": { "default": "OurFirstSPFWebPart description"
  }
},
  "officeFabricIconFontName": "Page",
  "properties": {
    "description": "OurFirstSpfWebPart",
    "n1": "1",
    "n2": "2"
  }
}
```

Y, finalmente, en mi método render he utilizado estos para sumar 2 números.

```
public render(): void {
  this.domElement.innerHTML = `
  <div class="${styles.ourFirstSpfWebPart}">
  <div class="${styles.container}">
  <div class="ms-Grid-row ms-bgColor-themeDark ms-fontColor-white ${styles.row}">
  <div class="ms-Grid-col ms-u-lg10 ms-u-xl8 ms-u-xlPush2 ms-u-lgPush1">
  <span class="ms-font-xl ms-fontColor-white">Welcome to SharePoint!</span>
  <p class="ms-font-l ms-fontColor-white">Customize SharePoint experiences using Web Parts.</p>
  <p class="ms-font-l ms-fontColor-white">${this.properties.description}</p>
  <a href="https://github.com/SharePoint/sp-dev-docs/wiki" class="ms-Button ${styles.button}">
  <span class="ms-Button-label">Learn more</span>
  </a>
  </div>
  </div>
  </div>
  </div>`;
}
```

Como puedes ver la sintaxis es muy parecida a Angular donde de alguna manera se enlaza lo que se esta digitando en el panel de propiedades, con lo que queremos que haga nuestro WebPart.

El resultado final

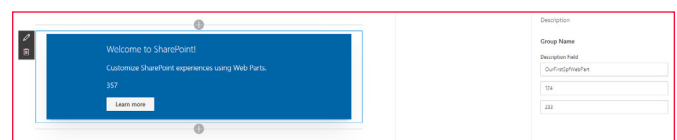


Imagen 9.- Resultado final obtenido.

LUIS VALENCIA
 Office Servers & Services MVP
 le.valencia@outlook.com
 @lvalencia
 http://www.luisevalencia.com

Como realizar una plantilla en Azure en formato ARM

Antes, en máquinas virtuales de tipo classic era muy fácil hacer una captura de una máquina virtual para luego utilizarla como plantilla para luego desplegar instancias.

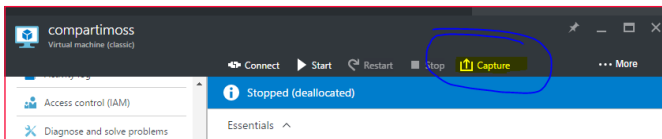


Imagen 1.- Como realizar una captura de máquina para máquinas "Classic".

Era tan fácil como darle al botón 'capture' . Pero el cuento ha cambiado...En el formato ARM, la cosa se complica un poco, ya que desde el portal de Azure no aparece el botón.

Voy a explicar a continuación como preparar una imagen de forma sencilla. También podemos hacerlo desde PowerShell, pero como veréis más adelante, Azure tiene un portal web dedicado a la gestión de los recursos Azure, que con tres clics nos va a permitir generar una imagen de nuestra máquina. Comencemos.

Desde este panel podemos observar los JSON y scripts que va a usar Azure para desplegar la máquina

Para empezar, vamos a crear una máquina de tipo ARM en el portal de Azure.

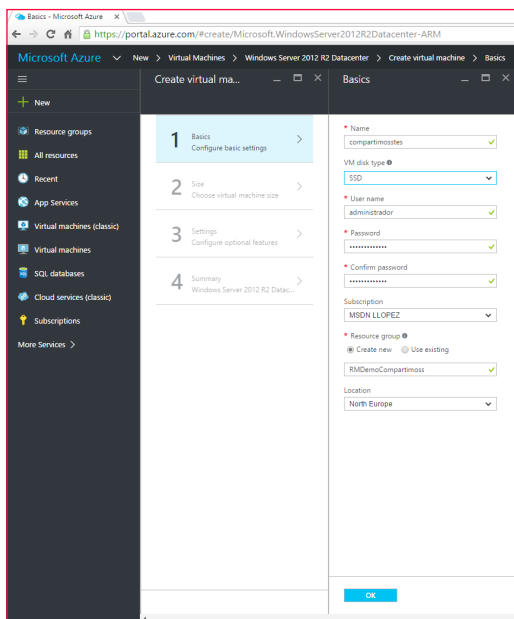


Imagen 2.- Creación de máquina ARM en Azure

Mientras que la estamos creando, podemos ajustar los diferentes parámetros para que se ajuste a nuestras necesidades. Yo lo dejo por defecto, porque es una demo.

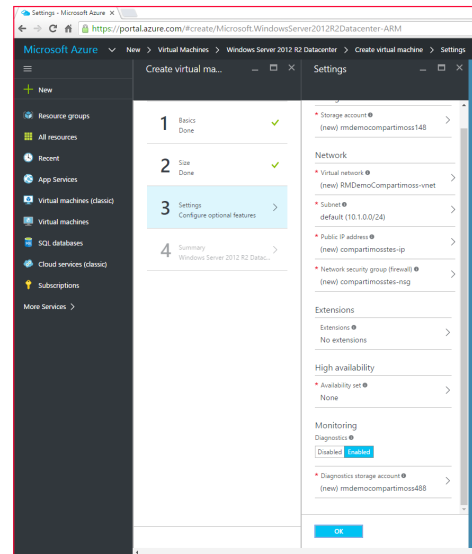


Imagen 3.- Parámetros adicionales de la máquina

Una vez que aceptamos los parámetros de la máquina, viene la primera parte de la magia.

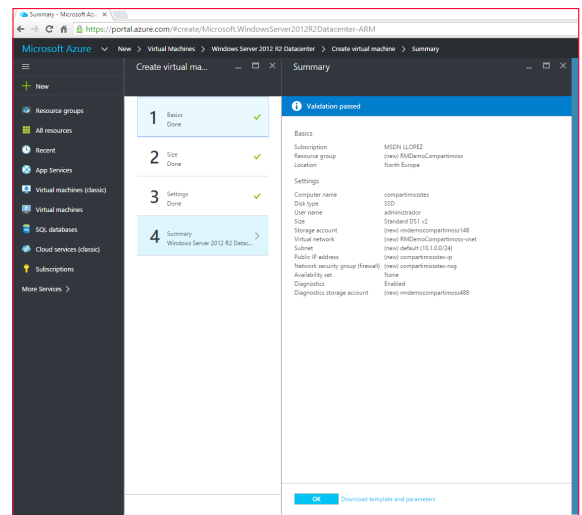


Imagen 4.- Resume den las configuraciones a aplicar.

En la pantalla de resumen, tenemos abajo un enlace a:

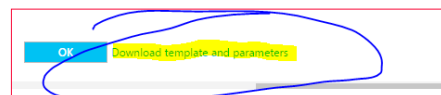


Imagen 5.- Enlace en la pantalla de resumen.

Si pulsamos sobre él se despliega otro panel

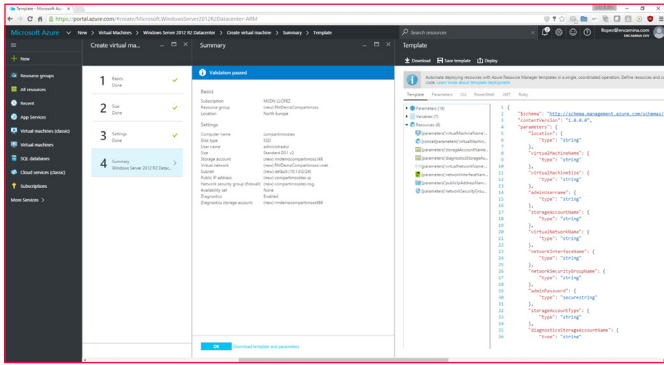


Imagen 6.- Panel desplegado.

Desde este panel podemos observar los JSON y scripts que va a usar Azure para desplegar la máquina. Si pulsamos sobre 'Download' se nos descargara un ZIP con los JSON y los scripts PowerShell, Ruby, bash y el código de C# para replicar la máquina. Almacenamos este ZIP porque lo usaremos posteriormente para desplegar instancias desde la imagen que capturemos.

Una vez guardado el ZIP podemos proceder a implementar nuestra máquina pulsando en OK. Esperamos un ratito y cuando se levante la máquina, nos conectamos a ella e instalamos el software que necesitamos. Ahora que ya la tenemos lista para plantillarla, le pasamos un sysprep y la apagamos.

Una vez la maquina se ha apagado tras ejecutar el sysprep, nos vamos al portal <https://resources.azure.com> nos logamos con la misma cuenta que en el portal de Azure.

En primer lugar, marcamos la casilla 'READ/WRITE' y seleccionamos la suscripción donde está la máquina.

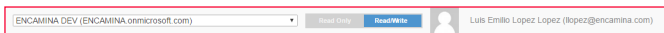


Imagen 7.- Selección de la máquina en una suscripción.

Una vez ubicada, desplegamos las suscripciones, y los grupos de recursos, y seleccionamos el grupo de corresponda.

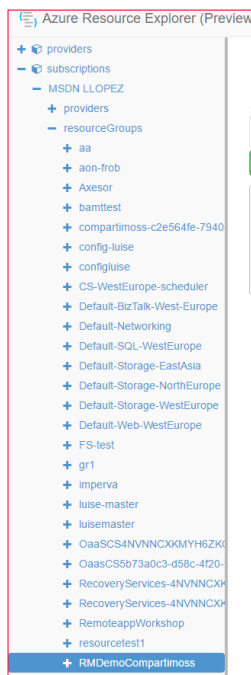


Imagen 8.- Grupos de recursos disponibles en Azure.

Lo desplegamos, hasta llegar hasta la máquina que queremos usar como plantilla.

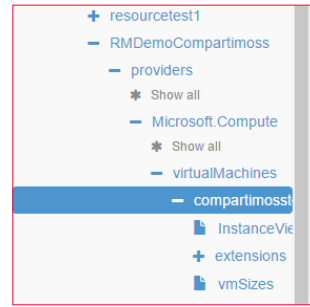


Imagen 9.- Máquina a utilizar como plantilla.

Una vez sobre la máquina, pulsamos sobre el botón 'Actions (Post/Delete)'.



Imagen 10.- Opción "Actions (POST, DELETE)"

Con lo que se nos desplegaran un montón de opciones a la izquierda:

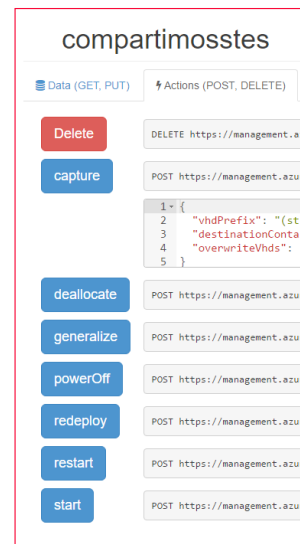
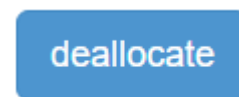


Imagen 11.- Opciones adicionales.

Desde esta ubicación, es donde capturaremos la maquina como plantilla. En primer lugar, le daremos al botón DEALLOCATE



Con esto apagaremos para Azure a la máquina.

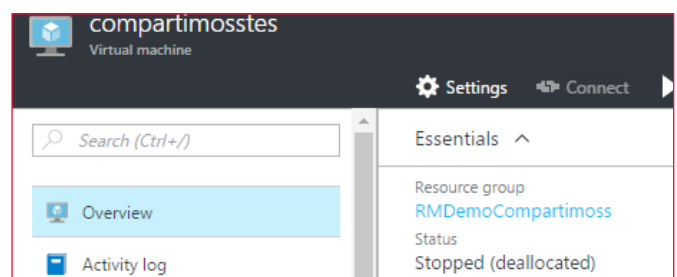


Imagen 12.- Máquina parada.

Una vez la maquina este en estado 'deallocated' le pulsaremos al botón 'Generalize'. Esto dejara a la maquina en el estado óptimo para capturarla como snapshot.



Es el momento de rellenar el formulario de la web en la sección 'Capture'. Yo suelo usar los valores reseñados, pero cada uno tiene sus gustos y necesidades.



Imagen 13.- Funcionalidad de "Capture".

Una vez pulsado el botón 'capture', en unos minutos tendremos nuestra plantilla lista dentro del 'Storage Resource'

donde ubicamos la máquina inicialmente en la ruta que se indica en la plantilla. Siempre es: system/Microsoft.Compute/Images/<destinationContainerName>

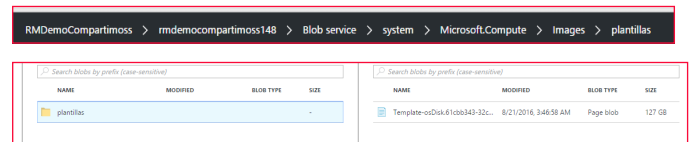


Imagen 14.- Plantilla de máquina lista para usar.

A partir de aquí, tenemos una imagen de plantilla de nuestra máquina virtual de origen y estamos en condiciones de instanciarla mediante scripting, usando los archivos que descargamos antes; pero eso os lo contare en futuras entregas.

LUIS EMILIO LÓPEZ

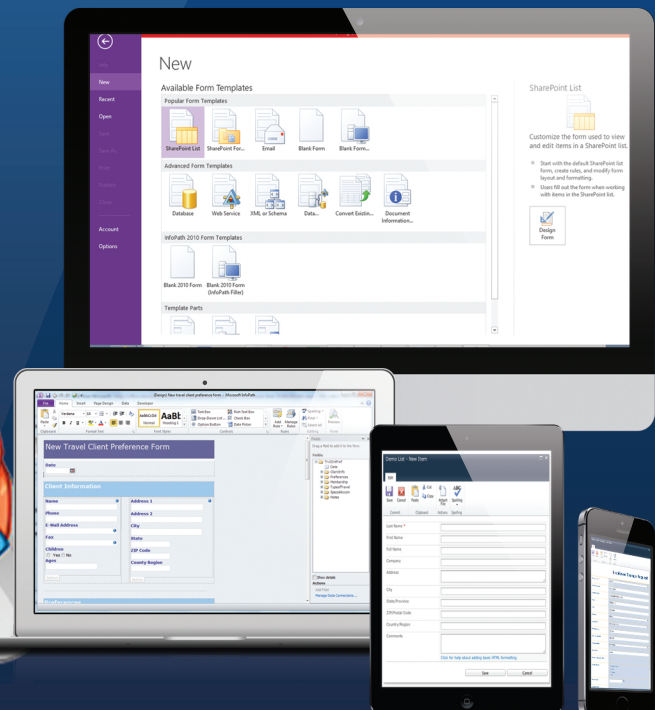
Responsable de Infraestructuras en Encamina

**Cree potentes formularios fácilmente,
SIN necesidad de conocimientos
técnicos**



La MEJOR alternativa para InfoPath

**Ensaye los Formularios
de KWizCom Forms**





21

Entrevista Yina Arenas

Soy la ingeniera liderando “Microsoft Graph” uno de los proyectos más ambiciosos y de alto perfil actualmente en Microsoft. Estoy transformando la plataforma de Office 365 y Azure que permite a los desarrolladores web construir aplicaciones colaborativas e inteligentes que usen los servicios de productividad de Microsoft.

Me llamo Yina Arenas, soy colombiana, estude Ingeniería Electrónica en la Universidad de los Andes en Bogotá y tengo un Master en ingeniería de Computación de la Universidad de Virginia. Desde el 2010 vivo en Redmond y trabajo para Microsoft en el desarrollo de interfaces de programación para que los desarrolladores puedan acceder de forma segura a la información en SharePoint, Azure



AD, Exchange y más servicios que forman parte del servicio Office 365.

Síguela en Twitter en @yina_arenas

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

Desde muy pequeña fui un “geek”, siempre me destacué en el colegio y tenía gran curiosidad por cómo funcionan las cosas. Pasaba mi tiempo entre haciendo experimentos de arte y jugando video juegos con mi hermano (Mario Bros, Mortal Kombat and Duke Nukem llevaron muchas horas de mi niñez), y aunque crecí queriendo dedicarme a la medicina, cuando llegó el momento de decidir mi carrera, tomé el camino de la ingeniería, un poco influenciada por el boom de internet al final de los 90. Desde entonces lo último en tecnología se convirtió en el frente de mi día a día. Como estudiante de ingeniería electrónica me interese por circuitos y sistemas embebidos, los precursores de lo que hoy conocemos como Internet de las cosas. Al graduarme trabajé por un año en el diseño e implementación redes de telefonía móvil y radio. En el 2006 decidimos mudarnos a los Estados Unidos con mi esposo e hijos y empezar estudios de postgrado en ingeniería de computación. Ahora con mi trabajo en Microsoft creo tecnología que impacta la vida de billones de personas en el planeta.

La educación en tecnología también ha sido una de mis pasiones, de pregrado desarrollé e implementé un programa para enseñar tecnología en colegios de escasos recursos en Bogotá. Durante mi maestría desarrollé un juego

de rol que enseña a los estudiantes de universidad las implicaciones sociales de la nanotecnología y ahora como profesional me involucro en actividades que promueven las ciencias, tecnología y matemáticas sobre todo en comunidades de minorías en tecnología, como las mujeres y los hispanos.

¿Cuáles son tus principales actividades tecnológicas hoy en día?

Mi trabajo implica crear las interfaces y la plataforma que permite que los desarrolladores se integren con los servicios de productividad de la nube de Microsoft. Empecé en el equipo de SharePoint en el 2010 diseñando el modelo de aplicaciones que permitió mover SharePoint de producto a servicio y mantener su capacidad de extenderlo para las necesidades de productividad de los usuarios. Trabajé en las interfaces (APIs), el modelo de autorización y autenticación (OAuth) y el modelo de seguridad para SharePoint 2013. Creé el servicio Office 365 Discovery que permite a los desarrolladores descubrir las direcciones de los usuarios en SharePoint para construir aplicaciones web y móviles, y luego llevé este mismo concepto de interfaces modernas y el modelo autenticación y autorización a Exchange y Azure AD, creando así una plataforma consolidada y conectando la información que hay en cada uno de los productos y servicios que componen Office 365. Hoy



mi día a día consiste en trabajar con docenas de equipos a través de Microsoft para unificar la plataforma de información e inteligencia de Microsoft, la llamamos Microsoft Graph. (<https://graph.microsoft.com>)

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

Mi familia ocupa el lugar número uno. Tenemos tres hijos varones, Santiago de 14, Benjamin de 11 y David de 2 años. Nos gusta mucho montar en bicicleta, hacer caminatas, jugar juegos de mesa y pasamos la mayoría de los fines de semana en sus partidos de futbol, los dos mayores, como buenos colombianos, llevan el amor por futbol en la sangre y son muy buenos jugadores.

¿Cuáles son tus hobbies?

Soy una persona muy visual y artística, durante varios años la joyería fue uno de mis más grandes hobbies, hacía bisutería y la vendía a boutiques de moda para financiar nuestros gastos como estudiantes y padres jóvenes (tuvimos nuestro primer bebe estando aun en el pregrado cuando tenía 19 años). Al mudarnos a los Estados Unidos pasé dos años disfrutando del rol de ser mama de tiempo completo antes de empezar mi maestría en la Universidad de Virginia, durante este tiempo tejer y correr se convirtieron en mis hobbies. Hoy en día me queda poco tiempo para el arte, mi tiempo libre es para mi familia (mi esposo y nuestros hijos), para ejercitarme y frecuentemente participo en actividades que promueven mujeres y diversidad en ciencia y tecnología.

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

La transición a la nube es inminente, tanto que ya no es visto como el futuro, sino como el presente. Lo que viene ahora es un mundo facilitado por la explotación de información, algoritmos inteligentes (Machine Learning) y servicios que empujan la barrera del conocimiento y que están al alcance de todos por medio de los móviles y el Internet de las cosas (IoT). En los próximos años veremos una proliferación de experiencias y servicios inteligentes y contextuales que nos ayudaran a completar tareas del día a día. Experiencias sin pantallas, asistentes inteligentes que hacen uso de la información disponible en la red, la información personal y la información contextual para hacer inferencias que resuelven y facilitan cada una de nuestras actividades de manera personalizada.

También veremos una transformación en el uso de la información (Big Data), hoy producimos más información por día que lo que producíamos por décadas hace unos años, de hecho, más del 90% de toda la información mundial se ha producido en los últimos dos años. La cantidad de información que tenemos disponible y el uso inteligente de esta información van a continuar transformando nuestras vidas, mejorando nuestros sistemas de salud, educación, seguridad y más aspectos de nuestra sociedad.

YINA ARENAS

Microsoft Office Graph Principal Program Manager

@yina_arenas

¿Conoces nuestras mini guías?



SharePoint y Azure – Azure Functions

Introducción

SharePoint y Azure forman una poderosa combinación. Los dos productos de Microsoft, el uno, SharePoint, que forma el corazón de las herramientas de colaboración y manejo de conocimiento, y el otro, Azure, que representa la estrategia de servicios en la nube de la compañía, se pueden combinar de múltiples formas para solucionar numerosos problemas empresariales. Esta serie de artículos analiza una por una, varias de las posibilidades en las que los dos productos de Microsoft se complementan y amplían su marco de operación.

Muchos de los servicios ofrecidos por Azure se pueden utilizar para ampliar y mejorar el funcionamiento de SharePoint, tanto on-premises como en la nube. Uno de los servicios que ofrece Azure son las “Funciones de Azure”, que es una solución de Plataforma como Servicio (PaaS) que proporciona microservicios y toda la infraestructura física más el software de base para crear unidades independientes de software que realizan algún tipo de servicio especializado. Las Funciones de Azure puede realizar para SharePoint un trabajo muy parecido al de los WebJobs que se describieron extensivamente en el artículo “SharePoint y Azure - Azure WebJobs” en el número 29 de Comparti-MOSS.

Cómo funcionan las Funciones de Azure

Las Funciones de Azure permite crear y ejecutar pequeñas piezas de código (“Funciones”) en la nube. Solamente es necesario crear el código en alguno de los lenguajes ofrecidos (CShare, FSharp, Node.js, Python, PHP, Batch, bash, Java) sin tener que pensar en la infraestructura para alojarlo o hacerlo ejecutar, pues de eso se encarga Azure, lo mismo que de escalar el sistema si se necesita más CPU y/o RAM. También solo se debe pagar por el tiempo que el código ejecuta, no por toda la infraestructura que necesite.

Dependencias tampoco son un problema pues la infraestructura de Funciones permite utilizar NuGet y NPM, de tal forma que se puedan utilizar librerías externas. Para proteger el acceso se puede utilizar OAuth junto con el Directorio Activo de Azure, Facebook, Google, Twitter o cuentas de Microsoft. Para escribir el código no es necesario disponer de Visual Studio pues las Funciones se pueden codificar di-

rectamente desde el Portal de Azure, aunque Integración Continua se puede lograr por medio de GitHub, Visual Studio Team Services. Por supuesto, si es preferible, el código se puede desarrollar en Visual Studio utilizando el Azure SDK, aunque otras suites de desarrollo (XCode, Eclipse, IntelliJ IDEA) también están soportadas.

Si se utiliza Visual Studio, el despliegue hacia Azure se puede realizar utilizando FTP, Git o por medio de despliegue directo usando la Web. En este caso, una de las ventajas es poder debuggear y utilizar el IntelliSense de Visual Studio, el que no está disponible si se desarrolla el código utilizando el Portal de Azure. El despliegue utilizando PowerShell también es posible, y existe toda una colección especializada de cmdlets de PowerShell que utilizan el Azure REST Management API para realizar la tarea.

Las Funciones de Azure permiten procesar datos, integrar sistemas, trabajar con sistemas externos (como SharePoint) y construir APIs y microservicios. El sistema de Funciones provee plantillas para facilitar la creación de código para sistemas de Azure Storage, reaccionar a eventos del Hub de Eventos de Azure, Queues del Bus de Azure, GitHub o HTTP lo mismo que ejecutar por medio de un esquema de tiempo predefinido.

El precio a pagar por las Funciones de Azure es muy económico y se debe costear solamente por el tiempo de ejecución y los recursos utilizados (RAM) durante ella, no por el sistema cuando no se está utilizando. El primer millón de ejecuciones y los primeros 400.000 GB de RAM por segundo utilizados es gratis; de ahí en adelante, cada millón extra de ejecuciones cuesta 20 centavos de dólar más 0.000008 dólares por cada GB de memoria utilizada por segundo.

Como se indicó anteriormente, las Funciones de Azure pueden substituir perfectamente los Azure WebJobs utilizados tradicionalmente para trabajar en conjunción con SharePoint en los siguientes escenarios típicos:

- Como reemplazo de Jobs en SharePoint OnPremises y para mover procesos que utilizan considerables cargas de recursos de los servidores de la granja (CPU y RAM) hacia servidores externos en Azure, que se pueden escalar de forma muy sencilla. También hay que hacer notar que la programación de Funciones de Azure, lo mismo que Azure WebJobs, es considerablemente más fácil que la programación de SharePoint Jobs.

- En SharePoint Online no es posible crear SharePoint Jobs customizados, por lo que Funciones de Azure y los Azure WebJobs forman su reemplazo.

Creación y configuración de una Función de Azure

Aunque toda la creación y configuración de Funciones de Azure se puede realizar remotamente desde Visual Studio o PowerShell, las siguientes instrucciones indican como hacerla directamente desde el Portal de Azure (lo mismo que su programación) para simplificar el proceso.

- 1.- Desde el Portal de administración de Azure (<http://portal.azure.com>) seleccione "Resource groups" -> "Add". Defina un nombre para el Grupo de Recursos, la suscripción a utilizar y la localización deseada.
- 2.- Seleccione el Grupo de Recursos creado en el punto anterior, utilice el botón de "Add" y en la casilla de búsqueda escriba "functions" y seleccione "Function App" en la lista resultados. Utilice el botón de "Create" en la nueva ventana. Una "Function App" es un contenedor de Funciones de Azure utilizado para agregar y organizar diferentes Funciones de una manera lógica.
- 3.- En la ventana de creación de la Function App indique un nombre, la suscripción a utilizar y el Grupo de Recursos (si no ha sido seleccionado automáticamente). El plan de servicio (seleccione "Dinámico" para que las funciones puedas escalar automáticamente), la localización del Servicio y la cantidad de memoria total que se pondrá a disposición de las Funciones en la Function App. Una Function App guarda la definición y configuración de sus funciones en un depósito de Azure Storage que es creado automáticamente, aunque se puede seleccionar alguno de los que ya existen en la Suscripción si así se desea
- 4.- Una vez creado el contenedor, su pantalla principal ofrece dos opciones para crear Funciones:
- 5.- Utilizando la ventana principal, se puede seleccionar si se desea crear una Función que utilice un esquema de tiempo (Timer), para procesar datos simplemente (Data processing) o para utilizar con aplicaciones Web o como un API (Webhook + API). También se puede seleccionar C# o JavaScript para crear la Función.

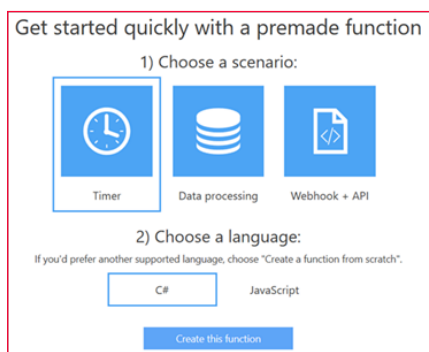


Imagen 1.- Escenarios para crear la Azure Function.

- Se puede utilizar también el botón de "New Function" en el menú al lado izquierdo, lo que ofrece un numero de plantillas que crean el código básico. Las plantillas se pueden filtrar por lenguaje de programación y/o tipo de escenario. Allí se puede crear una Función sin código si es necesario ("Empty - C#", por ejemplo).

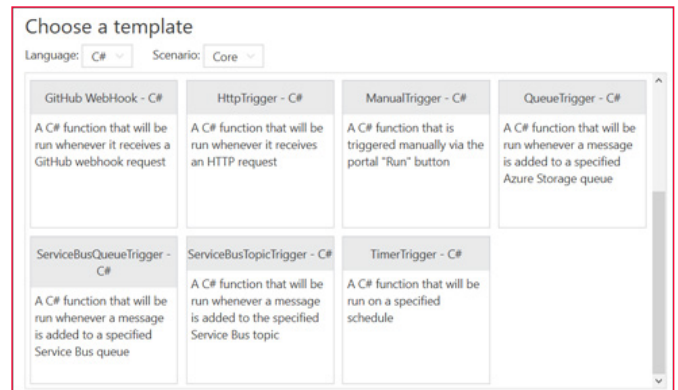


Imagen 2.- Plantillas para crear la Azure Function.

Ejemplo de utilización de una Función de Azure para SharePoint

El siguiente ejemplo simplemente lee el número de documentos en una Biblioteca de SharePoint y es exactamente igual al ejemplo utilizado en el artículo sobre WebJob Apps en el número 29 de CompartiMOSS. Aunque extremadamente sencillo en cuanto a funcionalidad, la idea del ejemplo es mostrar cómo crear y publicar una Función de Azure, no crear nueva funcionalidad. En el ejemplo se utiliza SharePoint Online, pero el código se puede utilizar sin ningún problema con una granja de SharePoint OnPremises, cambiando solamente el URL del servidor.

- 1.- Haga clic sobre "New Function" en el menú vertical izquierdo del portal de la Function App, y seleccione "TimerTrigger - C#". En la parte inferior de la página se puede definir el nombre de la Función y su esquema de tiempo; más información sobre la sintaxis del esquema se dará más adelante en este artículo, pero por el momento deje el valor por defecto ("0****") que hace que la Función dispare automáticamente cada minuto
- 2.- El editor de código de Funciones de Azure permite trabajar en diferentes formas:
 - El menú al lado izquierdo permite abrir las ventanas para escribir el código ("Develop"), definir disparadores, variables de entrada y de salida ("Integrate"), manejar la Función ("Manage") y monitorear su funcionamiento ("Monitor").

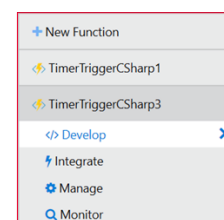
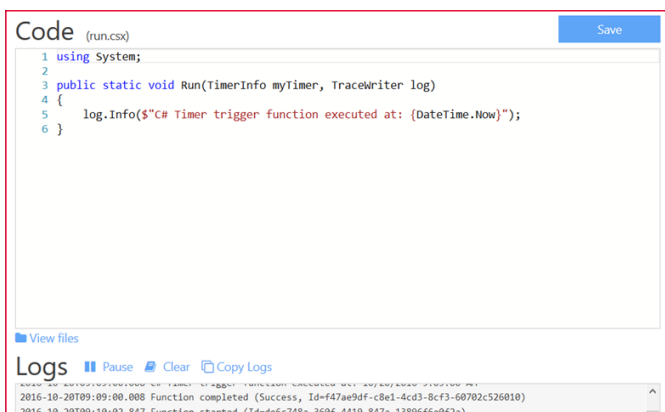


Imagen 3.- Acceso al editor para escribir código para la Azure Function.

- El área principal de trabajo en la página permite (cuando se selecciona “Develop”) crear el código, ver los logs, ejecutar y detener la Función y ver los valores de las variables.



```
Code (run.csx) Save
1 using System;
2
3 public static void Run(TimerInfo myTimer, TraceWriter log)
4 {
5     log.Info($"C# Timer trigger function executed at: {DateTime.Now}");
6 }
```

View files
Logs Pause Clear Copy Logs

2016-10-20T09:09:00.008 Function completed (Success, Id=f47ae9df-c8e1-4cd3-8cf3-60702c526010)
2016-10-20T09:10:02.847 Function started (Id=de6c748a-360f-4419-847a-13896fe8f27a)

Imagen 4.- Editor de código de la Azure Function.

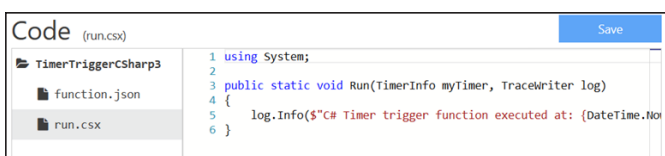
- 3.- El código en este caso (C#) es muy parecido al código que se utiliza en una aplicación de consola normal de Windows, y consiste de la declaración de directivas “using” al principio, y una función “Run” que contiene el código a ejecutar. Como se utiliza DotNet 4.6, toda la estructura interna de C# se puede utilizar en el código, incluyendo definiciones de variables, clases, rutinas, herencia, etc.

Los ensamblados básicos de .NET Framework son referenciados automáticamente (System, System.Collections.Generic, System.IO, System.Linq, System.Net.Http, System.Threading.Tasks, Microsoft.Azure.WebJobs y Microsoft.Azure.WebJobs.Host). Si se desea utilizar un dll propio, se puede subir el archivo con el dll al directorio donde se encuentra el archivo “run.csx” (como se indica en el siguiente punto del artículo) y referenciarlo utilizando la directiva.

```
#r "AssemblyName"
```

Como se indicó anteriormente, también se puede utilizar NuGet para subir todos los archivos con referencias necesarios. NuGet ofrece, además de la facilidad para referenciar todos los dlls necesarios en una operación, la posibilidad de mantener los dlls actualizados.

- 4.- Como el objetivo del ejemplo es trabajar con SharePoint, la forma más fácil para agregar todas las referencias necesarias es utilizar NuGet. En la sección principal del Editor de código haga clic sobre «View files» (en el borde inferior de la casilla de «Code»), lo que abre el explorador de archivos.



```
Code (run.csx) Save
TimerTriggerCSharp3
├── function.json
├── run.csx
└── run.csx
1 using System;
2
3 public static void Run(TimerInfo myTimer, TraceWriter log)
4 {
5     log.Info($"C# Timer trigger function executed at: {DateTime.Now}");
6 }
```

Imagen 5.- Explorador de archivos para las Azure Functions.

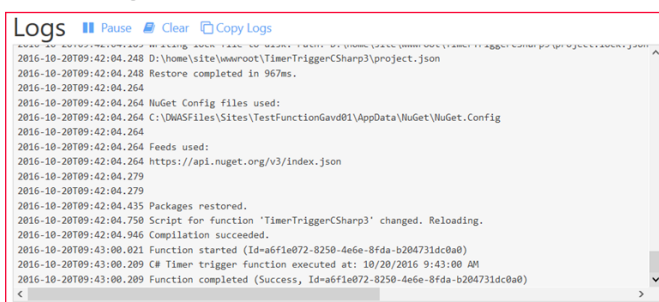
Como se puede ver, el archivo “run.csx” es el que con-

tiene el código de la Función. El archivo “function.json” contiene por defecto la definición del disparador, aunque se pueden agregar más definiciones para el uso interno de la Función si es necesario.

- 5.- Los paquetes NuGet que se desean aplicar se definen en un archivo adicional. Utilizando el “+” del explorador cree un nuevo archivo llamado “project.json” (es obligatorio que ese sea el nombre). En la zona de código copie el siguiente fragmento:

```
{
  "frameworks": {
    "net46": {
      "dependencies": {
        "AppForSharePointOnlineWebToolkit": "3.1.5"
      }
    }
  }
}
```

Este código indica que se va a utilizar el NuGet “AppForSharePointOnlineWebToolkit” versión 3.1.5 que utiliza .NET Framework 4.6. Los NuGet y las versiones disponibles se puede encontrar en el sitio de bibliotecas NuGet en Internet (<http://www.nuget.org>). Si se está trabajando con SharePoint OnPremises, se puede utilizar el NuGet “AppForSharePointWebToolkit”, que produce los mismos resultados. Al utilizar el botón de “Save”, note en la ventana de Logs que el NuGet es descargado e instalado:



```
Logs Pause Clear Copy Logs
2016-10-20T09:42:04.248 D:\home\site\wwwroot\TimerTriggerCSharp3\project.json
2016-10-20T09:42:04.248 Restore completed in 967ms.
2016-10-20T09:42:04.264
2016-10-20T09:42:04.264 NuGet Config files used:
2016-10-20T09:42:04.264 C:\DNASFiles\Sites\TestFunctionGavd01\AppData\NuGet\NuGet.Config
2016-10-20T09:42:04.264
2016-10-20T09:42:04.264 Feeds used:
2016-10-20T09:42:04.264 https://api.nuget.org/v3/index.json
2016-10-20T09:42:04.279
2016-10-20T09:42:04.279 Packages restored.
2016-10-20T09:42:04.435 Packages restored.
2016-10-20T09:42:04.750 Script for function 'TimerTriggerCSharp3' changed. Reloading.
2016-10-20T09:42:04.946 Compilation succeeded.
2016-10-20T09:43:00.021 Function started (Id=a6f1e072-8258-4e6e-8fda-b204731dc0a0)
2016-10-20T09:43:00.209 C# Timer trigger function executed at: 10/20/2016 9:43:00 AM
2016-10-20T09:43:00.209 Function completed (Success, Id=a6f1e072-8258-4e6e-8fda-b204731dc0a0)
```

Imagen 6.- Descarga de los paquetes de NuGet necesarios.

- 6.- Seleccione el archivo run.csx y reemplace el código por defecto con el siguiente fragmento:

```
using Microsoft.SharePoint.Client;
using System;
using System.Security;

public static void Run(TimerInfo myTimer, TraceWriter log)
{
    string mySharePointSiteUrl = "https://dominio.sharepoint.com/sites/teamsiteone";
    string mySharePointBiblioteca = "NombreBiblioteca";
    string myCuenta = "usuario@dominio.onmicrosoft.com";
    string myClave = "clave";

    using (ClientContext myContext = new ClientContext(mySharePointSiteUrl))
    {
        SecureString myClaveSegura = new SecureString();
        foreach (char unChar in myClave)
        {
            myClaveSegura.AppendChar(unChar);
        }

        myContext.AuthenticationMode = ClientAuthenticationMode.Default;
    }
}
```

```

myContext.Credentials = new SharePointOnlineCreden-
tials(myCuenta, myClaveSegura);

try
{
    List myLista = myContext.Web.Lists.GetByTitle(myShare-
PointBiblioteca);
    myContext.Load(myLista);
    myContext.ExecuteQuery();

    if (myLista != null && myLista.ItemCount > 0)
    {
        log.Info(myLista.Title.ToString() + " tiene " + myLista.Item-
Count + " documentos");
    }
    else
    {
        log.Info("No hay documentos");
    }
}
catch (Exception ex)
{
    log.Info("Error: " + ex.ToString());
}
log.Info($"C# Timer trigger function executed at: {DateTime.
Now}");
}

```

Inicialmente se crean las directivas "using" para trabajar con el dll cliente de SharePoint y con las clases de System. Dentro de la rutina de "Run" se definen primero variables con el URL del sitio de SharePoint Online donde se encuentra la Biblioteca con los documentos, el nombre de la Biblioteca y los datos para logearse en el sitio (usuario y clave). Luego, utilizando el Contexto del sitio de SharePoint, se logea al usuario, se crea un objeto para contener la Biblioteca y, luego de ejecutar la consulta, se muestra por pantalla el número de documentos en ella. Note que se utiliza el objeto "log" en lugar de "Console" para mostrar la información en la ventana de Logs. Aunque el ejemplo es muy básico en cuanto a funcionamiento de SharePoint, muestra claramente cómo se puede hacer contacto con el sistema y realizar algún tipo de trabajo en él.

7.- Utilice el botón de "Save", y revise el funcionamiento de la Función en la ventana de Logs. Note que la función dispara automáticamente cada minuto.

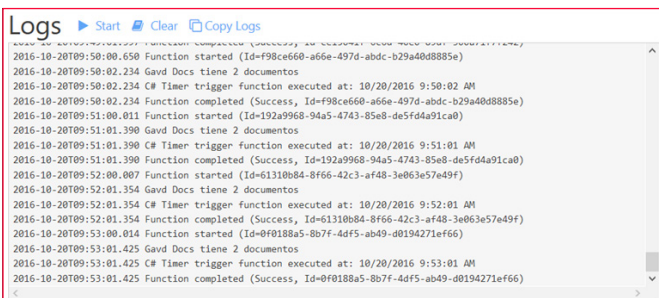


Imagen 7.- Funcionamiento de la Azure Function en la ventana de Logs.

8.- Haga clic sobre "Monitor" en el menú al lado izquierdo para ver la información sobre el funcionamiento de la Función, incluyendo el tiempo que demora cada ejecución (importante para calcular el precio final a pagar por el uso de Función en Azure):

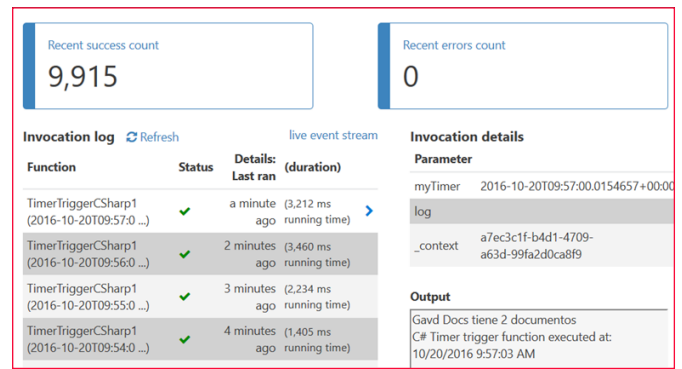


Imagen 8.- Monitorización del funcionamiento de la función.

9.- La Función se puede detener (y reiniciar) utilizando el botón "Pause" y "Start" que se encuentra al lado derecho de "Logs".

10.- La forma de temporizar la Función se definió en el momento de creación en Azure (punto 5). El archivo llamado "function.json" contiene la definición del temporizador como una expresión CRON (<https://en.wikipedia.org/wiki/Cron>) que incluye seis campos: {segundo} {minuto} {hora} {día} {mes} {día de la semana}. Algunos ejemplos de expresiones se pueden encontrar en la sección de "Integrate" (menú izquierdo en la definición de la Función). El valor se puede configurar directamente en el archivo "function.json", parámetro "schedule" o utilizando la casilla de "Schedule" de la sección "Integrate"

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

Conclusiones

Dentro de las múltiples maneras de utilizar los servicios de Azure para complementar el funcionamiento de SharePoint, las Funciones de Azure permiten mover operaciones que requieren muchos recursos de CPU y/o memoria fuera de la granja de SharePoint hacia procesos remotos en Máquinas Virtuales, y ejecutar programas según un esquema de tiempo predefinido, tal como lo hacen los Jobs nativos de SharePoint y las aplicaciones de WebJob de Azure.

Funciones de Azure se pueden utilizar en conjunto con instalaciones de SharePoint OnPremises como una alternativa a los SharePoint Jobs nativos, o con SharePoint Online para mimetizar el funcionamiento de los Timer Jobs personalizados que no es posible de crear en SharePoint Online.

GUSTAVO VELEZ
MVP Office Servers and Services
gustavo@gavd.net
http://www.gavd.net

Usando R-script en Power BI

¿Qué es R-script?

R es un lenguaje y entorno utilizado para informática estadística y gráfica. Es una modificación del lenguaje S, por lo que existe código escrito en S, que es capaz de ser implementado en R sin modificaciones.

R proporciona una amplia variedad de modelos estadísticos (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series de tiempo, clasificación, agrupación, etc.) y técnicas gráficas, lo que la convierte en una herramienta muy extensible y manipulable por parte del usuario.

Una de las fortalezas de R es la facilidad con la que se pueden producir gráficos de publicación con buena calidad y diseño, incluyendo símbolos matemáticos y fórmulas donde sea necesario.

R está disponible como Software Libre bajo los términos de la GNU General Public License de la Free Software Foundation en forma de código fuente. Compila y ejecuta en una amplia variedad de plataformas UNIX y sistemas similares (incluyendo FreeBSD y Linux), Windows y MacOS.

Por mayor información acerca de R y sus capacidades referirse a la página oficial R-project.

R está disponible como Software Libre bajo los términos de la GNU General Public License

Habilitar R en Power BI

Como primer paso para tener disponible R en Power BI, será realizar su instalación. Al tratarse de un software libre, el mismo puede ser obtenido de múltiples orígenes y repositorios; siguiendo las recomendaciones de Microsoft, se recomienda obtener los ejecutables desde la página Revolution Open o el repositorio CRAN.

Instalación desde Revolution Open:

Descargar la última versión estable disponible:

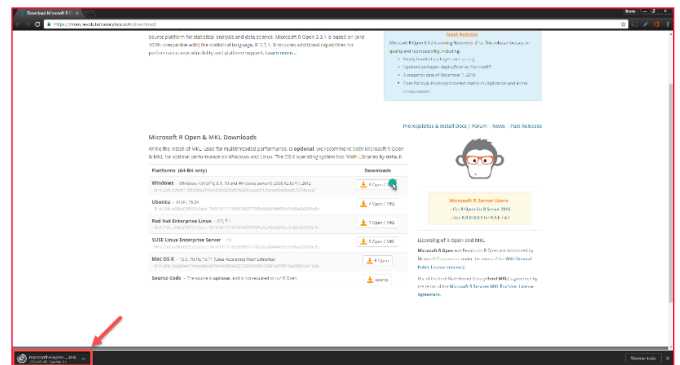


Imagen 1.- Descarga de R desde Revolution Open.

Una vez descargado el msi, ejecutarlo y seguir los siguientes pasos (marcados con el cursor y los steps):

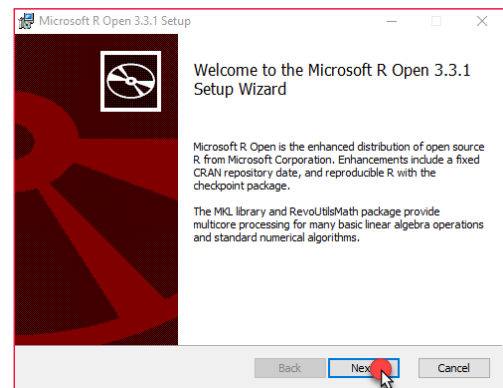


Imagen 2.- Instalador de R.

Una vez iniciado el instalador, aceptar los términos y condiciones y continuar con la instalación

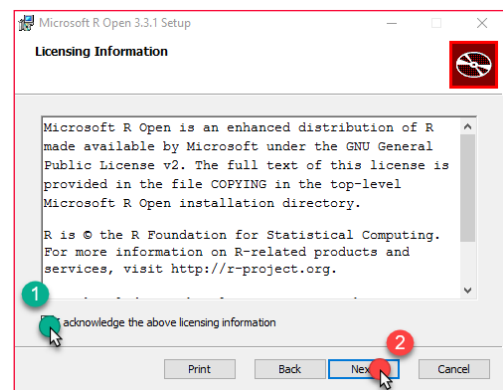


Imagen 3.- Información de licencia para R.

Este paso es opcional, aunque se recomienda la descarga de Intel MKL, debido a que mejora el rendimiento de ejecución de cálculos y medidas. De desear descargar el componente, dejar marcada la opción Install Math kernel



Library (Intel® MKL), en caso contrario desmarcar la opción y continuar con la instalación.

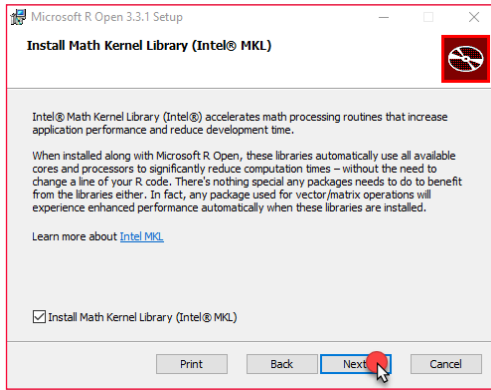


Imagen 4.- Ventana para seleccionar la instalación de MKL.

Si se eligió descargar MKL, aceptar los términos y condiciones y proseguir con la instalación.

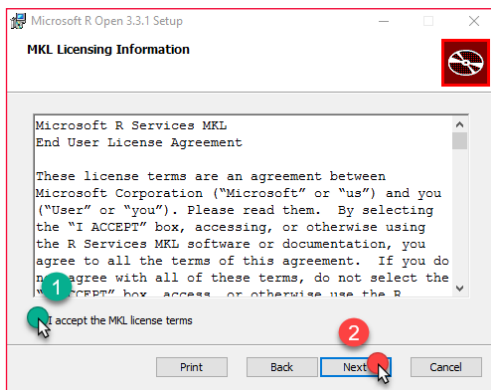


Imagen 5.- Acuerdo de licencia para MKL.

Especificar la carpeta destino de la instalación (se recomienda dejar la que viene por defecto, de forma de que Power BI la pueda detectar automáticamente)

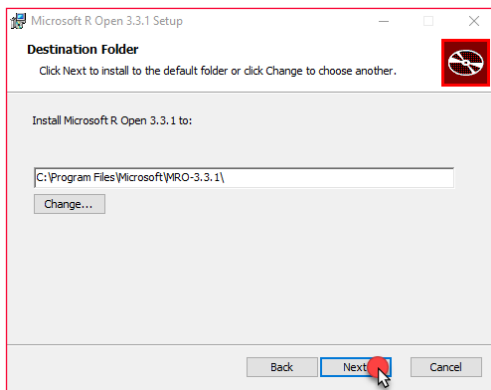


Imagen 6.- Ruta de instalación de R.

El lenguaje R puede ser utilizado en Power BI para diversas funcionalidades, ya sea para modelar data sources

Una vez completados los pasos anteriores, se encuentran configuradas todas las opciones necesarias para realizar la instalación.

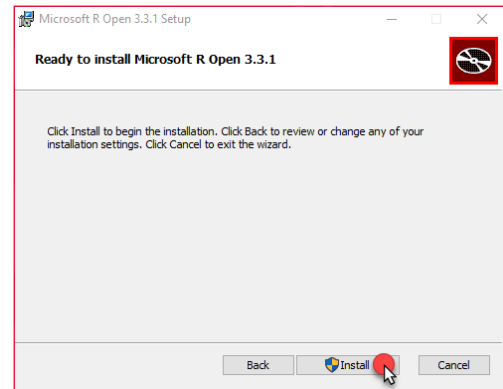


Imagen 7.- Inicio del proceso de instalación de R.

Una vez finalizada la instalación cerrar el dialogo de instalación.

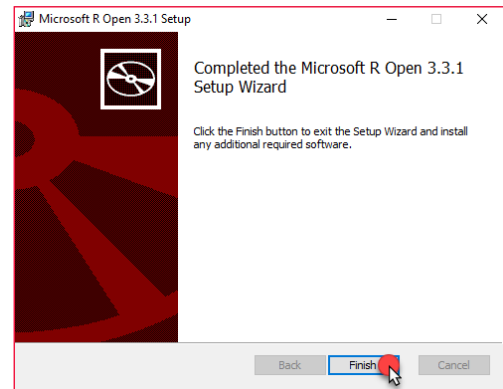


Imagen 8.- Finalización de la instalación de R.

Referenciar R desde Power BI

Al completar la instalación, Power BI identificará automáticamente la instalación realizada (siempre y cuando se haya dejado la carpeta de destino la predefinida). Para verificar la referencia o cambiar la carpeta de destino realizar los siguientes pasos:

- Ingresar a Power BI Desktop.

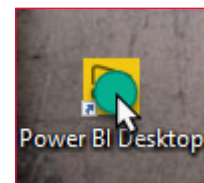


Imagen 9.- Acceso a Power BI Desktop.

- Una vez en la aplicación dirigirse a Archivo (File) -> Options and Settings -> Options.

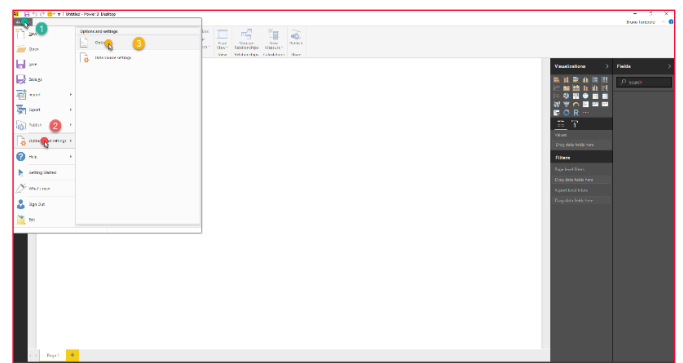


Imagen 10.- Acceso a las opciones de Power BI.

- Dentro de las opciones dirigirse a la pestaña R script. En la pestaña podremos ver la carpeta destino de la instalación de R-script.

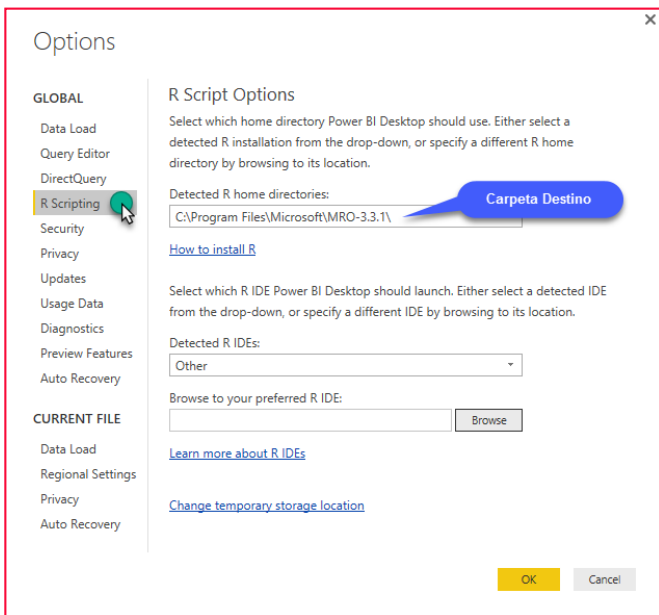


Imagen 11.- Carpeta de destino de la instalación de R.

En el siguiente ejemplo, se utiliza la sentencia require, para que se realice la importación del módulo timeDate (módulo que habilita funcionalidades de manejo de fechas).

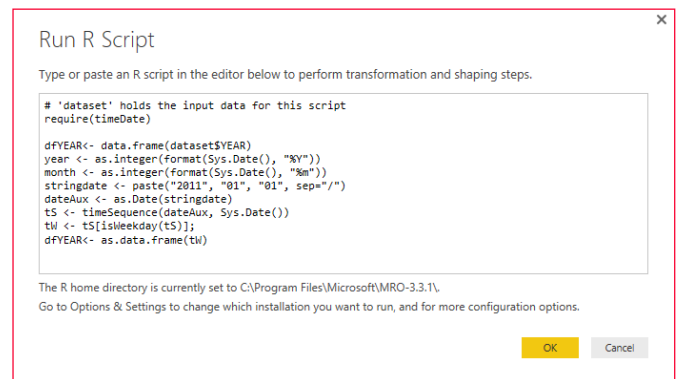


Imagen 12.- Ejemplo de uso de require para importar un módulo.

Si el modulo necesario no existe o no se ha importado, Power BI mostrará el siguiente error en la ejecución del script.



Imagen 13.- Error en Power BI si el módulo no se encuentra.

Importación de módulos

La instalación por defecto de R, incluirá ciertos módulos, pero resultará imperativo el obtener funcionalidades extras. Estas importaciones se pueden realizar directamente desde la consulta R que se está ejecutando a través de la utilización del comando:

```
install.packages('[NOMBRE MODULO]')
```

Todos los módulos de R disponibles pueden hallarse en Available CRAN Packages By Name

Problemas de permisos

Es común encontrarse con problemas de permisos al intentar instalar módulos de R directamente desde Power BI. Estos errores de permisos se deben a que la carpeta library (carpeta en donde se guardan las referencias de los módulos importados) de R no está habilitada para escritura por defecto. Para solucionar el incidente, debemos modificar la seguridad de la carpeta y habilitar la escritura de la misma.

R proporciona una amplia variedad de modelos estadísticos

Módulos y sentencia require

Las consultas R, a menudo incluyen sentencias de require, las cuales especifican que para ejecutar la porción de código contigo es necesario utilizar uno o más módulos específicos.

Para solucionar el incidente es necesario agregar la sentencia de instalación del módulo previa a la sentencia de requerimiento require, tal como se muestra a continuación.

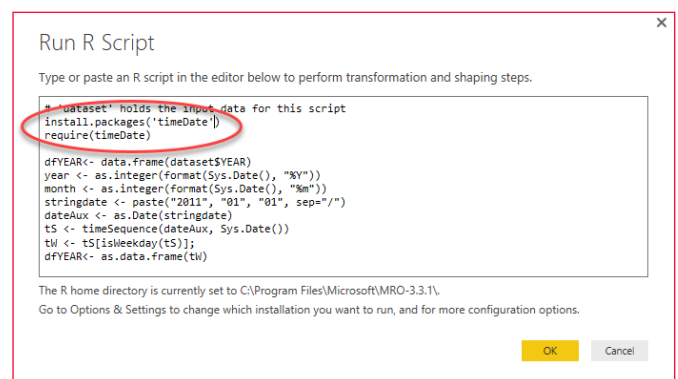


Imagen 14.- Solución al error que se produce cuando no se encuentra el módulo.

Utilización de R-script

El lenguaje R puede ser utilizado en Power BI para diversas funcionalidades, ya sea para modelar data sources, crear visualizaciones o simplemente crear un data source de cero a partir de comandos de R. La posibilidad de utilizar este lenguaje es lo que le da una flexibilidad extra a Power BI y es lo que más cautiva a los programadores a la hora de ver a Power BI como algo más que una herramienta de reportería.

R-script como source de entrada

Dentro de las opciones de data sources se encuentra la opción de utilizar una consulta R-script. La misma viene integrada por defecto a PowerBI Desktop y puede ser accedida desde la opción Get Data -> More

opción se encuentra disponible en la sección Queries de Power BI, bajo la pestaña Transform. Partiendo de un reporte con cierto origen estipulado:

- Debemos acceder a Edit Query:

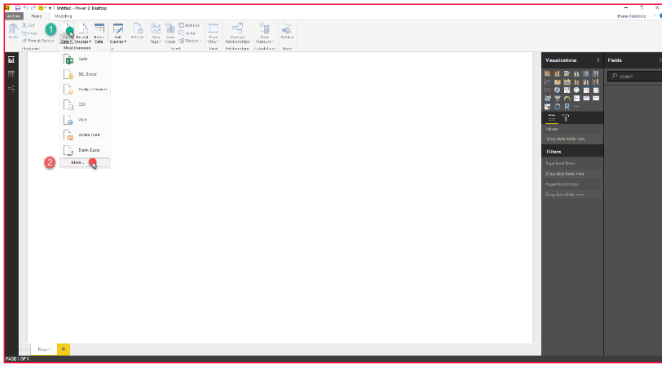


Imagen 15.- Acceso a la opción de usar una consulta R como data source de Power BI.

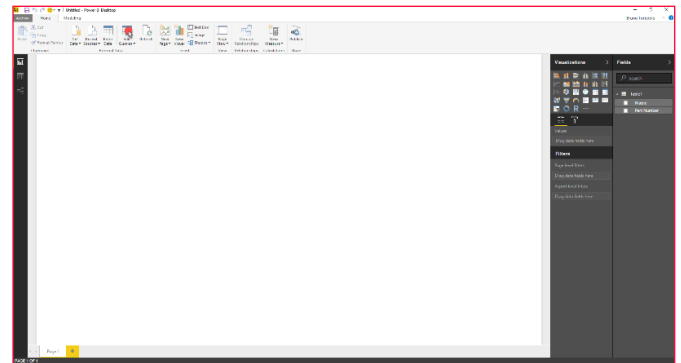


Imagen 18.- Acceso a la opción Edit Query.

- Bajo la pestaña Transform, en la sección Scripts, encontraremos la opción R

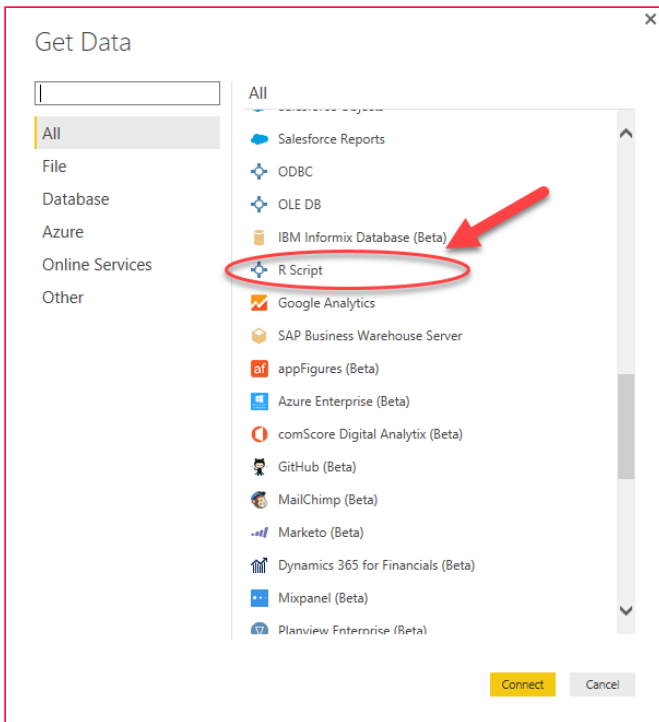


Imagen 16.- Origen de datos R en Power BI.

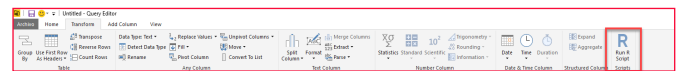


Imagen 19.- Opción Run R Script en la pestaña Transform.

R script para crear visuales

Sin duda una de las opciones más utilizadas, debido a las características del lenguaje, es el utilizarlo para crear visualizaciones personalizadas dentro de Power BI. La utilización de R dentro de Power BI para las visuales es muy simple, solo hay que elegir el tipo de visualización R y especificar el origen de datos, para que se nos habilite un IDE de R embebido en la página de Power BI Desktop.

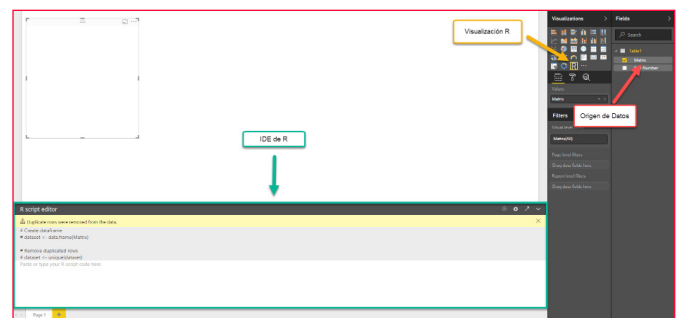


Imagen 20.- Utilización de R para representaciones visuales en Power BI.

Aplicaciones prácticas de R-script en Power BI

Muchos ejemplos pueden ser encontrados en el foro de Power BI. Nosotros nos centraremos en dos experiencias personales que hemos tenido, la primera a la hora de modelar entradas de archivos de texto con formatos aleatorios a lo largo del tiempo y la utilización de R para exportar grandes volúmenes de datos de una visual a un csv (Power BI solo posibilita la exportación de hasta 1500 entradas)

Modelado de un TXT

En los orígenes disponibles tendremos la opción R script.

Una vez elegida la opción tendremos un recuadro en donde especificar la consulta R.

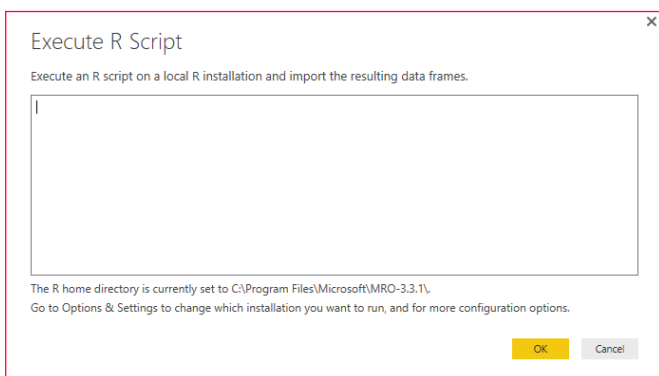


Imagen 17.- Ventana para especificar la consulta R.

R script en modelado de información

Power BI ofrece la posibilidad de modelar la información de cualquier origen a través de la utilización de R. Esta

En este caso vamos a reflejar cómo obtener información de un archivo de texto que genera un sistema ERP, el cual no sigue un pre-modelado, sino que la estructura del archivo depende de la cantidad de información exportada. Para poder llevar la entrada a un formato específico y de esa forma crear las estructuras necesarias para crear los reportes solicitados, se requirió la utilización de scripts en R.

- Lo primero realizado fue importar el archivo de texto como data source y se procedió a editar la consulta a través de la inserción de un R-script.

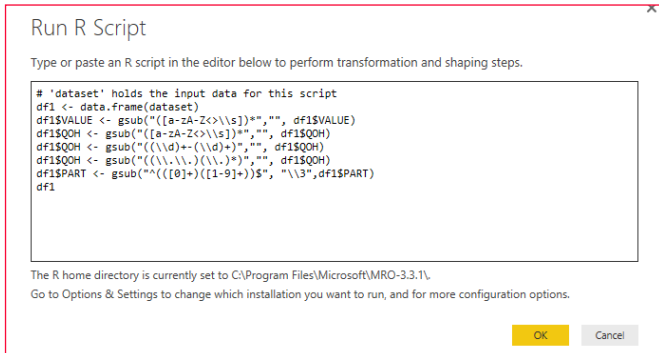


Imagen 21.- Importación del archivo de texto con R-Script.

- Power BI ofrece la tabla de entrada como un dataset, el cual puede ser accedido a través de la invocación de dataset. A través de esto lo primero en efectuar es pasar el dataset a un frame el cual posibilite la modificación de las columnas y celdas de la tabla como una matriz. Esto se puede efectuar a través de la utilización de la sentencia:

```
df1 <- data.frame(dataset)
```

En este paso, hemos convertido la tabla en un frame el cual está guardado en la variable df1. El siguiente paso será modelar la información de las entradas a través de la utilización de la formula gsub.

La fórmula gsub permite a través de la utilización de regex, reemplazar sub cadenas de una cadena. La fórmula se aplica individualmente a cada una de las celdas de la columna especificada como entrada. El formato de la fórmula es el siguiente:

```
gsub("[REGEX PATTERN]", "[REPLACEMENT STRING/PATTERN]", [INPUT COLUMN])
```

Una vez modeladas las entradas de las columnas VALUE (accesible a través de df1\$VALUE), QOH (accesible a través de df1\$QOH) y PART (accesible a través de df1\$PART), especificamos el valor de retorno df1, de forma de que se auto genere la tabla resultado con las entradas modeladas y listas para ser utilizadas en el modelo.

Exportación de grandes registros a CSV

Power BI nos ofrece la posibilidad de exportar a un csv las entradas utilizadas en una visual (la exportación contendrá los filtros y transformaciones realizadas) de forma de manipular las entradas fuera del ámbito del programa.

Para exportar las entradas de una visual debemos posicionarnos sobre el vértice superior derecho y seleccionar la opción Export Data



Imagen 22.- Exportación de los datos de un visual.

Esta funcionalidad cuenta con una limitante de registros a exportar, si el mismo supera el umbral permitido, se nos presentará el siguiente mensaje indicando que se exportará la información con errores.

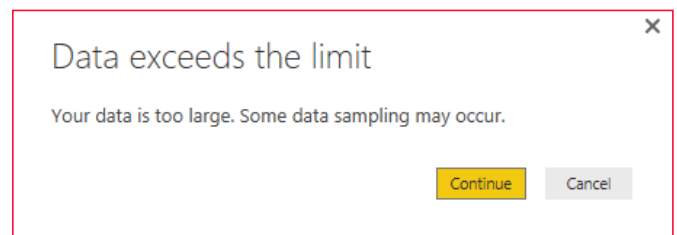


Imagen 23.- Error mostrado cuando se excede el límite de datos a exportar.

Ante estos casos podemos utilizar un script R para exportar hasta 150000 registros a un csv. Para efectuar esto debemos elegir la visualización R y utilizar el siguiente script.

```
write.table(dataset, file="[PATH OF FILE]", sep = "[SEPARATOR]", row.names = FALSE)
plot(dataset);
```

Conclusión

Esto es solo un punto de partida para demostrar la flexibilidad que ofrece Power BI a través de este lenguaje, esperamos que con esto se motiven a seguir investigando e inventando nuevas funcionalidades con el lenguaje R y su aplicación en Power BI.

BRUNO TORTELORO
Software Developer en Arkano Software

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.



Integración continua en SharePoint con Team Services y PnP



En el siguiente artículo veremos cómo podemos hacer uso del framework del PnP junto a Visual Studio Team Services, para configurar una Build de Integración Continua en proyectos de SharePoint. La mayor parte de lo comentado en este artículo, es igualmente válido para SharePoint On Premises, si bien las diferentes imágenes o fragmentos de código, están orientados a SharePoint Online.

Entornos en proyectos SharePoint Online

Históricamente, y sobre todo a estas alturas, todos tenemos claro cómo trabajar a nivel de entornos con SharePoint On Premises. Para entornos de desarrollo, cada developer dispone de una Máquina Virtual (en su desktop, hospedada en Azure, en algún servidor de la organización, etc.). Además, existirá otra granja específica para Integración Continua y QA.

Sin embargo, con la llegada de SharePoint Online, este modelo no es todo lo válido que nos gustaría, y tenemos que buscar otras opciones. Mi recomendación pasa por algo parecido a la siguiente imagen, que explicare justo debajo de ella:

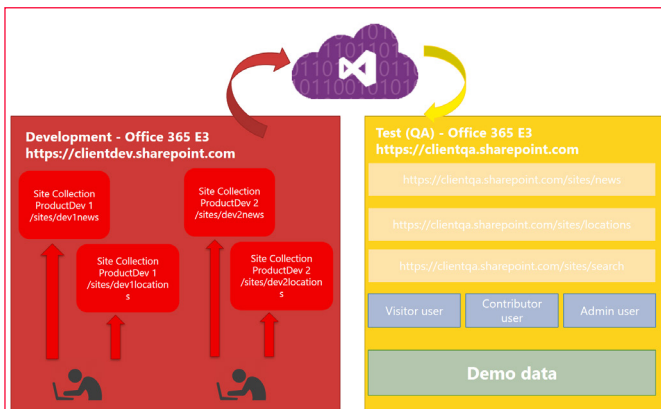


Imagen 1.- Modelo de desarrollo para SharePoint Online.

Para el entorno de desarrollo, contaremos con una Tenant de Office 365, en un plan que sea suficiente para el proyecto que estamos desarrollando (E3, etc.). Dentro de esa

tenant, asignaremos una Site Collection específica a cada Developer, pudiendo incluso crear una Site Collection para cada Developer-Modulo/Producto, en el caso de que nuestro proyecto sea muy grande y este dividido en diferentes Modulos o Productos, como, por ejemplo, un Módulo específico para Noticias, o para Búsqueda, o para Localizaciones, etc. Haciendo uso de herramientas como Gulp, podemos hacer fácilmente que cada desarrollador, “apunte” a la Site Collection propia, y los cambios de un desarrollador, no afecten a los de otro. Para esta tenant de desarrollo, podemos contar únicamente con una licencia válida, que compartirán todos los desarrolladores.

Para el entorno de QA, tendremos una tenant diferente de Office 365. En este caso recomiendo que el plan contratado sea el mismo que tendrá la tenant de Producción. Además, como será el entorno para el equipo de QA, lo aconsejable será tener distintas licencias/usuarios, para que al menos se puedan probar los diferentes roles que necesita nuestro proyecto (al menos 3, para poder hacer pruebas con un usuario Visitor, Contributor y Admin). En la tenant de QA, crearemos una Site Collection para cada módulo: Raiz, News, Offices... lo que necesite nuestro proyecto. Además, es muy importante que consigamos un juego de datos muy similar al que nos encontraremos en producción, ya que será vital para las pruebas. Para esto, el propio PnP puede ser de utilidad, ya que las plantillas pueden contener datos de prueba para insertar ítems en listas, archivos, etc.

veremos cómo podemos hacer uso del Framework del PnP junto a Visual Studio Team Services

Finalmente, el paso de desarrollo a QA, estará totalmente automatizado usando una Build de Team Services. Así, la build estará programada para que, o bien cada cierto tiempo (i.e: diaria), o bien en cada commit al Control de código, se ejecute la build, actualizando así el entorno de QA con los últimos cambios desarrollados.

Configurando la Build de CI para SharePoint Online

Nota: Este artículo asume cierta familiarización con el

Framework PnP, de no ser así, puedes leer alguno de mis artículos previos referentes al PnP en la web de Comparti-MOSS. Además, nada de esto es válido para proyectos SharePoint en modo Farm Solutions.

Llegados a este punto, veamos cómo podemos configurar una Build en Team services que despliegue nuestra solución en una tenant de SharePoint Online. Antes de continuar, para este ejemplo y las imágenes a continuación, podéis descargar el siguiente proyecto desde mi repositorio de GitHub: <https://github.com/luismanez/sp-saturday-barcelona-2016>.

Antes de nada, lo primero que necesitamos en la solución, es un proyecto de consola que utilice el Framework del PnP para desplegar nuestra plantilla o plantillas. Esta acción es muy sencilla usando el PnP, y aquí tenéis el snippet de código para hacerlo:

```
//get ClientContext
AuthenticationManager authManager = new AuthenticationManager();
ClientContext context =
    authManager.GetSharePointOnlineAuthenticatedContextTenant(url, user, password);

//Load template from XMLFileSystemTemplateProvider
XMLFileSystemTemplateProvider provider = new XMLFileSystemTemplateProvider(distFolder, "");

var template = provider.GetTemplate(templateName);
template.Connector = new FileSystemConnector(distFolder, "");

//Apply template
context.Web.ApplyProvisioningTemplate(template);
```

Llamaremos a esta aplicación de consola desde nuestra build, para lanzar el proceso de provisioning remoto a través del PnP.

Nota: Lo idea en este punto, sería utilizar los propios comandos PowerShell del PnP. Sin embargo, dichos comandos necesitan una instalación (o registro manual) dentro de la máquina. Desafortunadamente, en el momento de escribir este artículo, no parece que haya manera de hacer algo así en la máquina donde se ejecuta el Agente cloud, por lo que necesitamos ejecutar nuestra propia aplicación para aplicar la plantilla PnP.

Creando una nueva Build en Team Services

Primero de todo, al pulsar en el botón de Nueva Build, tenemos disponibles diferentes plantillas, según el tipo de proyecto que tengamos. Inicialmente, trabajaremos con la plantilla de Visual Studio.

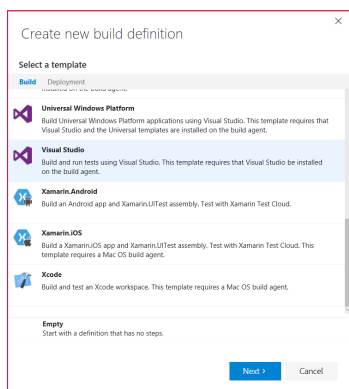


Imagen 2.- Creación de una definición de build.

puedes leer alguno de mis artículos previos referentes al PnP en la web de Comparti-MOSS

Compilando nuestra solución de VS

Nuestro siguiente paso, será elegir con que control de código estamos trabajando (GitHub, VSTS, Subversion...), así como la rama por defecto que utilizaremos. Además, seleccionaremos el tipo de Agente que queremos utilizar. El Agente es donde toda la magia sucede, y tenemos 2 opciones, podemos usar un servidor local como Agente (este será el caso de trabajar con SharePoint On Premises), o bien usar un Agente hospedado en el mismo Team Services (esto se conoce con el nombre de Cloud Agent, ya que como TS, correrá en la nube).

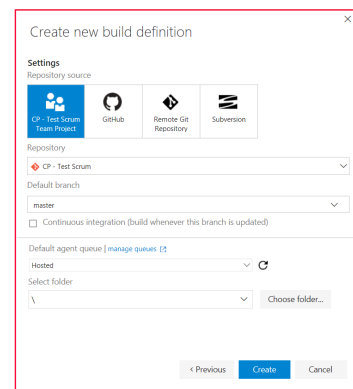


Imagen 3.- Selección del gestor de código fuente.

Una vez creada la Build, nuestro siguiente paso será configurar las diferentes acciones a ejecutar.

Nuestra primera acción, y que vendrá configurada por defecto por haber elegido la plantilla de VS, será la de restaurar todos los paquetes Nuget necesarios en nuestra Solución.

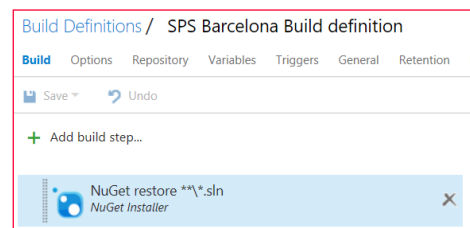


Imagen 4.- Restauración de los paquetes de NuGet.

Dicha tarea no requiere ninguna configuración adicional a la que trae ya por defecto (básicamente le decimos que archivo de tipo VS Solution queremos utilizar). El siguiente Build step, será simplemente compilar todos los proyectos de nuestra solución. Para ello, la plantilla también nos ha configurado un Step de Build Solution.

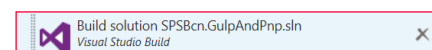


Imagen 5.- Step de Build Solution generado en la plantilla.

En nuestro caso, le especificamos la Solution que quere-

mos utilizar:

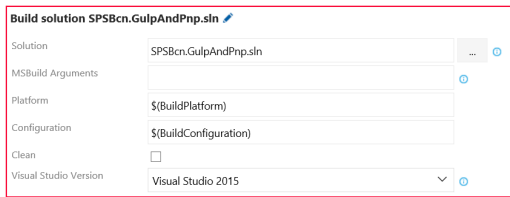


Imagen 6.- Configuración de la solución a utilizar.

Ejecutando Tareas de Gulp

El siguiente paso, será ejecutar ciertas tareas Gulp que tenemos en nuestro proyecto. Os aconsejo que reviséis el código del ejemplo utilizado en el artículo. Allí veréis que hacemos uso de Gulp para ciertas tareas, como por ejemplo, concatenar todos los ficheros JS en un único .min.js, y además, comprimirlo para mejorar el rendimiento. Además, tenemos otras tareas Gulp, como aplicar reemplazos de tokens en las plantillas XML del PnP, compilar SASS a CSS, etc.

Por lo tanto, el primer paso para usar Gulp en nuestra Build, será restaurar los paquetes requeridos por Gulp. Para ello, usaremos el Step de NPM



Imagen 7.- Restauración de los paquetes de Gulp.

Y configuraremos el Step para que apunte a nuestro proyecto (Working directory) donde hacemos uso de Gulp:

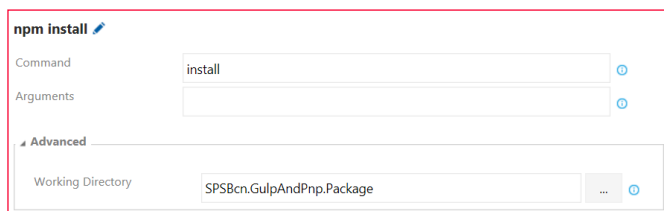


Imagen 8.- Configuración del Step para hacer uso de Gulp.

Una vez restaurados los paquetes que necesita Gulp, crearemos un nuevo Step de Gulp:

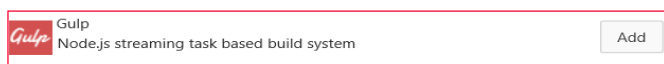


Imagen 9.- Creación del nuevo Step de Gulp.

Y lo configuraremos para ejecutar la tarea de Gulp que queremos, así como la ubicación del proyecto que se requiere de Gulp:

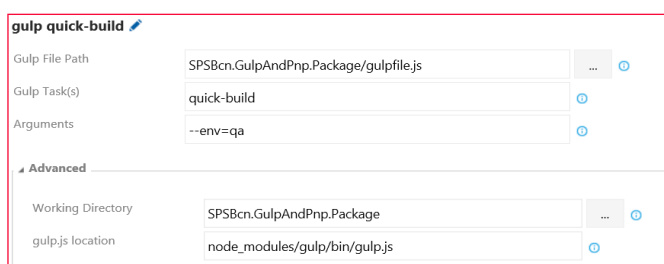


Imagen 10.- Configuración del Step de Gulp.

Remote Provisioning usando el PnP

El último paso, será utilizar nuestra aplicación de consola para lanzar el Provisioning remoto de nuestra plantilla PnP. Para ello, utilizaremos el Build Step del Command Line:

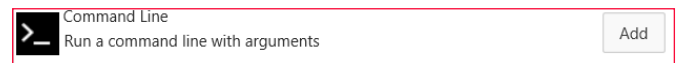


Imagen 11.- Uso del Build Step de Command Line.

Y lo configuraremos para usar la aplicación de consola de nuestra solution, además de pasar varios argumentos necesarios para la consola:

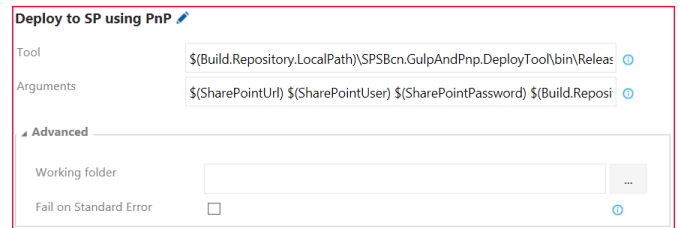


Imagen 12.- Configuración del Step de Command Line

Donde:

- Tool: Es la ubicación de la aplicación de Consola ya compilada en un paso previo. Aquí podemos hacer uso de diferentes Tokens que nos proporciona Team Services. En este caso, necesitaremos:

```
$(Build.Repository.LocalPath)\SPSBcn.GulpAndPnp.DeployTool\bin\Release\SPSBcn.GulpAndPnp.DeployTool.exe
```

- Arguments: Diferentes parámetros que necesita la aplicación de consola. En nuestro caso, necesitamos los datos de conexión a SharePoint: Url, Usuario, Password, así como la ubicación del fichero XML con la plantilla PnP.

```
$(SharePointUrl) $(SharePointUser) $(SharePointPassword) $(Build.Repository.LocalPath)\SPSBcn.GulpAndPnp.Package\dist\src\pnp\ProvisioningTemplate.Package.xml
```

En este caso, además de hacer uso de los Tokens predefinidos en Team Services, hacemos uso de Variables definidas a nivel de Build. Podemos configurarlas desde la Tab de Variables:

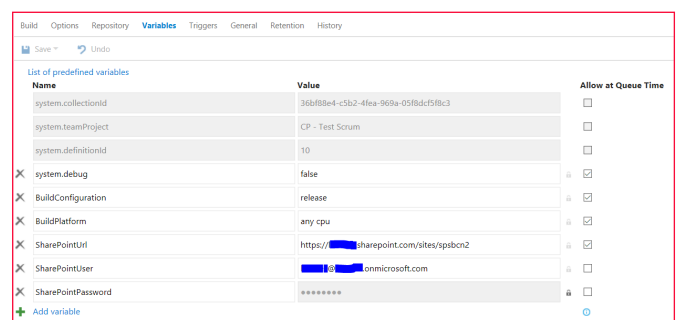


Imagen 13.- Parámetros requeridos en la aplicación de consola.

Fijaros como podemos marcar una variable como tipo

“Password” (icono del candado), de tal forma que el valor no aparece en pantalla, pero Team Services lo descifrá y enviará en la ejecución del Step.

Build Triggers

Como último paso, configuraremos de qué modo queremos ejecutar la Build. Para ello iremos a la tab de Triggers, y elegiremos la opción que mejor se adapte a nuestras necesidades:

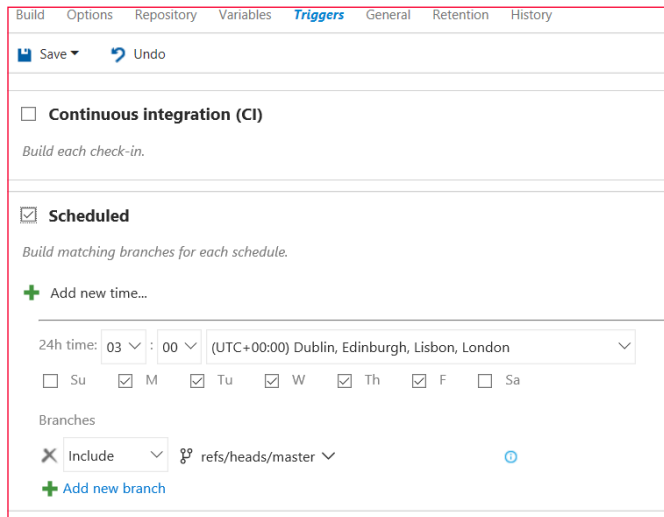


Imagen 14.- Configuración de la ejecución de la build.

Con estos sencillos pasos, y siempre haciendo uso del fra-

mework del PnP, podemos tener rápidamente configurado un entorno de Integración Continua para proyectos de SharePoint Online.

Seguimiento del programa PnP

Recordad las siguientes maneras de estar al día con el programa PnP, ya que ha habido cambios respecto a lo publicado en artículos anteriores:

- MS Tech Community: Pretende reemplazar la red de Yammer previa. <https://techcommunity.microsoft.com/t5/SharePoint-Developer/bd-p/SharePointDev>
- Canal PnP en YouTube: Reemplazando también el canal existente en Channel 9. https://www.youtube.com/channel/UC_mKdhw-V6CeCM7gTo_ly7w
- Blog del PnP: <http://dev.office.com/blogs>
- Community call: <https://github.com/OfficeDev/PnP/wiki/Community-call>
Apunta esta cita en tu calendario. Cada mes el equipo de PnP hace un meeting donde informa del estado del programa, el roadmap, muestra ejemplos, etc.

LUIS MAÑEZ

SharePoint / Cloud Solutions Architect en ClearPeople LTD

@luismanez

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

Power BI Embedded

En el presente artículo vamos a presentar una solución de Power BI Embedded con la finalidad de dejar claro de qué se trata este nuevo servicio que se ofrece sobre la plataforma Azure. En primer lugar, definamos cual es el público objetivo de utilización de este servicio. Soluciones y aplicaciones que deseen mostrar reportabilidad son las que justamente podrán explotar este servicio, con la funcionalidad principal de mostrar reportes en aplicaciones web.

Cuáles son los requerimientos para configurar una solución utilizando Power BI Embedded:

- Aprovisionar un nuevo Workspace Collection en Azure.
- Generar un nuevo Workspace en Worspace Collection previamente creado.
- Importar un reporte de Power BI Desktop con formato PBIX en un Workspace existente.
- Utilizar la siguiente información e incorporar la misma en el Web Config de nuestra aplicación Web:
 - Access Key.
 - ApiURL.
 - Workspace Collection.
 - Workspace ID.

A continuación, haremos un recorrido sobre cada uno de los puntos anteriormente mencionados para guiar los mismos en una solución web.

Creación de Workspace Collection

En primer lugar, debemos ingresar al portal de Azure a través de: <https://portal.azure.com>. En este caso vamos a generar un nuevo recurso, y seleccionamos la opción de Power BI Embedded. Debemos definir datos tales como el nombre de Workspace Collection Name, Subscripción, Grupo de Recurso, Location, y Pricing.

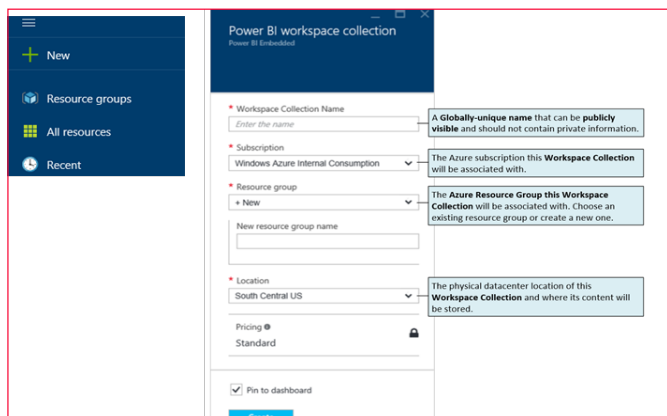


Imagen 1.- Generación del recurso de Power BI Embedded.

vamos a presentar una solución de Power BI Embedded con la finalidad de dejar claro de qué se trata este nuevo servicio

Al generar un nuevo Workspace Collection deberemos tener presente dado que serán utilizados en nuestra solución web de Visual Studio datos de dicho elemento tales como: WorkSpace Collection Name, Access Keys.

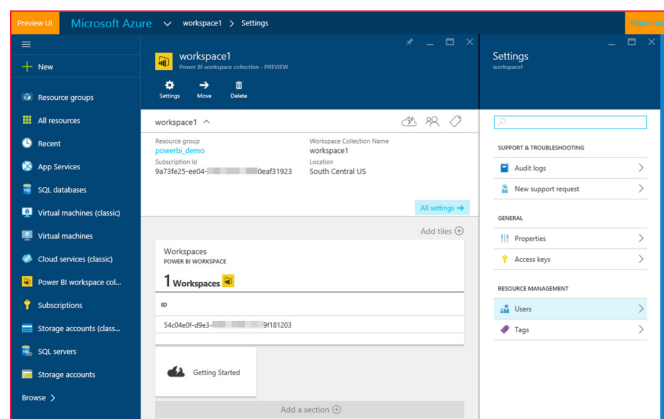


Imagen 2.- Parámetros importantes del Workspace.

Access Keys

Las claves de acceso para el Workspace las obtendremos al hacer clic en la opción AccessKey:

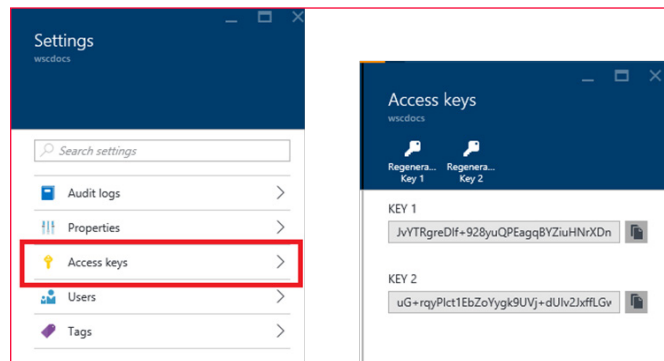


Imagen 3.- Claves de acceso generadas.

Definición o Aprovisionamiento de Workspace

En el siguiente paso lo que vamos a definir es la generación

de un Workspace que nos va a proveer de un contenedor de soluciones de Reportería de Power BI. Por un tema de gobernabilidad es buena idea generar un Workspace que pertenezca al Workspace Collection que anteriormente definimos.

logramos embeber un reporte de Power BI en una aplicación web

Con esta pequeña sintaxis en una aplicación de Consola se define un Workspace:

```
var workspace = await
CreateWorkspace(workspaceCollectionName);
workspaceId = workspace.WorkspaceId;
Console.ForegroundColor = ConsoleColor.Cyan;
Console.WriteLine("Workspace ID: {0}", workspaceId);
```

Importación de Reporte de Power BI Desktop

En la secuencia que estamos manejando, ahora es momento de tomar el archivo con el reporte que hayamos generado en Power BI Desktop. En este paso lo que estamos logrando es importar un archivo de formato PBIX a nuestro WorkSpace, esto permitirá embeber finalmente nuestro reporte en una aplicación Web.

```
if (string.IsNullOrEmpty(workspaceCollectionName))
{
    Console.WriteLine("Workspace Collection Name:");
    workspaceCollectionName = Console.ReadLine();
    Console.WriteLine();
}

if (string.IsNullOrEmpty(workspaceId))
{
    Console.WriteLine("Workspace ID:");
    workspaceId = Console.ReadLine();
    Console.WriteLine();
}

Console.WriteLine("Dataset Name:");
var datasetName = Console.ReadLine();
Console.WriteLine();

Console.WriteLine("File Path:");
var filePath = Console.ReadLine();
Console.WriteLine();

var import = await ImportPbix(workspaceCollectionName,
workspaceId, datasetName, filePath);
Console.ForegroundColor = ConsoleColor.Cyan;
Console.WriteLine("Import: {0}", import.Id);
```

Embebiendo en Aplicación Web

Luego de que hayamos generado la Aplicación Web en Visual Studio tal como hacemos con cualquier aplicación Web lo que deberemos configurar es el Web Config de dicho aplicativo.

Mediante el siguiente código lograremos setear correctamente la información que se requiere para nuestro reporte:

```
<appSettings>
  <add key="powerbi:AccessKey" value="O+U8vwfS/nXpT/
KjT3m+D/JcPEUQSVWIPoATENI/TmIJBvZ55Xq5ig4bEB/
fngc/3tyzleT/kOYE7hLUL7LrA==" />
  <add key="powerbi:ApiUrl" value="https://api.powerbi.com" />
  <add key="powerbi:WorkspaceCollection"
value="GastontPoweBI" />
  <add key="powerbi:WorkspaceId" value="8d7dcf4f-34c3-
4158-a220-b5050d2b7a2c" />
</appSettings>
```

Como vemos en la información anterior los datos necesarios para dicho Web.config son:

- Access Key – ver en Portal Azure.
- Api URL – dirección/URL de Power BI.
- WorkspaceCollection – Nombre del WorkspaceCollection creado anteriormente (también visible en nuestro Portal Web de Azure).
- WorkSpaceId – Este requerimiento es el ID del Workspace que deseamos embeber. También es un dato que podemos extraer de nuestro portal de Azure.

Conclusiones

De esta manera logramos embeber un reporte de Power BI en una aplicación web. Al compilar nuestra aplicación contaremos con todo el backend de la aplicación, y además la posibilidad para el usuario de visualizar todos los reportes generados:

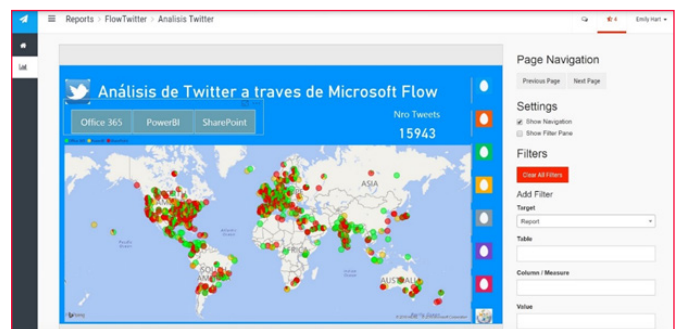


Imagen 4.- Informe Power BI embebido.

GASTÓN CRUZ

Business Intelligence Tecnical Manager | Arkano Software

Prepara la migración desde tus granjas OnPremises a SharePoint Online con SMAT

La herramienta SharePoint Migration Assessment Tool (SMAT) es una utilidad de línea de comandos que escanea los contenidos y personalizaciones de una Granja de SharePoint 2013 con el objetivo de identificar posibles problemas que se pueden dar, así como elementos no soportados en un escenario de migración a SharePoint Online. SMAT realiza un escaneo con impacto mínimo sobre la Granja y genera una serie de informes detallados sobre los posibles impactos en una migración potencial a SharePoint Online.

Configuración y Uso de SMAT

La configuración y uso de SMAT en una Granja de SharePoint 2013 es realmente sencilla:

- Lo primero que haremos es descargar SMAT desde este enlace. La descarga de SMAT consiste en un archivo .zip que contiene los siguientes elementos:

Microsoft.Applications.Telemetry.dll	9/20/2016 4:25 PM	Application extens...	318 KB
Microsoft.Applications.Telemetry.Server.dll	9/20/2016 4:25 PM	Application extens...	21 KB
Microsoft.Bond.dll	9/20/2016 4:25 PM	Application extens...	140 KB
Microsoft.Bond.Interfaces.dll	9/20/2016 4:25 PM	Application extens...	18 KB
Microsoft.Sharepoint.Migration.Scan.dll	9/20/2016 4:25 PM	Application extens...	981 KB
scandef	9/19/2016 11:59 AM	JSON File	6 KB
SiteSkipList	8/18/2016 12:34 PM	Microsoft Excel C...	1 KB
SMAT Read Me	9/22/2016 1:23 PM	Microsoft Word D...	24 KB
SMAT	9/20/2016 4:25 PM	Application	32 KB
SMAT.exe	9/19/2016 11:59 AM	CONFIG File	1 KB
SMAT	10/16/2016 10:36 ...	ZIP File	515 KB

Imagen 1.- Contenido del archivo .zip de SMAT.

- Una serie de ensamblados que son utilizados por el ejecutable de SMAT.
- Dos archivos de configuración que son modificables. Uno de ellos, scandef.json, permite habilitar o deshabilitar configuraciones particulares de escaneo mediante SMAT como por ejemplo habilitar / deshabilitar el escaneo de listas grandes en la Granja modificando la sección del archivo que se incluye en el siguiente fragmento de código.

```
{
  "Name": "LargeListViews",
  "Type": "LargeListViewsScanner",
  "SupportedVersions": [
    "2013",
    "2016"
  ],
  "ReportCategoryType": "SPSite",
  "Property": {
    "ListItemThreshold": 3000,
    "LookupColumnThreshold": 12
  },
  "Enabled": false
}
```

Microsoft liberó una herramienta denominada SharePoint Migration Assessment Tool (SMAT)

El otro archivo, SkipSitesList.csv, permite indicar que sitios de la granja no van a ser escaneados por SMAT. Este archivo tiene la estructura que se muestra en la Figura 2.

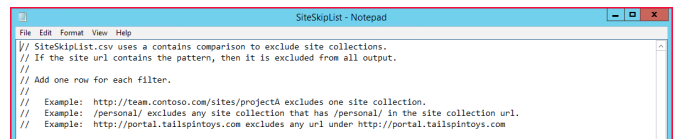


Figura 2.- Estructura del archivo SiteSkipList.csv.

- Un archivo tipo Readme con la descripción de SMAT y cómo utilizarla.
- Los ejecutables de SMAT.
- Para comenzar a utilizar SMAT, creamos en primer lugar una carpeta en la que almacenar los archivos de log generados por SMAT. A continuación, tenemos que abrir el símbolo del sistema y acceder a la ubicación de la herramienta para ejecutar SMAT de la siguiente forma:

```
SMAT.exe [-o] <SMAT_LOGS_FOLDER> [-t] <NUMBER_PARALLEL_THREADS> [-sv]
```

- El parámetro -o indica la carpeta dónde generar los archivos de log e informes generados por SMAT. En el caso en el que la carpeta ya contenga archivos de ambos tipos, serán sobre-escritos en cada nueva ejecución de SMAT.
- El parámetro -t indica el máximo número de hilos en los que SMAT se ejecutará en paralelo. El valor mínimo de este parámetro 1 y el máximo el total de procesadores de la máquina dónde se ejecute SMAT.
- Finalmente, el parámetro -sv permite saltarse algunos chequeos previos realizados por SMAT.
- Supongamos que ejecutamos SMAT configurando los

parámetros -o y -t que admite. A continuación, SMAT empieza a realizar su trabajo recopilando la información relativa a nuestra granja e identificando posibles problemas para una potencial migración a SharePoint Online.

el siguiente artículo de soporte ([Nota: Además se indican las pautas para remediar estos problemas en un escenario real de migración](https://support.office.com/en-us/article/Migration-Assessment-Scan-Reports-Roadmap-5bf2619b-14fd-421c-8b84-869743f1eca2?ui=en-US&rs=en-US&ad=US)):

<https://support.office.com/en-us/article/Migration-Assessment-Scan-Reports-Roadmap-5bf2619b-14fd-421c-8b84-869743f1eca2?ui=en-US&rs=en-US&ad=US>

Por citar algunos, SMAT presenta informes relativos a Add-Ins desplegados e instalados en la Granja, páginas personalizadas, Soluciones de tipo Full Trust, Listas con más de 20.000 elementos, etc.

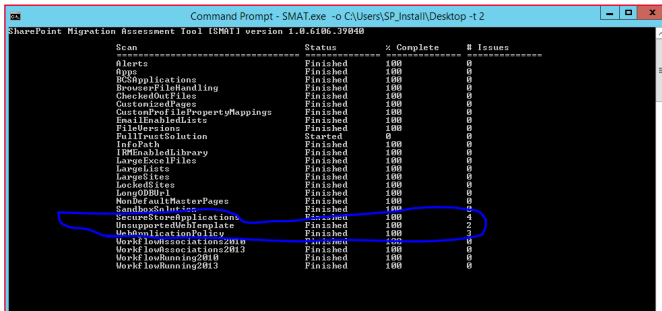


Imagen 3.- Ejemplo de ejecución de SMAT en una granja de SharePoint 2013.

- En mi escenario particular la ejecución de SMAT ha encontrado potenciales problemas con Web Templates no soportadas, aplicaciones en el Secure Store Service y Políticas de Aplicación Web. Además, SMAT nos permite indicar el dominio de nuestro tenant de SPO y con esto concluirá la ejecución de SMAT.
- Como resultado de la ejecución de SMAT, se generan los siguientes archivos y carpetas:

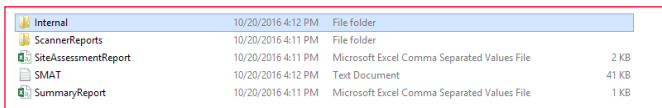


Imagen 4.- Archivos y carpetas generados por la ejecución de SMAT en la Granja.

- Un archivo SMAT.txt con todas las operaciones de escaneo realizado.
- Dos archivos CSV globales denominados SummaryReport.csv y SiteAssessmentReport.csv. El primero contiene un resumen de los resultados de la ejecución de SMAT en la Granja identificando las características en distintos ámbitos (Granja, Aplicación Web, Colección de Sitios) que causarán problemas en la migración. Y el segundo un resumen detallado por cada sitio de los problemas encontrados como muestra la Imagen 4.

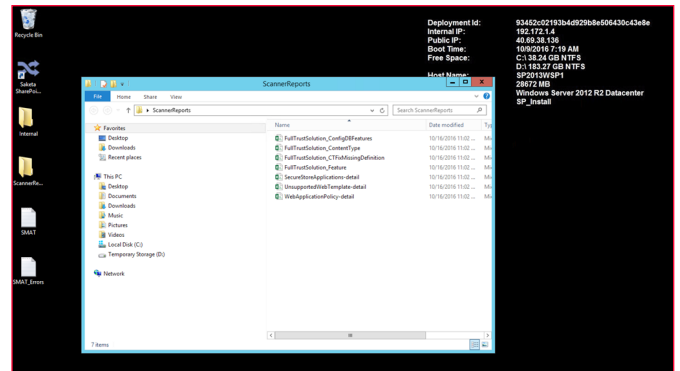


Imagen 6.- Resultado de la ejecución de SMAT en la Granja de SharePoint 2013.

- Un conjunto de informes detallados almacenados en la carpeta Internal y consistentes en archivos .txt y .csv con el detalle de las Soluciones de Tipo Full Trust encontradas por SMAT. Por ejemplo, los archivos WebParts.csv y WebParts.txt contienen información relativa a WebParts desplegadas en la Granja: nombre del WebPart, característica que despliega el WebPart, Solución Full Trust.

SMAT es una herramienta gratuita desarrollada por Microsoft que permite escanear una Granja completada de SharePoint 2013

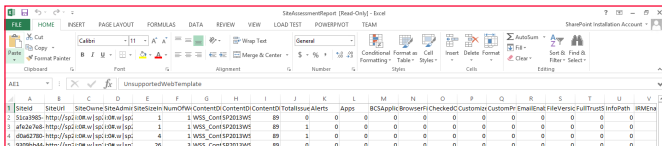


Imagen 5.- Informe SiteAssessmentReport.csv.

es una utilidad de línea de comandos que escanea los contenidos y personalizaciones de una Granja de SharePoint 2013

- Una serie de informes almacenados en la carpeta ScannerReports con los distintos problemas potenciales que nos podemos encontrar en una migración. El detalle de los distintos tipos de informes que se pueden generar y su contenido lo tenéis en

Assemblies	10/20/2016 4:11 PM	Text Document	1 KB
ContentTypes	10/20/2016 4:11 PM	Microsoft Excel C...	2 KB
ContentTypes	10/20/2016 4:11 PM	Text Document	1 KB
CustomFields	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
CustomWorkflowActivities	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
DependencyFiles	10/20/2016 4:11 PM	Microsoft Excel C...	23 KB
EventHandlers	10/20/2016 4:11 PM	Text Document	0 KB
EventReceivers	10/20/2016 4:11 PM	Microsoft Excel C...	3 KB
Exception_SolutionAnalyzer	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
FeatureIds-AllFeatures	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
FeatureIds-DeployingContentTypes	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
FeatureReceivers	10/20/2016 4:11 PM	Text Document	1 KB
Features	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
FieldTypes	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
FTCRresult	10/20/2016 4:11 PM	XML File	257 KB
ListInstances	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
ListTemplates	10/20/2016 4:11 PM	Microsoft Excel C...	2 KB
ListTemplates	10/20/2016 4:11 PM	Text Document	1 KB
MasterPages	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
MasterPages	10/20/2016 4:11 PM	Text Document	0 KB
PageLayouts	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
PageLayouts	10/20/2016 4:11 PM	Text Document	0 KB
Pages	10/20/2016 4:11 PM	Microsoft Excel C...	1 KB
SiteColumns	10/20/2016 4:11 PM	Text Document	1 KB
WebParts	10/20/2016 4:11 PM	Microsoft Excel C...	3 KB
WebParts	10/20/2016 4:11 PM	Text Document	1 KB

Imagen 7.- Archivos .csv y .txt contenidos en la carpeta Internal de SMAT.

• Como muestra de informes generados, a continuación, os dejo dos capturas de pantalla relativas a:

- Por un lado, al informe UnsupportedWebTemplate-detail.csv que contiene aquellos sitios que han sido creados con plantillas no soportadas en SPO (en mi caso Centro de Inteligencia Empresarial y MySite). Como podéis ver, para cada Sitio detectado se muestra información relativa al Id del Sitio, Url del Sitio, Propietario, # de Subsitos que contiene, etc.

SiteId	SiteName	SiteAdmin	SiteGroup	NumOfWebContent	ContentID	ContentIDFull	WebSite	WebTitle	WebTemp	WebTemplate	ScoreId
0362250	http://sp2013hsp3/Inte/BCenter	0362250	0362250	4	1	WSS_Content0362250	89 http://sp2013hsp3/Inte/BCenter	1000	BCENTERSITE	03730768-2605-4330-8646-4f42d1	
4362626	http://sp2013hsp3/My	0362250	0362250	1	1	WSS_Content0362250	89 http://sp2013hsp3/My Site	54	SPOSMSTESTHOST	03730768-2605-4330-8646-4f42d1	

Imagen 8.- Informe UnsupportedWebTemplate-detail.csv generado por SMAT.

- Y por otro el informe SecureStoreApplications-detail.csv que contiene el listado de las Aplicaciones creadas en la Aplicación de Servicio de Almacenamiento Seguro y cuya migración a SharePoint Online no está soportada.

ne no está soportada.

ApplicationId	Name	FriendlyName	ContactEmail	ApplicationType	CredentialNameTicketTime
0362250	0362250	0362250	0362250	0362250	0362250
0362250	0362250	0362250	0362250	0362250	0362250
0362250	0362250	0362250	0362250	0362250	0362250
0362250	0362250	0362250	0362250	0362250	0362250

Imagen 9.- Informe SecureStoreApplications-detail.csv.

Conclusiones

SMAT es una herramienta gratuita desarrollada por Microsoft que permite escanear una Granja completada de SharePoint 2013 para facilitar encontrar situaciones problemáticas e incompatibilidades en un escenario de migración desde SharePoint 2013 OnPremises a SharePoint Online en Office 365.

JUAN CARLOS GONZÁLEZ MARTÍN

Office Servers and Services MVP

Cloud & Productivity Advisor

jcgonzalezmartin1978@hotmail.com

[@jcg1978](https://www.linkedin.com/in/jcg1978)

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



SharePoint Framework: El Despliegue

Introducción

El nuevo SharePoint Framework, “SPFX” de aquí en adelante, supone un cambio brutal en cuanto a la forma de desarrollar sobre nuestro servidor preferido. Uno de los puntos más críticos tradicionalmente en SharePoint ha sido llevar una ALM de forma correcta. Ahora con el nuevo SPFX se han facilitado las cosas, de una forma un tanto diferente, facilitando bastante dicho ciclo. Este proceso quedaría de la siguiente forma:

- 1.- Crear plantilla del SPFX con Yeoman.
- 2.- Desarrollo de los WebParts en local utilizando NodeJS.
- 3.- Testeo y pruebas en local.
- 4.- Generación de los CDN (lugar donde va a estar ubicado el JavaScript).
- 5.- Generación de un fichero .sapp e instalarlo en el catálogo de aplicación.
- 6.- Testeo y pruebas en el entorno de producción.

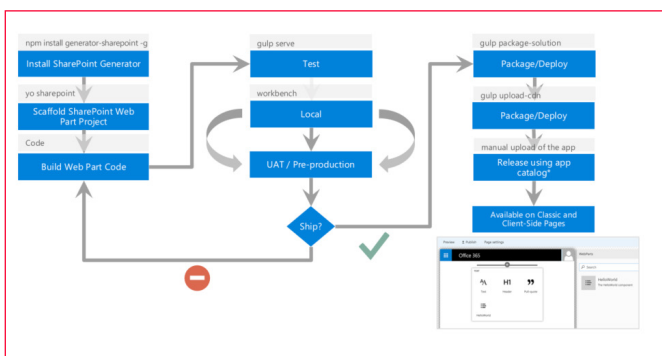


Imagen 1 Proceso de desarrollo en el SPFX.

En este artículo vamos a centrarnos en el proceso del despliegue y que es lo que debemos de conocer dentro de la solución para poder pasar nuestro desarrollo de un entorno NodeJS a un entorno de Office 365 con el menor número de inconvenientes posibles.

El empaquetado de la solución

Una vez se ha terminado el desarrollo, pasamos a crear el fichero .sapp para ubicarlo en el catálogo de aplicaciones. Para ello disponemos de una tarea de Gulp que se encarga de este proceso. La instrucción es la siguiente:

```
$ gulp package-solution
```

Una vez termina este proceso, tendremos una carpeta “sharepoint” dentro de nuestra solución. En esta carpeta tendrá un fichero “nombredelasolucion”.sapp.

¿Que contiene este fichero?

El contenido de este fichero es:

- AppManifest.xml, en el que está el Identificador de esta App, así como las propiedades del mismo.
- Un xml con el Identificador de la característica dentro de SharePoint junto con un fichero con el mismo nombre, pero con la extensión .config. Dentro de estos ficheros tendremos las características básicas de la característica: Title, Description, ID, Scope, Web.
- Una carpeta con el identificador de la característica en la que dentro de ella hay un fichero .WebPart y .xml. En el fichero .WebPart se dispone todas las características propias del WebPart implementado. El fichero .xml sustituye a los antiguos “Modules” e indica la información de donde se va a ubicar el WebPart y que es el contenido del mismo.

se liberó en “Preview de desarrollo” el tan esperado nuevo Framework de desarrollo sobre SharePoint

Una vez, subimos este fichero a nuestro catálogo de aplicaciones, lo instalamos y lo desplegamos en una página., la primera vez no nos funcionará. ¿Por qué? El motivo es que se están buscando el/los ficheros js resultantes en un directorio local. En los tutoriales iniciales que mostró el equipo de producto indicaba que si levantábamos nuestro servidor NodeJS en local y arrancábamos nuestra solución visualizábamos nuestro desarrollo. Esta acción puede vernos en determinadas circunstancias en las que necesitamos ver que está ocurriendo, pero mi opinión es que si necesitamos realizar esto para depurar/ probar nuestro desarrollo por lo general en algo nos estamos equivocando.

¿Dónde dejó los ficheros JS para utilizarlos en Office 365/SharePoint?

El equipo de producto ha intentado atender a todas las sugerencias que les ha realizado la Comunidad, indicándole que el tiempo de descarga de un fichero JS en SharePoint Online era demasiado elevado e inaceptable en muchas ocasiones. Por este motivo la solución que ha propuesto es que estos ficheros estén alojados en un CDN (Common Delivery Network), que es un punto de acceso público cuya función es servir estos servicios de la forma más rápida posible.

Ahora bien, esto tiene un “pequeño” problema. Estamos hablando de que toda la lógica de nuestros desarrollos está en un sitio “público” y esto puede provocar ataques y fugas de conocimiento en productos/proyectos realizados sobre esta nueva plataforma. Para solucionarlo Microsoft lo que ha optado es por incluir un CDN dentro de tu Tenant de Office 365 de tal forma que este solo pueda ser accedido desde el propio tenant y no responda a petición que no sean de tu tenant. Este proceso también está en Preview y aún no sabemos cómo será este proceso.

En la versión del SPFX que hay ahora, está preparada para poder alojar el desarrollo dentro de un CDN de Azure. No es objeto de este artículo ver cómo crear un CDN en Azure, pero por si alguien necesita ayuda basta con seguir los pasos del siguiente artículo: <https://dev.office.com/sharepoint/docs/spfx/web-parts/get-started/deploy-web-part-to-cdn>

```
$ gulp deploy-azure-storage
```

Lo que sí que es objeto de este artículo es ver cómo debemos de configurar nuestra solución para cuando generemos el paquete tenga todas las rutas bien puestas y configuradas. Para ello el equipo de producto lo que ha creado dentro del template de Yeoman es una carpeta “config” el sentido de la misma es que dentro de la misma agrupemos todos los ficheros de configuración de las tareas de Gulp. Por defecto, SPFX ya tiene una tarea Gulp que se encarga de subir nuestro WebParts a los CDN de Azure. Para ello vamos al fichero `deploy-azure-storage.json` ubicado en dicha carpeta y configuramos los datos del mismo. Si todos los datos los hemos introducido correctamente ejecutamos la siguiente instrucción y en unos segundos el bundle en el CDN de Azure:

En caso de que no utilices el CDN de Azure, puedes utilizar el que prefiera, SPFX es agnóstico tanto a la hora de la elección de framework que utilizar, como entorno de desarrollo y cloud que utilices. Eso sí, en caso de utilizar otros CDN se deberá de implementar dicha tarea.

Una vez tenemos ya tenemos el JavaScript ubicado en el CDN, todavía nos queda un paso más y es indicarle en el proceso de generación de paquete cual es la ruta donde se encuentran JS, ya que como hemos podido ver por defecto sino se indica nada los buscará en la máquina de desarrollo.

Para indicarle al spapp la ubicación de los ficheros, debemos de abrir el fichero `write-manifest.json` ubicado en la

carpeta “config” y configurarle la ruta de donde tenemos nuestros ficheros JS.

¿Dónde está ubicado los ficheros de los frameworks JavaScript de nuestro desarrollo?

Por poner un poco de contexto, la solución de Yeoman está basada en WebPack. Es una herramienta para ayudarnos a empaquetar nuestro código JavaScript. Por decirlo de una forma sencilla, se encarga de unificar todos nuestros ficheros en un único fichero. De esta forma conseguimos tener menos tamaño de descarga de los ficheros JS y de esta forma nuestro desarrollo puede ser más rápido. Ahora bien, dentro de este código no se incluye por defecto las librerías de terceros. Tenemos dos opciones:

- Utilizar la instrucción `require` que nos indica que ese fichero lo tiene que cargar, y lo agrega en el bundle
- Invocar al CDN del propio framework en nuestro desarrollo.

¿Aún queda algo que tengamos que tener en cuenta?

Hemos cargado los ficheros JS y CSS dentro de nuestro bundle pero todavía nos queda ver cómo resolvemos la carga de los templates HTML que muchos Frameworks JavaScript utilizan. La técnica puede ser algo similar a la carga de los CDN de los Frameworks JS, podemos incluirlos dentro del bundle junto con el resto de JS y CSS utilizando `require` o bien ubicarlos en un CDN de forma manual. Para más detalles de las formas de carga de los templates dentro del SPFX este artículo de Waldek es de obligatoria lectura <https://blog.mastykarz.nl/load-angular-templates-sharepoint-framework-client-side-web-parts/>

Conclusión

Esta nueva forma de desarrollo o el nuevo Framework de desarrollo abre un abanico importante a personalizar SharePoint de una forma estable y sin temor a un problema de actualización. Es un cambio que todos los usuarios de SharePoint estábamos esperando. Pero este cambio conlleva muchos cambios y el principal es entender de una forma correcta el nuevo Tooling de desarrollo y sacarle todo su partido. Como toda novedad tiene una curva de aprendizaje, pero una vez se pasa esta curva estoy seguro que os va a gustar mucho.

ADRIÁN DIAZ CERVERA

Architect Software Lead at Encamina
MVP Office Server and Services

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

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

adiaz@encamina.com

@AdrianDiaz81



Enterprise Mobility Suite Parte 1 – Introducción

Enterprise Mobility Suite (EMS), Es un conjunto de productos hospedados por Microsoft para un mejor precio. Individualmente, la suma del coste de estos servicios asciende a casi 3 veces el coste de tener todo incluido, haciéndolo más atractivo para ir con todo el paquete. Para una buena ejecución de la estrategia de movilidad completa de su cliente, ayudando así a los usuarios a ser productivos en los dispositivos y proteger los activos corporativos.

Enterprise Mobility Suite (EMS), Es un conjunto de productos hospedado por Microsoft

Gartner ha publicado este año 2016 el Cuadrante Mágico de Gestión de la Movilidad empresarial (EMM) y, en base a la integridad de visión y capacidad de ejecución en el mercado de la movilidad empresarial, que han posicionado a Microsoft en el Cuadrante Visionario.



Imagen 1.- Cuadrante mágico de Gartner para EMM.

La suite consta de los siguientes productos:

- 1.- Azure Active Directory Premium (AAD Premium) es una solución de administración de identidades y acceso con un conjunto de funcionalidades que tienen como objetivo proteger el acceso a las aplicaciones locales y en la nube, así como ayudar

a mantener la seguridad de los datos corporativos. Este servicio incluye entre otras características:

- Inicio de sesión único (SSO): Puedes gestionar y unificar identidades entre múltiples plataformas, dispositivos y servicios en la nube (o en local) para lograr un inicio de sesión único. Esto permite que los empleados utilicen una única credencial de acceso, corporativa y segura para acceder a aplicaciones OnPremise y servicios Cloud. Post de la configuración del AD Connet.
- Autoservicio: Ofrece un portal de autoservicio a través del cual los propios usuarios que olviden una contraseña o bloqueen una cuenta, pueden restablecerlas a partir de una pregunta de seguridad o un código enviado a su dispositivo móvil.
- Autenticación Multifactor (MFA): Integra una funcionalidad que ayuda a prevenir el acceso no autorizado a datos en la nube a través de una segunda autenticación mediante una App móvil, una llamada telefónica automatizada o un mensaje de texto.

Gartner publicó su Cuadrante Mágico de Gestión de Identidad y Acceso como Servicio (IDaaS) para 2016 y Azure Active Directory se colocó en el cuadrante de "líderes", y se coloca muy fuertemente en la integridad de la visión. Copias gratuitas disponibles se pueden acceder aquí .



Imagen 2.- Cuadrante mágico de Gartner para IDaaS.

2.- Azure Rights Management (Azure RMS) Ayuda a datos confidenciales seguros y archivos donde quiera que vayan. Con esta funcionalidad, se adjunta un certificado a cada archivo que tenga un correo electrónico o en SharePoint Online para limitar el acceso al archivo(s) de la persona(s) en el correo electrónico e incluso impedir la impresión del archivo.

- Crear políticas de seguridad: Tanto de identidad y autorización, tales como requerir la contraseña de usuario corporativo para trabajar con el documento, no permitir capturas de pantalla y muchas otras todas destinadas a evitar el acceso a usuarios no autorizados y asegurar el cumplimiento de las políticas de la empresa.
- Establecer directivas: Para proteger tus archivos (SharePoint Online) y correo electrónico (Exchange Online) en cualquier dispositivo y con las que podrás aplicar restricciones tales como no permitir el copiar, cortar y pegar, así como protección frente a la impresión de documentos o la descarga de archivos.
- Define reglas de transporte: Para cifrar mensajes de correo electrónico saliente y descifrar respuestas cifradas entrantes o para evitar que un destinatario externo a la organización pueda realizar alguna de estas funciones: acceder y reenviar el contenido de un correo electrónico.

3.- Microsoft Advanced Threat Analytics Ayuda a identificar las infracciones y las amenazas utilizando análisis del comportamiento y proporciona un informe claro y aplicable en una simple línea de tiempo de ataque.

Microsoft Advanced Threat Analytics Ayuda a identificar las infracciones y las amenazas

4.- Microsoft Intune proporciona gestión de dispositivos móviles, gestión de aplicaciones móviles, y las capacidades de gestión de la PC en la nube. Usando Intune, las organizaciones pueden proporcionar a sus empleados con acceso a las aplicaciones corporativas, datos y recursos desde prácticamente cualquier lugar en casi cualquier dispositivo, mientras que ayuda a mantener la información corporativa segura.

- Dispositivos autorizados: Puedes gestionar todo tipo de dispositivos (iOS, Android, WP y PC's) así como establecer políticas de seguridad para los que pueden acceder a la información corporativa. Con esta función, tanto emails como documentos de todo tipo sólo pueden ser utilizados por dispositivos autorizados por la compañía e incluso es posible bloquear el uso de conexiones USB.
- Borrado selectivo de datos: Con esta función, si un

dispositivo se pierde, ha sido robado o si el empleado ya no forma parte de la organización, los administradores pueden borrar los datos corporativos que había dentro del dispositivo, pudiendo opcionalmente hacerlo sin eliminar los datos personales del usuario.

- Despliegue automatizado de aplicaciones: Permite instalar de manera automatizada nuevas aplicaciones y actualizaciones de las mismas, así como de los sistemas operativos.
- 5.- Microsoft Identity Manager (MIM) Ofrece en las instalaciones de gestión de identidad y acceso. Permite sincronizar identidades entre diferentes directorios, bases de datos y aplicaciones. MIM ofrece la posibilidad de dar acceso privilegiado en la forma de acceso Just-In-Time, que permite a los usuarios adquirir los privilegios necesarios para sus tareas al tiempo que limita el día para el acceso de la cuenta.
- 6.- Microsoft Cloud App Security Este componente proporciona una visibilidad más profunda de la seguridad de sus aplicaciones en la nube con controles completos y una mayor protección.

Como se puede observar, el paquete es muy amplio y añade una gran cantidad de funcionalidad en torno a la gestión de la seguridad y la identidad. La mayoría de los clientes sólo van por un par de estos productos a la vez, pero una vez que el adquiera toda la suite, la implementación de la funcionalidad sólo añade valor a la experiencia del usuario final, manteniendo la seguridad.

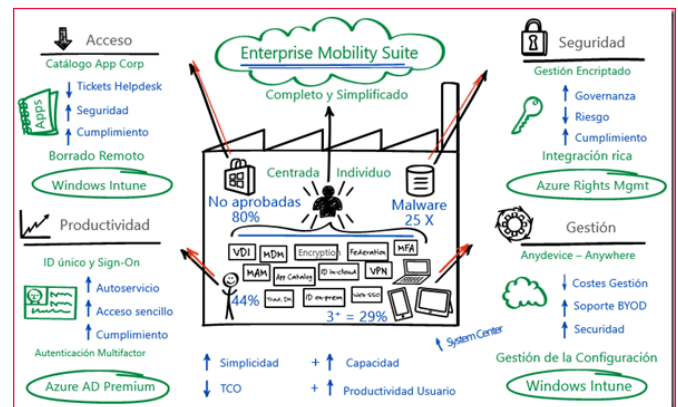


Imagen 3.- Overview de los elementos y capacidades de EMS.

Conclusión

EMS es el servicio para proteger, prevenir fuga de información y tener control sobre la documentación de la empresa o de los dispositivos.

JORGE CASTAÑEDA
MVP Office Servers & Services
jcastaneda@asr.pe
@xorxe
http://www.asr.pe

GUÍA PRÁCTICA LÍMITES DE USO Y TAMAÑOS DE SHAREPOINT

LEYENDAS

- Límite recomendado. Si se sobrepasa, se puede penalizar el rendimiento y la experiencia de usuario en el uso de las
- Límite máximo. Ni tú IT seréis capaces de sobrepasarlo. Así que ni siquiera lo intentes.
- SharePoint 2013
- SharePoint Online/2016

ADMINISTRACIÓN DEL SITIO

COLECCIONES DE SITIOS

250K	100 GB	1 GB	25 TB
SITIOS POR COLECCIÓN	POR COLECCIÓN	POR COLECCIÓN para empezar	POR COLECCIÓN Tamaño máximo

PÁGINAS

25
ELEMENTOS WEB

EMPIEZA A TRABAJAR

LAS BIBLIOTECAS

30M	5K
ARCHIVOS	ELEMENTOS POR VISTA
	Incluye carpetas y todo tipo de archivos

TAMAÑO DE ARCHIVOS

250 MB	2 GB	10 GB
EN TAMAÑO DE ARCHIVOS	EN TAMAÑO DE ARCHIVOS	EN TAMAÑO DE ARCHIVOS
Límite estándar	Tamaño máximo	

NOMBRES DE ARCHIVOS Y CARPETAS

128 CAR	256 CAR	260 CAR
NOMBRE DE ARCHIVO	NOMBRE DE CARPETA	RUTA DE ARCHIVO + CARPETA

HISTORIAL DE VERSIONES

400K	511
VERSIONES MAYORES	VERSIONES MENORES

EDITORES

10	99
AUTORES EN SIMULTÁNEO	AUTORES EN SIMULTÁNEO

NOMBRES DE ARCHIVOS Y CARPETAS

256 CAR	256 CAR	260 CAR
NOMBRE DE ARCHIVO	NOMBRE DE CARPETA	RUTA DE ARCHIVO + CARPETA

LÍMITES DE LISTAS

30M	5K
ELEMENTOS	ELEMENTOS POR VISTA

LÍMITES DE ARCHIVOS ADJUNTOS

250 MB	2 GB	250 MB
EN TAMAÑO DEL ADJUNTO	EN TAMAÑO DEL ADJUNTO	EN TAMAÑO DEL ADJUNTO

HISTORIAL DE VERSIONES

400K	0*
VERSIONES MAYORES	VERSIONES MENORES

*Las versiones menores sólo están disponibles en las bibliotecas de documentos. Prueba, en este caso, con la aprobación de contenido.

PARA COMPARTIR

BLOGS

5K	1K
POSTS POR BLOG	COMENTARIOS POR POST

SEGUIR

400K
FOLLOWS POR USUARIO

SUMINISTRO DE NOTICIAS

512	100	300x300
CARACTERES POR POST	RESPUESTAS POR POST	ES EL TAMAÑO EN PÍXELES DE LAS IMÁGENES

BIBLIOTECAS

LISTAS

Guía práctica sobre límites de uso y tamaños de SharePoint

Hay algo que todos los usuarios de SharePoint deben tener en cuenta: los límites de la herramienta que están utilizando. Algunos de ellos pueden ser realmente molestos, pero la mayoría no te darán problemas...al menos en gran medida.

Eso es así hasta que descubres su existencia justo en el momento en que estás terminando un proyecto, o quizás cuando lo quieres compartir o cuando necesitas enviarlo. Si conoces estas limitaciones de antemano, puedes evitarte un mal rato en el futuro. En caso en el que te encuentres con uno de estos límites en una situación importante, seguramente lo recordarás por siempre. Pero es mejor seguir la regla de oro: revisa bien antes de darle uso. SharePoint no es infinito.

Permíteme repetirlo:



Imagen 1.- SharePoint no es infinito.

Cada sistema tiene sus propios límites, pero, en general, los estándares son similares o incluso los mismos. Tu departamento de IT quizás pueda cambiar algunos de estos valores; de todos modos, debes asegurarte de que sabes bien todo lo que está implementado, para que así no descubras –otra vez, demasiado tarde– que los límites que menciono aquí no son realmente los que se debieron establecer para ti.

(Muchos de los límites pueden ser restringidos más adelante por tu equipo de IT, y por lo tanto, no se pueden expandir– ten en cuenta que si hacen algún cambio, generalmente no será a tu favor).

Microsoft ofrece una lista en profundidad de sus límites (<https://technet.microsoft.com/es-es/es/library/cc262787.aspx>), pero es un recurso que puede llegar a agobiar. Además, casi todo el artículo es irrelevante para los usuarios habituales de la herramienta. Es este el motivo por el que decidí hacer una selección de aquellos límites que pueden afectar el día a día de un usuario de la plataforma.

Quizás el día en que te topes de frente con un límite, no

sea tan traumático, pues recordarás las perlas de sabiduría que viste en este post. Porque, por supuesto, ¡tú eres un SharePointer muy bueno y consciente!

Un último recordatorio:



Imagen 2.- Advertencia sobre el origen de la información mostrada en el artículo.

Tipos de límites

Microsoft define tres tipos de límites (<https://technet.microsoft.com/es-es/es/library/cc262787.aspx#Boundaries>) en SharePoint 2013 y SharePoint Online:

- **Boundaries:** Static limits that cannot be exceeded by design
- **Thresholds:** Configurable limits that can be exceeded to accommodate specific requirements
- **Supported limits:** Configurable limits that have been set by default to a tested value

Imagen 3.- Tipos de límites definidos por Microsoft para SharePoint 2013 y SharePoint Online.

- **Límites máximos:** son límites estáticos que no se deben exceder en el diseño.
- **Umbrales:** son límites configurables que se pueden sobrepasar para atender requerimientos específicos.
- **Límites admitidos:** son límites configurables que se establecieron por defecto en base a valores ya probados.

Estoy en desacuerdo–de forma respetuosa, creo–con la necesidad de hacer estas tres definiciones, aunque los umbrales y los límites admitidos son prácticamente lo mismo. Aquí está mi interpretación de esos conceptos:

- **Límites máximos:** Estos son límites que no pueden cambiar ni tú ni el propietario del sitio ni el equipo de IT. El hecho de que sea más o menos estricto depende de qué tipo de sistema SharePoint fue implementado por el departamento de IT:
 - **SharePoint On-Premises 2013:** Si tu compañía usa SharePoint 2013 en sus propios servidores, entonces las revisiones y actualizaciones de la plataforma pueden producir mejoras en los límites. Pero, generalmente, una actualización completa de tu sistema (por ejemplo, de SP2010 a SP2013) es ne-

cesaría para producir cambios positivos en relación con esas limitaciones.

- **SharePoint Online:** Si usas SharePoint Online en Office 365, se pueden conseguir mejoras en los límites cuando Microsoft decida hacerlo. Es algo así como cuando tienes una cuenta en Gmail: el límite se incrementa cuando Google quiere ser competitivo al ofrecer más espacio a sus usuarios. Así que sonrío, pues podrías llegar un lunes en la mañana a la oficina y sorprenderte con una buena cuota de espacio adicional.
- **Umbrales y límites admitidos:** ambos son límites sugeridos que se pueden cambiar o sobrepasar, pero que Microsoft recomienda no modificar en base a los resultados de las baterías de pruebas realizados sobre la plataforma. ¿Quieres romper un límite? Entonces, tendrás que asumir que puedes penalizar el rendimiento de tu Granja. Los límites se basan en la habilidad de SharePoint para desarrollar una determinada acción: por ejemplo, puedes subir un archivo de 2GB —a SharePoint no le va a importar—, pero es posible que se tarde mucho en cargarlo, y quizás el motor de búsquedas dé un error durante el proceso; por eso, el límite se establece en un sentido práctico, y muchas veces es algo que ni el propio SharePoint puede controlar. También se pueden establecer estos límites para controlar los problemas de rendimiento desde el lado del usuario: tener millones de archivos en una biblioteca afectará la rapidez de carga cada vez que intentes acceder a la misma.

A veces pienso que las recomendaciones de Microsoft no tienen mucho que ver con las pruebas, sino con su capacidad para solucionar esas situaciones por sus propios medios. Bueno, ¿sabes qué? Redmond prefiere estar seguro que lamentarlo después. Esos límites los protegen de cualquier acción legal que quieran emprender los clientes. Por supuesto, eso no quiere decir que Microsoft esté a salvo de demandas (https://en.wikipedia.org/wiki/United_States_v._Microsoft_Corp.).

SharePoint 2013 vs. SharePoint Online

En primer lugar, si no tienes clara la diferencia entre SharePoint y Office 365, puedes revisar mi artículo anterior aquí (<http://icansharepoint.com/explained-sharepoint-versus-office-365/>). Con seguridad, te ayudará a entender cómo estos límites pueden afectarte.

En caso de que tu organización disponga de una infraestructura SharePoint 2013 propia, las limitaciones que experimentes serán estáticas mientras dure su vida útil o hasta que el sistema se actualice a SharePoint 2016. También puedes contar con estos límites hasta que migres a SharePoint Online como parte de una migración/actualización a Office 365. SharePoint se actualiza de forma frecuente, generalmente cada trimestre, y tu departamento de IT

también puede hacerlo sin que te enteres. Desafortunadamente, esas mejoras escalonadas no te darán todos los beneficios que puedes obtener con una actualización de la versión completa.

De todos modos, SharePoint Online—que es gestionado y administrado por Microsoft, se beneficiará en ocasiones de las actualizaciones de rendimiento y características a medida que Microsoft mejora el producto y despliega releases más pequeñas y segmentadas que las que habitualmente se realizan cada 3 años para SharePoint On-Premises (SharePoint 2007, 2010, 2013, 2016. ¿Te das cuenta del patrón?).

Eso significa que si tienes Office 365 con SharePoint Online y OneDrive para Empresa, podrás ver las mejoras gracias a las actualizaciones de funciones que se hacen de forma más periódica; así, no tendrás que esperar a que el departamento de IT actualice SharePoint. Aunque te parezca adecuado actualizar una plataforma como SharePoint cada 3 años, en muchas ocasiones ocurre que la actualización puede llevar dos ciclos de producto: todo depende del equipo de Tecnología. Por ejemplo, tu compañía puede estar trabajando con SharePoint 2007 por tanto tiempo que en IT decidan saltarse SharePoint 2010 e ir directamente a SharePoint 2013. Lo más seguro es que sea bien recibido por los usuarios, pero significaría estar estancado con la versión de 2007 por mucho tiempo.

SharePoint Online es una mejor opción cuando se trata de obtener mejoras en las funciones, pero también significa que tendrás que estar al día de los cambios que se publican en los blogs de Microsoft y SharePoint. Así podrás saber todo sobre esas novedades, y también sobre la degradación o eliminación de alguna función, que quizás esté sucediendo justo bajo tus pies.

Sitios y colecciones de sitios

Todos los sitios de SharePoint forman parte de una familia de sitios, denominada colección de sitios. Usualmente, el departamento de IT es el que crea las colecciones de sitios. Los sitios hijos, nietos, bisnietos, etc., que se encuentran dentro de esa colección (piensa que son como las ramas de un único árbol), muchas veces son creados por el equipo de IT y otras, por cualquier usuario que lo necesite. Todo depende de las reglas que establezca tu departamento de IT.



Imagen 4.- Sitios vs. Colecciones de Sitios.

El departamento de IT puede crear una gran cantidad de

colecciones de sitios. Generalmente, designan una para cada organización de tu compañía (por ejemplo, departamento, oficina geográfica, ese tipo de cosas). Planificar esto con tiempo es un buen ejemplo arquitectura de información (AI) inteligente, algo que, aunque no esté entre tus pensamientos cotidianos, realmente incide en la forma en la que buscas lo que necesitas para hacer tu trabajo. Una arquitectura de información eficiente es aquella que no se nota, simplemente porque tiene sentido.

todos los usuarios de SharePoint deben tener en cuenta: los límites de la herramienta que están utilizando

Las colecciones de sitios y los sitios son dos cosas diferentes, y funcionan de manera distinta. Por lo tanto, sus límites varían.

Microsoft recomienda que las colecciones de sitios no deben:

- 1.- Contener más de 250.000 sitios (límite admitido), algo que, por cierto, es un número alucinantemente grande.
- 2.- Contener más de:
 - 100 GB de información (una cantidad relativamente pequeña, en comparación con los sitios que se pueden admitir) en SharePoint 2013; o
 - 1 TB de información en SharePoint Online (la cantidad estándar empieza en 1 GB, así que debes solicitar un incremento a tu administrador de IT).
- 3.- Tener más de 2 millones de usuarios (límite admitido), lo que es, otra vez, un número gigantesco (y probablemente sólo sea importante para IT, porque ellos controlan quiénes son esos usuarios).
- 4.- Tener más de 5.000 usuarios en un grupo de SharePoint (límite admitido), algo que, en realidad, se contradice con el elemento anterior; y
- 5.- Utilizar más de 10.000 grupos de SharePoint.

Estos no son límites máximos. Si decides ir más allá, puedes hacerlo, pero el rendimiento y las operaciones de respaldo podrían afectarse de forma negativa, como explica Redmond.

No se mencionan de forma explícita los límites para sitios individuales, excepto que las páginas de SharePoint están limitadas a 25 elementos web por página (umbral). Esto me sorprendió porque he creado páginas que probablemente se acerquen a ese límite. Definitivamente, hay casos en los que pueden estar justificados los 25 elementos web. Así que ten esto en cuenta si eres propenso a usar muchos elementos web en tus páginas de SharePoint.

Algo que hay que tener en cuenta es que los sitios comparan el tamaño permitido para toda la colección. Eso significa que un sitio extremadamente pesado puede “robar” espacio a otros que estén dentro de la misma colección.

He visto que esto ocurre en la práctica y créeme que no es nada bonito. Especialmente porque, muchas veces, los propietarios de los sitios “que compiten” ni siquiera saben sobre la existencia del otro hasta que se dan cuenta de que está acaparando espacio. No seas esa persona.

Sólo hay dos soluciones para este problema: 1) borrar contenido, o 2) mover alguno de los sitios (sea el agresor o la víctima). Ambas opciones son terribles. Muchísimo. Nadie quiere revisar contenido y borrarlo. Y mover los datos puede ser una verdadera pesadilla si no tienes una herramienta costosa para hacer la migración. E, incluso, si la tuvieras, puede seguir siendo complicado.

Bibliotecas de documentos

Una biblioteca de documentos es, a los ojos de SharePoint, como una gran carpeta. Contiene el historial de versiones, los metadatos personalizados (por ejemplo, las categorías y otros elementos similares), la estructura de la carpeta y otras herramientas. Aquí probablemente invertirás la mayor parte de tu tiempo, pues se trata de la elaboración, edición y eliminación de archivos que se publican en SharePoint. El dueño de un sitio puede crear un montón de bibliotecas en un sitio determinado.

Microsoft recomienda que:

- 1.- Las bibliotecas de documentos no contengan más de 30.000.000 documentos (<https://technet.microsoft.com/en-us/library/cc262787.aspx>) (límite admitido). Se trata de una cantidad extremadamente grande. Puedes obtener una capacidad mayor, pero Microsoft no lo aconseja.
- 2.- Por defecto, el límite de tamaño de los archivos se establece así:
 - SharePoint 2013 (<https://technet.microsoft.com/en-us/library/cc262787.aspx#ListLibrary>): 250 MB (límite admitido), pero tu departamento de IT puede incrementar esa cantidad hasta 2GB (límite máximo). Ese valor de 2GB es el máximo absoluto. A diferencia del límite admitido de 30.000.000, sugerido en el punto anterior, el tamaño de 2GB puede ser muy limitante en ciertas ocasiones, especialmente cuando hablamos de elementos multimedia y videos.
 - SharePoint Online (<https://blogs.office.com/2016/02/17/auditing-reporting-and-storage-improvements-for-sharepoint-online-and-onedrive-for-business/>): 10 GB (límite máximo). Ahí lo tienes.

Los nombres de los archivos y las carpetas tienen los siguientes límites de caracteres:

- 1.- Los nombres de los archivos no pueden superar los 128 caracteres ([https://technet.microsoft.com/es-es/library/ff919564\(v=office.14\).aspx](https://technet.microsoft.com/es-es/library/ff919564(v=office.14).aspx)) en SP2013 (límite máximo) y 256 caracteres en SPO (límite máximo).
- 2.- Los nombres de las carpetas no pueden tener más de 256 caracteres (<https://technet.microsoft.com/>

[es-es/es/library/ff919564\(v=office.14\).aspx](https://es-es/es/library/ff919564(v=office.14).aspx)).

- 3.- Y algo importante: la combinación de la ruta de la carpeta más el nombre del archivo se limita a 260 caracteres ([https://technet.microsoft.com/es-es/es/library/ff919564\(v=office.14\).aspx](https://technet.microsoft.com/es-es/es/library/ff919564(v=office.14).aspx)). Todos estos son límites máximos. ¡Ten mucho cuidado con las carpetas anidadas! Puede traerte verdaderos problemas en el futuro.

(Nota: la fuente de este límite se basa en un artículo de SP 2010, pero, después de una investigación, parece que no hubo cambios en SP2013/SPO).

¿Y por qué este último asunto es importante? Porque a la gente le encantan las carpetas. Por alguna razón, un árbol interminable de carpetas parece dar calor a los corazones de muchas personas. Y eso puede llevarte a tener un muy mal día. ¿Por qué?

Piensa en este ejemplo: un árbol de carpetas que tiene la siguiente estructura

- Presupuesto Anual del Departamento de Finanzas para 2016
 - Documentación de apoyo para el Presupuesto 2016
 - Borrador inicial del Presupuesto 2016 para revisión del Comité Ejecutivo
 - Documentos de borrador firmados del Presupuesto 2016 para compartir internamente
 - Comentarios iniciales de Mary sobre el Presupuesto
 - DocumentoInicialRevisionParaFirmarPosteriorRevisionCFO-Mary.docx

Esto no abrirá de forma correcta en tu navegador, así busques en los niveles más profundos. En total, se sobrepasa el límite de 260 caracteres.

Límites máximos: son límites estáticos que no se deben exceder en el diseño

Si crees que esa estructura de carpetas es una locura, está bien. Ahora mira a tus compañeros. La mayoría de ellos no lo piensa así. Créeme. Esto pasa todo el tiempo.

Ahora, el historial de versiones permite (<https://technet.microsoft.com/es-es/es/library/cc262787.aspx#ListLibrary>) hasta 400.000 versiones mayores (límite admitido) y 511 versiones menores (límite máximo). Ambos números son altísimos. Puedes leer mi artículo completo sobre el historial de versiones en este post del blog (<http://icansharepoint.com/the-absolute-basics-of-version-history/>).

Cuando haya coautores de un documento, Microsoft recomienda (<https://technet.microsoft.com/es-es/es/library/cc262787.aspx#ListLibrary>) que no más de 10 autores trabajen en simultáneo (límite umbral) en un mismo documento

de Word o en una presentación de PowerPoint. De todos modos, se puede cambiar ese valor para que 99 personas trabajen al mismo tiempo (límite máximo). Microsoft señala que el rendimiento se puede convertir en un problema. Pero, es evidente que se trata de un asunto de logística: imagina que una docena de personas editen un documento a la vez. Si no afrontas conflictos en plena edición, sería un verdadero milagro.

Por último, las vistas en SharePoint están limitadas a 5.000 elementos (https://support.office.com/en-us/article/Manage-large-lists-and-libraries-in-SharePoint-2013-and-SharePoint-2016-b8588dae-9387-48c2-9248-c24122f07c59?ui=en-US&rs=en-US&ad=US#_toc330996252). Eso incluye archivos y carpetas. Eso quiere decir que puedes tener 30.000.000 documentos en una biblioteca, pero no puedes mostrar más de 5.000 de ellos (o las carpetas en las que están alojados) en una vista. Este es, decididamente, el talón de Aquiles de SharePoint, conocido como el Umbral de Vista de Lista. Y si no sabes nada sobre eso, asegúrate de buscarlo en Google. Internet estará feliz de explicarte ese pequeño inconveniente. Parece que no es gran cosa, pero puede afectarte en algún momento.

Herramientas para compartir en SharePoint

Las fotos de perfil de Delve (Office 365) se cargan mejor si tienen una resolución de 96x96 píxeles (<https://support.office.com/en-us/article/View-and-edit-your-profile-in-Office-Delve-4e84343b-eedf-45a1-aeb9-8627ccca14ba?ui=en-US&rs=en-US&ad=US>). SP2013 sugiere 300x300 píxeles, pero hay un problema conocido (<https://spsawyer.wordpress.com/2013/07/31/sharepoint-2013-user-profile-photo-sizes/>) con estas imágenes: se ven borrosas. De todas formas, hacer un recorte cuadrado siempre será la mejor alternativa.

Los posts del suministro de noticias no pueden sobrepasar los 512 caracteres (<https://technet.microsoft.com/es-es/es/library/jj219700.aspx>) (no es ni Twitter ni Facebook). Las imágenes deben ser de 300 x 300 píxeles (<https://social.technet.microsoft.com/Forums/office/en-US/17ea5516-aae3-488c-9be7-12acb608625a/newsfeed-image-size?forum=sharepointgeneral>), así se comience con un tamaño más grande. Cada post puede tener un máximo de 100 respuestas (<https://technet.microsoft.com/es-es/es/library/jj219700.aspx>).

Algo de interés: si usas un blog, hay un límite de 5.000 posts por blog y 1.000 comentarios por post. Los documentos (<https://technet.microsoft.com/en-us/library/cc262787.aspx#Blog>) dicen que este es el límite admitido, así que, técnicamente, puede ser superior. Pero el uso del término “número máximo” en la descripción me hace pensar que realmente es un límite máximo. No se dice nada sobre si este asunto tiene un impacto similar en tu blog Mi Sitio. Asumo que sí, pero no hay nada explícito al respecto. Para saber más sobre hacer blogs en el trabajo –y por qué es algo bueno para ti y para tus empleados–, lee mi post anterior (<http://icansharepoint.com/why-blogging-at-work-makes-sense/>).

Adicionalmente, puedes seguir (<https://technet.microsoft.com/en-us/library/cc262787.aspx#Velocity>) hasta 400.000 “cosas” en SharePoint, sea a un compañero, un documento, un sitio, un elemento de una lista, etc. Eso es muchísimo. Si te preocupa la posibilidad de sobrepasar esa cantidad, quizás es momento de replantearte la forma en la que trabajas.

Tamaño de archivos en Office 365 Video

Desde febrero de 2016, se pueden subir videos de hasta 10GB (<https://blogs.office.com/2016/02/17/auditing-reporting-and-storage-improvements-for-sharepoint-online-and-onedrive-for-business/>) en el portal de Office 365. El portal de video es, esencialmente, una biblioteca más grande, por lo que funciona bajo las mismas reglas que las bibliotecas de documentos, al menos en su mayor parte.

De forma notable, el techo de 10GB es 5 veces mayor que el límite máximo de 2GB que se mencionó anteriormente. Por cierto, es un tamaño realmente pequeño para archivos de video, pero me doy cuenta de que SharePoint no se preocupa por saber si el archivo es un elemento multimedia que consume mucho espacio, o si es un típico documento pequeño.

Conclusión

SharePoint puede ser muy útil para un sinnúmero de cosas. Pero, como cualquier otro producto, tiene sus limitaciones. Y, a veces, esos límites son muy molestos, especialmente cuando estás acostumbrado a otras herramientas menos restrictivas, como Google Drive y DropBox.

En ese sentido, los límites no te molestarán si sabes de qué se tratan o, al menos, te dedicas a revisarlos antes de hacer algo tonto, como crear una estructura profunda con 15 carpetas, para incluir un solo documento que sabes que debe ir ahí. Pero, un momento, ya eso no me incumbe.

Para terminar, algo muy importante: ¿Me falta algo? ¿Presenté un dato incorrecto? ¡Dímelo! Lo arreglaré. Te lo prometo. Contáctame a través de mi cuenta de Twitter @icansharepoint.

Nota del autor

Este artículo ha sido traducido a partir del artículo original en inglés del blog del autor.

MATT WADE

Digital workforce expert and consultant

thatmattwade@gmail.com

@thatmattwade

<http://icansharepoint.com/>



Mentoring

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.



DocumentDB Query's

DocumentDB es el servicio NoSQL que nos proporciona Azure como servicio PaaS. Sabemos que las bases de datos NoSQL no tienen esquema y en el caso de DocumentDB guardan su información en formato JSON.

Entonces, si no tenemos esquemas y los datos se guardan en formato JSON ¿Cómo puedo consultar los datos que tengo? La respuesta es mediante lenguaje SQL (Structured Query Language).

En este artículo veremos que sentencias SQL pueden utilizarse y cómo hacerlo.

Esquema de una sentencia SQL en DocumentDB

```
SELECT [TOP <top_expression>] <select_list>
[FROM <from_specification>]
[WHERE <filter_condition>]
[ORDER BY <sort_specification>]
```

Como se puede observar el esquema para realizar consultas es muy familiar si se conoce SQL. Cada sentencia consiste en un SELECT acompañado opcionalmente por un FROM, WHERE o ORDER BY. Veamos un ejemplo.

Tenemos la siguiente información en nuestra collection Familias de DocumentDB:

```
{
  «id»: «FamiliaCompartiMOSS»,
  «padres»: [
    {
      «apellido»: «CompartiMOSS»,
      «nombre»: «Revista»
    },
    {
      «apellido»: «Apellido1»,
      «nombre»: «Nombre1»
    }
  ],
  «hijos»: [
    {
      «apellido»: «Apellido2»,
      «nombre»: «Nombre2»,
      «sexo»: «mujer»,
      «grado»: 1,
      «mascotas»: [
        {«nombre»: «Goofy»},
        {«nombre»: «Mickey»}
      ]
    },
    {
      «apellido»: «Apellido3»,
      «nombre»: «Nombre3»,

```

```
{
  «sexo»: «mujer»,
  «grado»: 8
},
{
  «direccion»: {
    «pais»: «España»,
    «ciudad»: «BCN»
  },
  «contacto»: {
    «email»: «email@email.com»,
    «telefono»: «000000000»
  },
  «registrado»: false
}
```

Y realizamos la siguiente consulta:

```
SELECT {
  «Nombre»: «f.id»,
  «Ciudad»: «f.direccion.ciudad»
} AS Familia FROM Familias f WHERE f.direccion.ciudad = "BCN"
```

El resultado será:

```
[
  {
    «Familia»: {
      «Name»: «FamiliaCompartiMOSS»,
      «Ciudad»: «BCN»
    }
  }
]
```

Aunque no tengamos un esquema fijo podemos ver cómo podemos realizar consultas utilizando la potencia de SQL. Ahora vamos a ver más en detalle todas las opciones.

DocumentDB es el servicio NoSQL que nos proporciona Azure como servicio PaaS

FROM

Clausula opcional, aunque si se aplica un filtro en la query o se proyecta algún campo como resultado, entonces será obligatorio definir esta cláusula. Como en una consulta SQL convencional, la cláusula FROM indica la collection sobre la cual se ejecutará la consulta, también puede identificar un subconjunto de la collection como fuente de la consulta.

Un ejemplo de utilización de un subconjunto de la collection sería:



```
SELECT *
FROM Familias.hijos
```

El resultado será:

```
[
  {
    «apellido»: «Apellido2»,
    «nombre»: «Nombre2»,
    «sexo»: «mujer»,
    «grado»: 1
  },
  {
    «apellido»: «Apellido3»,
    «nombre»: «Nombre3»,
    «sexo»: «mujer»,
    «grado»: 8
  }
]
```

También se puede utilizar alias en las consultas de la forma convencional: `SELECT f.id FROM Familias AS f`. Una vez se utiliza el alias la fuente ya no se reconocerá, es decir, si realizáramos la consulta `SELECT Familias id FROM Familias AS f` la query no funcionaría, como tampoco funcionaría `SELECT id FROM Familias AS f`.

Por último, también se puede utilizar la cláusula `IN`, que nos permite realizar iteraciones sobre JSON arrays. La cláusula `FROM` nos da soporte para iteraciones. Vamos a ver un ejemplo, cogiendo como base el ejemplo anterior:

```
SELECT *
FROM c IN Familias.hijos
```

El resultado será:

```
[
  {
    «apellido»: «Apellido2»,
    «nombre»: «Nombre2»,
    «sexo»: «mujer»,
    «grado»: 1
  },
  {
    «apellido»: «Apellido3»,
    «nombre»: «Nombre3»,
    «sexo»: «mujer»,
    «grado»: 8
  }
]
```

Podemos ver que al aplicar la cláusula `IN`, todos los resultados se devuelven en una única lista, mientras que en el primer ejemplo devolvía una lista por resultado.

WHERE

El comando `WHERE` es opcional, y si lo añadimos dentro de una query DocumentDB lo que estamos indicando que condiciones deben satisfacer los resultados que serán incluidos en el resultado de la ejecución.

En el siguiente ejemplo podemos ver como se aplica el operador `WHERE`:

```
[
  {
    «direccion»: {
      «pais»: «España»,
      «ciudad»: «BCN»
    }
  }
]
```

En el ejemplo anterior hemos aplicado el operador `=`, pero no es el único que podemos utilizar con esta cláusula, también se pueden utilizar los siguientes:

- Aritméticos: `+`, `-`, `*`, `/`, `%`
- Lógicos: `AND`, `OR`
- Strings: `||` (concatenación)
- Comparación: `=`, `!=`, `>`, `>=`, `<`, `<=`, `<>`
- Bits: `|`, `&`, `^`
- Binarios y unarios: `true`, `false`, `null`, `undefined`... Un ejemplo de uso de un operador binario sería:

```
SELECT * FROM Familias f WHERE f.registrado
```

Esta query nos devolvería todos aquellos documentos que contengan `true` como valor.

- `BETWEEN`: Palabra clave que especifica un rango de valores. Puede ser usada con strings o números. Por ejemplo:

```
SELECT *
FROM Familias.hijos [ 0 ] c
WHERE c.grado BETWEEN 1 AND 5
```

- `IN`: Palabra clave que se utiliza para verificar que un valor esté en un conjunto predefinido de valores. Por ejemplo:

```
SELECT *
FROM Familias
WHERE Familias.id IN («Familia2», «FamiliaCompartiMOSS»)
```

SELECT

Cualquier consulta sobre DocumentDB debe contener la cláusula `SELECT`. Esta cláusula nos indica los campos que serán devueltos al ejecutarse la query. Por ejemplo, si queremos devolver solo la dirección ejecutaríamos la siguiente query:

```
SELECT f.contacto
FROM Familias f
WHERE f.id = «FamiliaCompartiMOSS»
```

El resultado será:

```
[
  {
    «contacto»: {
      «email»: «email@email.com»,
      «telefono»: «000000000»
    }
  }
]
```

Dentro del operador SELECT podemos realizar las siguientes operaciones:

- Propiedades Anidadas: Devolver una propiedad que está dentro de otra. Por ejemplo:

```
SELECT f.direccion.pais, f.direccion.ciudad
FROM Familias f
WHERE f.id = " FamiliaCompartiMOSS «
```

El resultado será:

```
[
{
  «pais»: «España»,
  «ciudad»: «BCN»
}
]
```

- Proyecciones: Devolvemos el resultado en una estructura propia.

```
SELECT {
  «pais»: «f.direccion.pais»,
  «ciudad»: «f.direccion.ciudad»,
  «nombre»: «f.id»
} FROM Familias f WHERE f.id = « FamiliaCompartiMOSS «
```

El resultado será:

```
[[
  «$!»: {
    «pais»: «España»,
    «ciudad»: «BCN»,
    «name»: « FamiliaCompartiMOSS «
  }
]]
```

- Utilización de Alias: Cuando queremos devolver la estructura propia con un nombre propio.

```
SELECT
{
  «pais»: «f.direccion.pais»,
  «ciudad»: «f.direccion.ciudad»
} AS DireccionInfo,
( «nombre»: «f.id» ) NombreInfo
FROM Familias f
WHERE f.id = " FamiliaCompartiMOSS «
```

El resultado será:

```
[[
  «DireccionInfo»: {
    «pais»: «España»,
    «ciudad»: «BCN»
  },
  «NombreInfo»: {
    «nombre»: « FamiliaCompartiMOSS «
  }
]]
```

- Expresiones escalares: SELECT soporta expresiones del tipo constantes, expresiones aritméticas, lógicas.... Por ejemplo:

```
SELECT ((2 + 11 % 7) - 2) / 3
```

El resultado será:

```
[
{
  «$!»: 1.33333
}
]
```

- VALUE: Es una palabra clave que nos permite devolver un valor. Por ejemplo, si lo quisiéramos devolver una constante:

```
SELECT VALUE «CompartiMOSS»
```

El resultado será:

```
[
  «CompartiMOSS»
]
```

También nos permitiría eliminar etiquetas de los resultados generados:

```
SELECT VALUE f.direccion
FROM Familias f
```

El resultado será:

```
[
{
  «pais»: «España»,
  «ciudad»: «BCN»
}
]
```

- * : Operador especial que nos devolverá todos los campos de la consulta que realicemos:

```
SELECT *
FROM Familias f
WHERE f.id = " FamiliaCompartiMOSS
```

Clausula opcional, aunque si se aplica un filtro en la query o se proyecta algún campo como resultado

El Resultado será:

```
[
{
  «id»: «FamiliaCompartiMOSS»,
  «padres»: [
    {
      «apellido»: «CompartiMOSS»,
      «nombre»: «Revista»
    },
    {
      «apellido»: «Apellido1»,
      «nombre»: «Nombre1»
    }
  ],
  «hijos»: [
    {
      «apellido»: «Apellido2»,
```

```

«nombre»: «Nombre2»,
«sexo»: «mujer»,
«grado»: 1,
«mascotas»: [
  {«nombre»: «Goofy»},
  {«nombre»: «Mickey»}
],
{
  «apellido»: «Apellido3»,
  «nombre»: «Nombre3»,
  «sexo»: «mujer»,
  «grado»: 8
}
],
«direccion»: {
  «pais»: «España»,
  «ciudad»: «BCN»
},
«contacto»: {
  «email»: «email@email.com»,
  «telefono»: «000000000»
},
]

```

- BETWEEN: Tiene el mismo significado y uso que en el Where.

```

SELECT (c.grado BETWEEN 0 AND 10)
FROM Familias.hijos [ 0 ] c

```

- Operadores ? (Ternary) y ?? (Coalesce): Estos operadores nos permiten aplicar condiciones. Por ejemplo:

```

SELECT (c.grado < 5)? «directo»: «indirecto» AS nivelgrado
FROM Familias.hijos [ 0 ] c

SELECT f.nombre ?? f.apellido AS apellido
FROM Familias f

```

- TOP: Operador que nos permitirá devolver un número concreto de resultados deseados.

```

SELECT TOP 1 *
FROM Familias f

```

El resultado será:

```

[
  {
    «id»: «FamiliaCompartiMOSS»,
    «padres»: [
      {
        «apellido»: «CompartiMOSS»,
        «nombre»: «Revista»
      },
      {
        «apellido»: «Apellido1»,
        «nombre»: «Nombre1»
      }
    ],
    «hijos»: [
      {
        «apellido»: «Apellido2»,
        «nombre»: «Nombre2»,
        «sexo»: «mujer»,
        «grado»: 1,
        «mascotas»: [
          {«nombre»: «Goofy»},
          {«nombre»: «Mickey»}
        ]
      },
      {

```

```

«apellido»: «Apellido3»,
«nombre»: «Nombre3»,
«sexo»: «mujer»,
«grado»: 8
}
],
«direccion»: {
  «pais»: «España»,
  «ciudad»: «BCN»
},
«contacto»: {
  «email»: «email@email.com»,
  «telefono»: «000000000»
},
«registrado»: false
}
]

```

Arrays y Objetos

DocumentDB permite la creación de arrays y/o objetos. Podemos construir arrays de la siguiente forma:

```

SELECT [ “f.direccion.ciudad”, “f.direccion.pais” ] AS PaisCiudad
FROM Familias f

```

El Resultado será:

```

[
  {
    « PaisCiudad «: [
      «BCN»,
      «España»
    ]
  },
  {
    «CiudadState»: [
      «BCN»,
      «BCN»
    ]
  }
]

```

JOINS

DocumentDB nos permite realizar JOINS, pero a diferencia del concepto JOIN de una Base de Datos relacional, no nos sirve para relacionar tablas de forma que podamos obtener los datos deseados, sino que nos permite relacionar el schema consigo mismo, lo que en una base de datos relacional equivaldría a realizar una “self-join”.

El resultado de realizar la operación JOIN es un conjunto de N-tuplas con N-Values por tupla, con valores obtenidos de iterar los todos los alias de las collections sobre sus conjuntos.

El comando WHERE es opcional, y si lo añadimos dentro de una query DocumentDB lo que estamos indicando que condiciones

Por ejemplo, podemos realizar una JOIN entre la raíz y un

hijo, el resultado es el producto cruzado entre los dos objetos JSON.

Supongamos que, para este ejemplo, además de la entrada anterior, disponemos de la siguiente entrada

```
{
  «id»: «FamiliaCompartiMOSS2»,
  «padres»: [
    {
      «apellido»: «CompartiMOSS22»,
      «nombre»: «Revista2»
    },
    {
      «apellido»: «Apellido12»,
      «nombre»: «Nombre12»
    }
  ],
  «hijos»: [
    {
      «apellido»: «Apellido22»,
      «nombre»: «Nombre22»,
      «sexo»: «mujer»,
      «grado»: 1,
      «mascotas»: [
        {«nombre»: «Goofy2»},
        {«nombre»: «Mickey2»}
      ]
    }
  ],
  «direccion»: {
    «pais»: «España»,
    «ciudad»: «MAD»
  },
  «contacto»: {
    «email»: «email2@email.com»,
    «telefono»: «000000001»
  },
  «registrado»: true
}
```

Y realizamos la siguiente query:

```
SELECT f.id
FROM Familias f
JOIN c IN f.hijos
```

El Resultado será:

```
[
  {
    «id»: «FamiliaCompartiMOSS2»
  },
  {
    «id»: «FamiliaCompartiMOSS»
  },
  {
    «id»: «FamiliaCompartiMOSS»
  }
]
```

ORDER BY

En caso de añadir este operador, nos permitirá ordenar los

resultados de la query de la forma deseada, seguimos teniendo las dos entradas en nuestra collection.

```
SELECT f.id, f.direccion.ciudad
FROM Familias f
ORDER BY f.direccion.ciudad
```

El Resultado será:

```
[
  {
    «id»: «FamiliaCompartiMOSS»,
    «ciudad»: «BCN»
  },
  {
    «id»: «FamiliaCompartiMOSS2»,
    «ciudad»: «MAD»
  }
]
```

CONCLUSIONES

Como podéis observar, realizar queries sobre DocumentDB es muy similar a realizar queries sobre SQL, todos los operadores que tiene son familiares y funcionan de la forma que esperamos. Query playground es un sitio web proporcionado por Microsoft para poder practicar y ver más en profundidad como realizar queries en DocumentDB.

Destacar que DocumentDB también permite trabajar con funciones del tipo:

- Matemático: ABS, CEILING, EXP.....
- Comprobación de tipos: IS_ARRAY, IS_BOOL...
- Funciones de Strings: CONCAT, CONTAINS, ENDSWITH...
- Funciones de Arrays: ARRAY_CONCAT, ARRAY_CONTAINS...
- Funciones de espacio: ST_DISTANCE, ST_WITHIN...

Para saber más sobre funciones en DocumentDB, visitar los links del apartado referencias.

REFERENCIAS:

- <https://docs.microsoft.com/en-us/azure/documentdb/documentdb-sql-query#built-in-functions>
- <https://www.documentdb.com/sql/tutorial>
- <https://www.documentdb.com/sql/demo>

ROBERT BERMEJO

Arquitecto .Net

roberto.bermejo@sogeti.com

@robertbermejo

www.robertbermejo.com

Integración de SharePoint y Wordpress mediante webhooks, Azure y el API de Wordpress.com (I)

Introducción

SharePoint Online es una plataforma madura y en continua evolución que ofrece un amplio abanico de soluciones para la colaboración de personas. Por diversos motivos Microsoft abandonó la posibilidad de disponer de un Sitio Web público (<https://support.microsoft.com/en-us/kb/3027254>), lo que obliga a emplear otras herramientas para poder dar solución a la publicación de contenidos en internet (usuarios anónimos).

Tradicionalmente para conectar SharePoint Online con otros sistemas se empleaban, dependiendo del año y versión, diferentes aproximaciones que se pueden dividir en dos tipos:

- 1.- Las que lanzaban algún tipo de tarea que recorría SharePoint para detectar cambios de forma periódica.
- 2.- Y las que estaban basadas en patrones y técnicas específicas de SharePoint, como los Receptores de Eventos Remotos (RER) o los flujos de trabajo con sus llamadas HTTP entre otras técnicas.

En este artículo veremos cómo puede emplearse la característica de webhooks de SharePoint (actualmente en desarrollo) para facilitar un escenario de integración con Wordpress, que suele ser un requisito de la gente de Comunicación y Marketing, al ser este el estándar de facto para blogs y webs corporativos y por tanto una opción deseable en empresas con SharePoint Online y presencia online. Para realizar esta implementación vamos a usar el código del repositorio de SharePoint de Microsoft en Github: <https://github.com/SharePoint>

¿Qué son los webhook?

Los webhooks son un patrón de arquitectura usado precisamente para facilitar la integración entre sistemas web.

Sendgrid (https://sendgrid.com/docs/API_Reference/Webhooks/event.html) o Github (<https://developer.github.com/webhooks/>) disponen de ellos, permitiendo que suscribamos nuestras propias aplicaciones a determinados eventos de sus plataformas.

Para ampliar información sobre Webhooks hay bastante donde elegir debido al tirón que han tenido en los últimos

años, pero para empezar sugeriría leer el post que hizo Sendgrid en 2014: <https://sendgrid.com/blog/whats-webhook/>, el del padre del patrón, Jeff Lindsay: <http://progrium.com/blog/2012/11/19/from-webhooks-to-the-evented-web/> y si eres desarrollador ASP.NET el de Microsoft el año pasado: <https://blogs.msdn.microsoft.com/webdev/2015/09/04/introducing-microsoft-asp-net-webhooks-preview/>

La aparición de los webhooks para SharePoint son un paso más en la apertura y estandarización de las tecnologías Microsoft, siendo el reemplazo de soluciones menos optimas o más a medida, lo que debería permitir soluciones más robustas y fáciles de desarrollar tras su adopción.

SharePoint Online es una plataforma madura y en continua evolución que ofrece un amplio abanico de soluciones

¿Qué tenemos en Wordpress?

Al margen de si usamos un Wordpress de Azure (de Wordpress, Bitnami o incluso en un contenedor Docker) o una cuenta de pago o gratuita en Wordpress.com, disponemos de 2 APIs diferentes para gestionar contenidos en el sistema: XML-RPC y REST. Y por supuesto dos formas de acceder a nivel de seguridad: OAuth o el típico usuario y password.

La elección de REST y OAuth nos evitaría tener que mantener las credenciales del sistema, aunque quizá sea más fácil usar XML-RPC con credenciales de usuario si confiamos en la seguridad de nuestros sistemas y usuarios o si no queremos trabajar con OAuth por alguna razón.

Quien desee probar el API REST dispone de una consola de desarrollo en el propio Wordpress.com <https://developer.wordpress.com/docs/api/console/>

Requisitos previos

Antes de ponernos a desplegar código, recomiendo empezar con el aprovisionamiento de lo que vamos a necesitar, debido a que ciertos elementos pueden requerir un tiempo para completar su configuración y provocar parones en la puesta en marcha del código de ejemplo. Advierto esto tras haberse llegado a juntar durante la implementación de todo el sistema, errores personales, errores en el portal y

bugs de producto, provocando un retraso importante sobre el tiempo estimado inicialmente.

Lo primero que necesitaremos será disponer de lo siguiente:

- Un SharePoint Online de desarrollo, con la característica de First Release activada a nivel de suscripción.
 - Ojo con esto que en el portal antiguo de administración de O365 es sencillo de encontrar. En el nuevo está en Settings -> Organization profile -> Release preferences -> Release track.
 - Y de nuevo ojo que la aplicación de “First Release” puede durar algún tiempo (1 día).
 - Además, deberíamos disponer de un catálogo de aplicaciones configurado y un sitio de desarrollador con una nueva lista simple de nombre reconocible (ej: Wordpress) que contenga una lista con campos para Título y Contenido de tipo texto plano de una línea y multi-línea respectivamente.
- Una suscripción de Azure. La de desarrollador nos vale ya que los elementos a emplear son muy, muy baratos. Los elementos a crear dentro de esta se detallan más adelante.
- Un Wordpress. Sugiero usar uno gratuito en Wordpress.com por facilidad a la hora de implementar la configuración necesaria.
 - Recomendaría probar que el API está configurado y funcionando mediante la herramienta que nos ofrece la compañía. Para hacer un post podemos usar esta URL (siempre que estemos autenticados, claro): [https://developer.wordpress.com/docs/api/1.1/post/sites/\[NOMBREDELBLOG\].wordpress.com/posts/new/](https://developer.wordpress.com/docs/api/1.1/post/sites/[NOMBREDELBLOG].wordpress.com/posts/new/)
- El código del repositorio Github clonado, y compilado en un Visual Studio 2015: <https://github.com/SharePoint/sp-dev-samples/tree/master/Samples/WebHooks.List>
 - Podría ser necesario instalar las Office Tools Extension de <https://www.visualstudio.com/es/vs/office-tools/>
 - Y necesitaremos instalar las características necesarias mediante el botón derecho “Install Missing Features” en el proyecto correspondiente para que compile todo.
- Un SQL Management Studio o cualquier otro cliente que nos permita lanzar algo de SQL contra un Azure SQL Database.

Manos a la obra

Una vez obtenido todo lo anterior podemos empezar a seguir paso a paso la guía de despliegue que también encontraremos en el repositorio de Github, en el típico archivo Markdown: <https://github.com/SharePoint/sp-dev-samples/blob/master/Samples/WebHooks.List/Deployment%20guide.md>

Dado que la propia guía ya contiene imágenes paso a paso, y dado que podría haber cambios cuando la característica salga en release, me limitaré a hacer un resumen de las acciones necesarias, aunque recomendaría automatizar el despliegue en un script de PowerShell, si no entero, al menos de las partes más básicas.

- 1.- Creamos una Web App con el nombre que queramos.
 - Descargamos su perfil de publicación.
- 2.- Creamos una Azure SQL Database con el nombre que queramos en un Azure SQL Server existente o nuevo.
 - Añadimos nuestra IP actual para poder acceder a la instancia desde nuestro equipo.
- 3.- Creamos una cuenta de Azure Storage.
 - Obtenemos la cadena de conexión. *Nota: yo me topé con un problema en el Portal, que provocaba que no se cargase la pestaña con la cadena de conexión, pero esta puede generarse sin problemas obteniendo la Key en el portal antiguo o desde el propio Visual Studio, como se aprecia en esta imagen:*

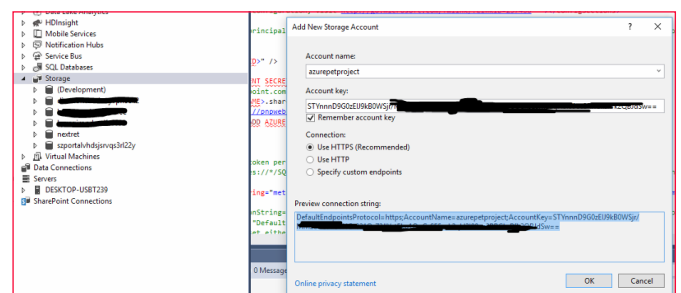


Imagen 1.- Cadena de conexión y account key.

- 4.- Registramos una app en el sitio de desarrollo, mediante la URL https://.../sites/NOMBREDELSITIO/_layouts/15/appregnew.aspx
- 5.- Conectamos con el SQL Management Studio a la base de datos creada y:
 - Ejecutamos el Script SharePointWebHooks.sql de la carpeta SQL\Scripts alojada en el proyecto SharePoint.WebHooks.Common.
 - Sobre la base de datos MASTER, ejecutamos (preferiblemente con otro password):

```
CREATE LOGIN WebHooksAdmin WITH PASSWORD = 'Pass@word!'
```

- Sobre la base de datos WebHooks generada en 5.a, ejecutamos:

```
CREATE USER [WebHooksAdmin] FOR LOGIN [WebHooksAdmin] WITH DEFAULT_SCHEMA = dbo
EXEC sp_addrolemember 'db_datareader', 'WebHooksAdmin'
EXEC sp_addrolemember 'db_datawriter', 'WebHooksAdmin'
```

- 6.- En la solución de Visual Studio reemplazamos la configuración existente para que cuadre con la nuestra. Es decir, nuestro Client ID, urls, usuario y

passwords, cadenas de conexión de Storage y SQL...
 • Recomiendo llevar a cabo este paso con calma. :-).

- 7.- Desplegamos el Add-In que contiene el proyecto SharePoint.WebHooks.MVCWeb en Azure.
 - Esto requiere el perfil de publicación de nuestra Web App y algunos retoques detallados en la guía de despliegue.
- 8.- Desplegamos el Add-in que contiene el proyecto SharePoint.WebHooks.MVC en SharePoint.
 - Ojo que se trata de un add-in a desplegar como paquete, es decir cargando el archivo .app en el catálogo de aplicaciones.
- 9.- Desplegamos el Web Job que contiene el proyecto SharePoint.WebHooks.Job como web job (botón derecho Publish as Azure Web Job).
 - Lo debemos publicar en modo de ejecución continua, mediante el perfil de publicación de nuestra Web app del punto 1.a. y cambiando de http a https.
 - Mucho ojo con le Web Job, debemos asegurarnos de que esté "Running". Bajo ciertas circunstancias publicamos y ese no es el estado. Podemos reiniciarlo sin piedad si es necesario.

Los webhooks son un patrón de arquitectura usado precisamente para facilitar la integración entre sistemas web

- 10.- Para terminar el despliegue, creamos una nueva App desde Contenidos del Sitio, del tipo que hemos cargado en el punto 8: SharePoint.WebHooks.MVC. Se nos pedirá que aceptemos la aplicación con los permisos que establece.

Una vez realizados estos "pocos" pasos deberíamos tener corriendo el sistema de tal modo que, al pulsar en la App que hemos añadido al sitio de SharePoint, se nos redirigirá a la aplicación de nuestra Web App

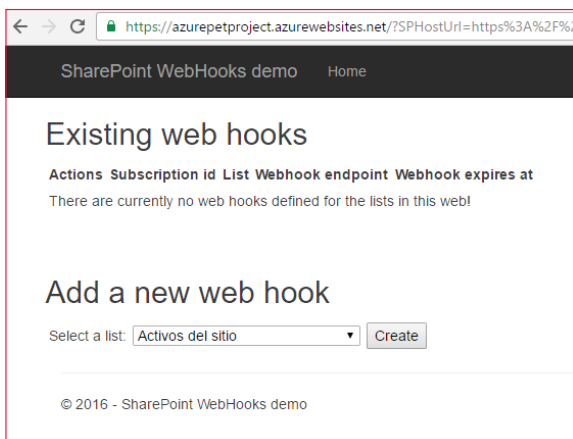


Imagen 2.- Aplicación Web desplegada.

Y seleccionando la lista que creamos para publicar en Wordpress, podemos generar una suscripción a esa lista.

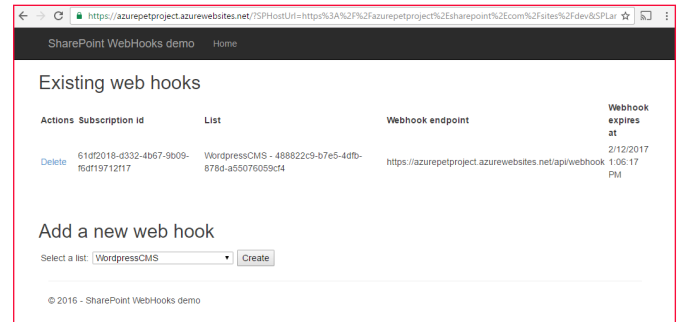


Imagen 3.- Generación de suscripción a una lista.

Y ahora sí, tenemos lista una aplicación suscrita a eventos de SharePoint suministrados mediante la característica de webhook de SharePoint, lo que nos ofrece de una forma más robusta y estándar de integrar SharePoint que en con las herramientas actuales.

Con todo el sistema funcionando, recibimos la notificación de forma rápida, no perdemos información gracias a la mecánica de reintento implementada en este patrón webhook, podemos guardarlas para tratarlas en lotes (lo que tiene sentido para ciertas integraciones), disponemos de un histórico de todo lo ocurrido gracias al almacenamiento en Azure... y todo ello de forma muy barata si establecemos adecuadamente los niveles de nuestros elementos (Pricing Tier).

Ya solo nos queda configurar la acción o acciones que queramos realizar al recibir la notificación. En nuestro caso: publicar un post en Wordpress. Para ello deberemos proceder a configurar nuestro Wordpress y añadir el código necesario en el proyecto SharePoint.WebHooks.Common.

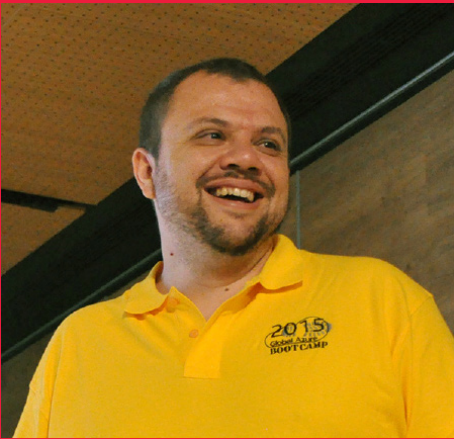
Conclusiones de la primera parte

Desgraciadamente, la cantidad de pasos necesarios para poner en marcha todo sistema es más extensa de lo que me gustaría (a mi o a cualquiera), y podría haber algunos cambios en la versión release de los webhooks de SharePoint, por lo que prefiero dejar para la segunda parte del artículo la publicación de contenidos desde Azure hacia Wordpress en el próximo número de CompartimOSS.

En cualquier caso, espero que esta primera parte haya dado una idea de la potencia de los webhooks en escenarios de integración entre plataformas, y en particular para nuestros desarrollos con SharePoint.

Continuará...

JOSÉ ÁNGEL BOLAÑO
 Solutions Architect en NextRET
 @noradrex



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.siderys.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

Ingeniero de Telecomunicaciones por la Universidad de Valladolid y Diplomado en Ciencias Empresariales por la Universidad Oberta de Catalunya (UOC). Cuenta con más de 12 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 Servers & Services 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. Hasta la fecha, ha publicado 9 libros sobre SharePoint & Office 365, así como varios artículos en castellano y en inglés sobre ambas plataformas.

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 tecnologías de Microsoft en español, todo el contenido deberá ser directamente relacionado con Microsoft y escrito en castellano. No hay limitaciones sobre el tipo de artículo o contenido, lo mismo que sobre el tipo de tecnología. 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: Artículos cortos: Máximo 1000 palabras y 1 o 2 figuras. Describen rápidamente una aplicación especial de alguna tecnología de Microsoft, o explica algún punto poco conocido o tratado. Experiencias de aplicación 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 tecnologías de Microsoft, 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) 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

