

Nº31 marzo 2017

67 %



Comparti MOSS

REVISTA ESPECIALIZADA EN TECNOLOGÍAS MICROSOFT

Entrevista Luis
Valencia

Introducing
Microsoft
PowerApps

Destripando
Modern Sites y
Pages en Share-
Point Online

Microsoft Ope-
rations Mana-
gement Suite
(OMS)

Staff

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

DIRECCIÓN GENERAL

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

DISEÑO Y DIAGRAMACIÓN

- Santiago Porras Rodríguez

Contacte con nosotros

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

BLOGS

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

REDES SOCIALES

Facebook:

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

LinkedIn:

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

Twitter:

@CompartiMOSScom

Contenido

03

Editorial

07

Control de Errores en SPFX, Out of the Box y con Application Insight

12

Introducción a FastTrack Network Analysis para Office 365

21

Destripando Modern Sites y Pages en SharePoint Online

26

Gobernabilidad en Power BI

31

Api Management y Logic Apps como Bus de Integración

43

Cloud PBX ¿Cuál es la solución adecuada para mí?

48

Introducing Microsoft PowerApps

54

Integrando Bot Framework y Microsoft Teams

60

Enterprise Mobility Suite Parte 2 – Azure AD Premium

04

Azure Serverless

10

Emulando DocumentDB

15

System Center: Una solución integral para la gestión de la nube privada

23

SharePoint y Azure: Azure Content Delivery Network (CDN)

29

Entrevista Luis Valencia

38

Temperatura en tiempo real, aplicación web con Raspberry, Azure IoT Hub y SignalR

45

Integración de SharePoint con Power BI

50

Microsoft Operations Management Suite (OMS)

58

Activación de las opciones de guardado de plantilla en SharePoint Online



03

Editorial

Siguiendo el plan maestro que nos hemos impuesto para CompartiMOSS, les ofrecemos hoy el número con la mayor cantidad de artículos, y con la mayor variedad de temas que hasta hoy, después de 31 ediciones, hemos producido. Nuestra ambición es convertir a CompartiMOSS en LA revista por excelencia sobre temas relacionados con tecnologías y plataformas de Microsoft. Y ya hemos recorrido un buen camino en esa dirección.

Dentro de los cambios que estamos introduciendo desde este número, el principal es que un grupo de personas tan motivadas como nosotros, el grupo de editores, se van a encargar de las diferentes secciones de la revista. Al final de la revista podrán encontrar quienes son los “Responsables de sección”: cada uno de ellos es un experto en su tecnología, y garantizará la calidad de los artículos correspondientes. Desde la dirección estamos en contacto permanente con ellos, además de estar trabajando para obtener algunos medios de trabajo que faciliten su tarea, como, por ejemplo, crear direcciones de Email para que ustedes, nuestros lectores, se puedan contactar más fácilmente con la revista.

Como podrán encontrar también, hemos incluido un artículo en inglés. Aunque la revista es, y seguirá siendo, eminentemente en español, la filosofía de apertura que siempre nos ha caracterizado nos hace ser receptivos a publicar información en otros idiomas. Ya lo hemos hecho en el pasado, y esperamos seguirlo haciendo en el futuro, y no solamente en inglés, sino con artículos en cualquier otro idioma que nos ayuden a comprender mejor el complicado, pero excitante, mundo de las tecnologías de Microsoft.

Finalmente, solo nos queda agradecerles a todos ustedes, nuestros lectores, por ayudarnos a seguir con CompartiMOSS y mejorarla cada vez más. Y por supuesto, reconocer la labor de todas las personas (editores, diagramadores, responsables de secciones) y entidades (nuestros patrocinadores) que colaboran para crear este trabajo que consideramos tan importante para toda la comunidad. Esperamos, como de costumbre, que se diviertan tanto leyendo este número, como nosotros nos divertimos haciéndolo.

El equipo editorial de CompartiMOSS

Azure Serverless

No, no es una nueva feature de Azure de la que no te hayas enterado. En este artículo vamos a entender que es “serverless” ese concepto tan en boca de todos, y los componentes de Azure que se pueden considerar parte de esta moda.

¿Qué es “serverless”?

“Serverless”, por supuesto, quiere decir “sin servidor”. Para cualquiera de nosotros es muy difícil pensar en algo en Internet que no tenga servidor, al menos servidor como ese ente al que le pedimos cosas y nos devuelve cosas: le hacemos preguntas, nos devuelve respuestas; le pedimos páginas, nos devuelve páginas; le pedimos fotos, nos devuelve fotos; le enviamos un archivo, nos devuelve su permalink; le pedimos dinero, nos devuelve un 404.

El serverless que está tan de moda se refiere más a servicios con los que no te tienes que parar a pensar en servidores, aunque sepas que en el fondo ahí estarán.

En general, todo el mundo lo define como la unión de dos “aaS”: el BaaS (Backend as a Service) y el FaaS (Functions as a Service).

El primer concepto, BaaS, se refiere a los servicios que pueden hacer de backend. Obvio. Esto quiere decir que son aquellos servicios de back a los que, sin preocuparte demasiado de su gestión o dimensionamiento, podrías acceder desde tu aplicación móvil, por ejemplo.

El segundo, FaaS, se usa para servicios que ejecutan tu código sin necesidad de que dimensiones ni te preocupes de escalar. Le dan lógica al BaaS, manteniendo su misma filosofía.

Tal vez no os hayáis quedado muy a gusto con estas definiciones, así que os daré la mía propia:

Serverless es todo aquel servicio para el que no me tengo que preocupar prácticamente nada de su infraestructura subyacente, que por defecto escala elásticamente y por el que se me cobrará por volumen de tráfico.

Estas son las tres condiciones que creo que hay que tener en mente para hablar de serverless, pero si queremos añadirle un “aaS” por ser más cool, yo diría que es MaaS (Management as a Service), ya que son servicios en los que la gestión propia de los servicios queda reducida a la nada, y

pensando como DevOps sólo nos preocuparíamos de ella en el deployment.

Azure Functions

Cuando se habla de Azure y de serverless, lo que sale siempre a colación es Azure Functions. Esta es una de las features de Azure que te permite ejecutar el código que tú quieras de un modo independiente a los servidores. Es algo así como unos webjobs evolucionados que autoescalan. De hecho, el motor subyacente es el mismo que el de los webjobs.

Lo que es Azure Functions ya lo expuso claramente en su momento Gustavo Velez en esta revista [1], por lo que sólo nos fijaremos en nuestras tres condiciones:

¿Me he de preocupar de su infraestructura?

La respuesta corta es “no”, la larga es un “más o menos no”.

Imagen 1.- Primer paso de la creación de una cuenta de Functions.

Para despreocuparnos de la infraestructura subyacente, hay dos cosas que tenemos que tener en cuenta. La prime-

ra es el Hosting Plan, en el que deberemos dejar la opción por defecto (Consumption plan) que hará que se nos cobre por consumo. La alternativa es vincularlo a un App Service Plan, y que nuestras funciones se ejecuten en las máquinas de ese App Service Plan... esto sería prácticamente igual que hacer unos webjobs, y estando esa parte mucho más madura no veo motivos para hacerlo. Con el consumption plan, nos contarán uno cada vez que se lance el disparador que asociemos a la función y que hace que se ejecute y nos cobrarán por el número de veces que se haya ejecutado.

“...es muy difícil pensar en algo en Internet que no tenga servidor, al menos servidor como ese ente al que le pedimos cosas y nos devuelve cosas...”

Como curiosidad, podremos ver que cuando creamos una cuenta de Functions con este plan, se nos creará un App Service Plan sobre el que se ejecutará, pero este será del Pricing Tier “Consumption”, que es uno que nosotros no podemos seleccionar cuando creamos un service plan... tal vez algún día nos den esa opción para otras cosas.

La segunda cosa a tener en cuenta es los recursos que consume nuestra función. No es lo mismo una ejecución de un segundo que de un minuto, al igual que no es lo mismo que necesite cien megas o mil. El tiempo de ejecución es una métrica bastante evidente, pero la memoria no tanto tal y como la necesitamos calcular. Si vemos la calculadora, observaremos que viene en tramos de 128MB y nuestro servicio se englobará en el menor de los que pueda contenerle. Por ejemplo, si nuestro servicio necesita 200MB para ejecutarse, se nos aplicará el precio de 256.

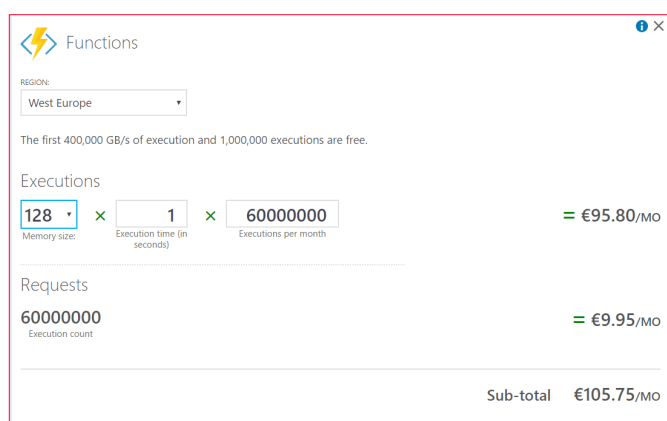


Imagen 2.- Calculadora de precios de funciones.

¿Escala dinámicamente y se me cobrará por tráfico?

Teniendo en cuenta las puntualizaciones que hemos hecho en el apartado anterior, podemos decir que sí a todo. No nos tenemos que preocupar de que escale. En función del número de ejecuciones, Azure se preocupará de alojarlo en más o menos máquinas, sin que nosotros tengamos que

estresarnos por picos que no sabemos si algún día llegarán.

Con la salvedad del tiempo y de la memoria, la única otra variable que nos aplica a la hora de calcular el precio es el volumen de ejecuciones, por lo que si tengo mucho pagaré más y si tengo pocas no pagaré. Con Azure Functions puedes tener servicios desplegados a coste cero.

¿Qué otras cosas hay?

Ahora que ya hemos visto lo que es serverless y un ejemplo detallado del porqué un servicio es serverless, toca pararse a pensar en la arquitectura de una solución completa:

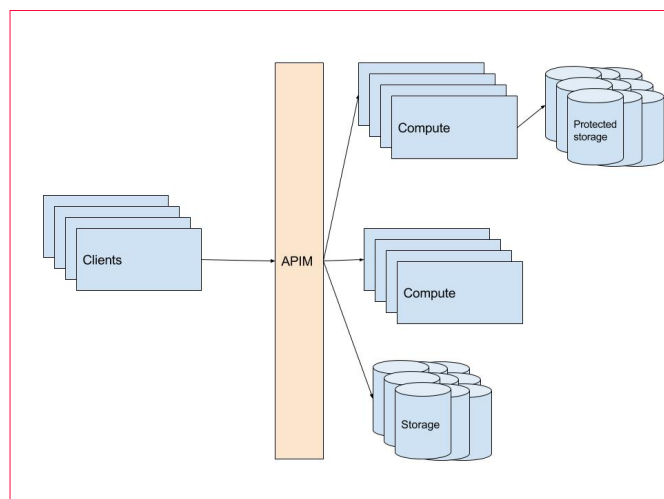


Imagen 3.- Posible arquitectura de solución serverless.

Como podemos ver en la imagen 3, no sólo de Functions puede vivir el informático, y en una arquitectura serverless es raro que vayamos a tener sólo una solución de computación como es Functions.

Functions nos puede dar un conjunto de microservicios, pero muy posiblemente necesitaremos algún tipo de almacenamiento de datos, al que con suerte podremos acceder directamente desde el cliente. “Con suerte”, porque es muy fácil pensar en casos en los que es prácticamente impensable que el cliente acceda directamente al almacenamiento sin más validaciones, como por ejemplo una base de datos de ventas ¿nos fiaríamos de no validar que las compras no hayan sido enviadas con un importe de 0?

Pero no sólo son estos servicios, ya que si pensamos en una solución muy grande y distribuida que pueda emplear servicios muy dispares es muy posible que nos interese usar algún servicio de gestión de APIs para unificar todo nuestro backend.

Tirando del hilo que hemos empezado con las Functions, podríamos llegar a la mayoría de los componentes de Azure que se pueden considerar serverless, sobre todo si nos fijamos en sus bindings y triggers [2]. Para que no se nos escape nada, sin embargo, creo que es mejor darse un paseo por la calculadora e ir fijándose en todos los elementos, ya que la forma en la que se facturen nos dirá mucho.

- Remote App: es lo primero que nos encontramos en

la calculadora cuyo modelo de precios nos indica que podría ser uno de estos servicios serverless. Es un servicio que permite ejecutar aplicaciones de escritorio remotamente mediante RDP... podríamos entrar en una discusión de si a este servicio de tan alto nivel lo incluimos o no en nuestro abanico de opciones, pero la verdad es que no tendría sentido ya que es un servicio que dejará de existir en el futuro cercano de Azure.

- **Storage:** tablas, colas y blobs. No necesitas saber cuál es su infraestructura, no te tienes que preocupar por el escalado y el precio es en función del tamaño total de lo guardado, los accesos, la transferencia y el tipo de réplica de los datos. Además, puedes acceder directamente desde tu aplicación (es un ejemplo muy claro de BaaS).
- **Media Services:** es un servicio de alto nivel para trabajar con audio y video de varias maneras. Como todos los servicios de alto nivel de Azure lo podemos considerar serverless, ya que desde el principio están pensados para que resulte sencillo y que el coste sea en base al consumo.
- **Azure Search:** es otro de esos servicios de alto nivel, y por tanto también podemos considerarlo como uno de los componentes de Azure serverless, que podremos usar siempre que queramos indexar información y meter capacidades de búsqueda a nuestras aplicaciones.
- **Mobile Engagement:** otro de estos servicios de alto nivel, en este caso para proveer de un servicio integral de gestión de aplicaciones móviles.
- **API Management:** ya hablé de él en esta misma revista [3]. Es un servicio con un coste elevado pero imprescindible cuando necesitas aunar múltiples servicios de distintas fuentes, o quieres evitar el vendor locking por ejemplo. Muy útil en arquitecturas serverless.
- **Notification Hubs:** otro servicio de alto nivel que permite mandar notificaciones a dispositivos móviles (que no teléfonos móviles, ya que incluye dispositivos que no son teléfonos). Se parte de 10 millones de push y luego se paga por cada millón. Es cierto que el hecho de tener un mínimo no cumple del todo con el pago por uso, pero en general creo que también cumple nuestras tres premisas para formar parte de nuestro set de servicios serverless.
- **DocumentDB:** es una base de datos orientada a documentos en la que puedes elegir que se te cobre por el uso (almacenamiento y velocidad que quieras), por lo que es otro de estos servicios que podemos sumar a nuestro conjunto de herramientas.
- **Machine Learning:** es un servicio que no manejo apenas por lo que no me atrevo ni a definirlo, pero por lo poco que lo conozco creo que es uno de esos servicios que se pueden considerar como serverless.
- **Cognitive Services:** son un montón de servicios inteli-

gentes que empezó a mover Microsoft el año pasado con el boom de la IA. Depende del servicio concreto se paga un fijo o se paga por uso, por lo que en cada caso habría que ver si nos vale o no para un sistema serverless en el que queramos pagar única y exclusivamente si se usa.

- **Hubs:** tanto el IoT Hub como los Event Hubs, son elementos a tener muy en cuenta cuando se está montando una arquitectura serverless que quieras que pueda escalar hasta el infinito. Tienen sus costes mínimos, por lo que se podría discutir si los metemos o no en el saco, pero de cualquier modo necesitaremos tenerlos en mente de cara a plantear una nueva solución.
- **Logic Apps:** son aplicaciones que se pueden definir con un sistema de drag and drop y unir los puntos para hacer flujos de datos con distintos triggers y distintas acciones. Está pensado principalmente para que un usuario final avanzado, pero no técnico (alguien de marketing por ejemplo), pueda definirse sus propias funcionalidades. Se cobra completamente por uso, y se integra con muchos de los servicios que hemos enumerado, por lo que lo podremos considerar para la arquitectura de cualquier aplicación serverless que queramos.

Conclusión

Hay otros servicios que no he metido en la lista, como son los servicios para desarrolladores como Team Services o HockeyApp, o los servicios que están pensados más para los grandes sistemas corporativos como ServiceBus o Active Directory, que perfectamente se podrían considerar como serverless. Además, aunque muchas veces se nos olvide, hay multitud de servicios en el marketplace que cumplen con nuestras tres premisas del serverless.

Con todos estos, estamos en disposición de decir que Azure Serverless no es sólo Functions y desde luego no acaba de llegar porque hay múltiples servicios y algunos de ellos están aquí desde el primer día.

[1]: <http://www.compartimoss.com/revistas/numero-30/share-point-y-azure-azure-functions>

[2]: <https://docs.microsoft.com/en-us/azure/azure-functions/functions-triggers-bindings>

[3]: <http://www.compartimoss.com/revistas/numero-28/introduccion-a-azure-api-management>

JAVIER LÓPEZ GONZÁLEZ

mail@javilopezg.com

@javilopezg

<https://javilopezg.com>

Control de Errores en SPFX, Out of the Box y con Application Insight

A pesar de que existe una gran cantidad de documentación sobre el SPFx hay algunos elementos que existen, pero los desarrolladores no los utilizamos debido a que no los conocemos. Todo lo que contiene el SPFx lo podemos encontrar en la siguiente dirección: <https://sharepoint.github.io/>. En la misma se encuentra todos los métodos, clases, funciones que hay disponibles así como la librería en la que se encuentra y la cual deberemos de importar en nuestra solución.

En este artículo vamos a ver uno de esos métodos ocultos: la clase "Log". En primer lugar, vamos a crear un proyecto utilizando la plantilla de Yeoman de SharePoint.

```
>yo @microsoft/sharepoint
```

Para este caso no hace falta seleccionar ningún Framework en particular dentro de la selección del Framework JavaScript que deseamos utilizar. Hay que tener en cuenta que en caso de que se seleccione la opción de "No Framework JavaScript" hay un error en la última actualización de dicho código. Este error es debido a que hay una librería fuera de versión y para solucionarlo (mientras el equipo de producto lo soluciona) hay que lanzar la siguiente instrucción:

```
>npm install @types/lodash@4.14.50
```

Una vez ya tenemos la aplicación funcionando, lo que debemos hacer para poder utilizar el Framework es importar la referencia donde está el LOG, para ello hay que añadir el siguiente fragmento de código en la clase:

```
import { Log } from '@microsoft/sp-core-library';
```

Nota: Dado que el Framework lo estamos adoptando desde una fase muy temprana, en anteriores versiones la referencia del log (y de otras utilidades) estaba ubicada en la librería @microsoft/sp-cliente-base

Esta clase Log tiene los siguientes métodos:

- error(source: string, error: Error, scope?: ServiceScope): void
- info(source: string, message: string, scope?: ServiceScope): void
- verbose(source: string, message: string, scope?: ServiceScope): void

ceScope): void

- warn(source: string, message: string, scope?: ServiceScope): void

Dependiendo del nivel de criticidad elegiremos un método u otro, estos métodos no tienen mucho misterio salvo el parámetro opcional "scope", esta variable es interesante añadirse ya que nos puede proporcionar más información del error que se ha producido, ya que proporciona información del contexto de ejecución del propio "WebPart".

Si dentro del método Render (que hay por defecto en la solución generada) añadimos las siguientes líneas de código:

```
Log.info("Render", "Empezamos a renderizar el contenido");
Log.warn("Aviso", "Esta arrancando");
Log.error("Error", new Error("Es un error"));
Log.verbose("Verbose", "Logueo Verbose");
```

Compilamos la solución de nuevo, arrancamos el Workbench y abrimos la consola del navegador veremos efectivamente que se ha registrado los logs tal y como esperábamos.

Render: message	DefaultLogHandler.js:9
⚠ Aviso: WARNING message	DefaultLogHandler.js:12
❌ Error: ERROR: Es un error	DefaultLogHandler.js:15
Verbose: VERBOSE message	DefaultLogHandler.js:6

Imagen 1.- Mensajes de Log dentro de la consola del navegador.

Esta utilidad está muy bien, cuando estamos desarrollando ya que podemos depurar de una forma controlada nuestra aplicación; sin embargo, el mayor problema que nos encontramos es el momento en el que tenemos que la aplicación está en el cliente, este nos reporta un error y no conseguimos dar con la tecla. Y claro está, no le vamos a decir que abra las herramientas de desarrollo y que nos muestre los mensajes que hay en la consola de depuración. Existen librerías/soluciones complejas en la que todos los errores se guardan en una base de datos local y posteriormente esta se sincroniza con una base de datos central.

Para evitar todo este sobre coste y para tener un control de errores de cualquier aplicación vamos a utilizar Application Insight.

Application Insights

Application Insight es un producto PaaS ofrecido dentro de Azure, que se puede utilizar en todos los desarrollos que nos planteemos. Tiene un SDK para casi cualquier plata-

forma y si no, también dispone de una API REST con la que interactuar.

Además de ser una herramienta para almacenar los log, se puede establecer un mecanismo para saber que elementos de tu aplicación son más usados o menos. De esta forma obtenemos importante información sobre los gustos del usuario. Lo cual se puede utilizar para tomar decisiones sobre la evolución del producto/solución: mejorar aspectos más usados, eliminar características que no se utilizan, etc.

Además, otra de las ventajas que tiene es la posibilidad de extraer los datos obtenidos e integrarlos con otras herramientas como pueda ser Power BI para hacer informes personalizados.

Como empezar a utilizarlo

En primer lugar, vamos al portal de Azure y creamos un nuevo recurso de Application Insights (en Azure un recurso es una instancia de un servicio). Para ello seleccionamos Servicio Para Desarrolladores -> Application Insight.

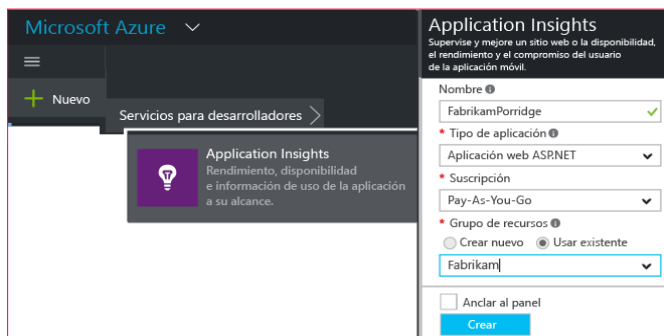


Imagen 2.- Crear servicio de Application Insight.

Donde pone Tipo de Aplicación, en nuestro caso deberemos de poner Otro tipo de Aplicación. Esta elección solamente afecta a los paneles por defecto que se visualizarán cuando se empiecen a capturar los datos de la aplicación.

Una vez el servicio se a aprovisionado, deberemos guardarnos la Key (que es lo que identifica nuestro servicio). Esta clave la podemos encontrar en la siguiente pantalla:

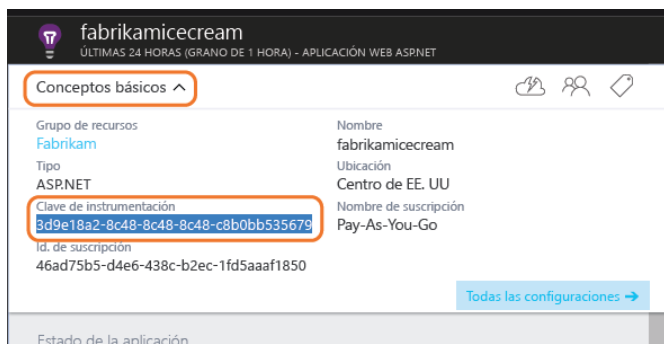


Imagen 3.- Obtener la Key de Application Insights.

A continuación, tendremos que instalar Applitation Insights en nuestra aplicación. Para ello cuando estamos en un entorno de .NET hacemos uso de Nuget como gestor de las librerías externas. Dentro del mundo JavaScript se hace uso (de momento) de NPM, que es un gestor de paquetes de NodeJS. Por lo que para instalar Application

Insight dentro de nuestro desarrollo, nos deberemos ir a la biblioteca donde tenemos la solución y ejecutar la siguiente instrucción:

```
>npm install applicationinsights --save
```

Una vez ejecutado automáticamente dentro del fichero package.json ha debido añadir la siguiente línea "applicationinsights": "^0.18.0"

"A pesar de que existe una gran cantidad de documentación sobre el SPFx hay algunos elementos que existen, pero los desarrolladores no los utilizamos debido a que no los conocemos"

Application Insight lo podemos utilizar de dos formas: una en la que el propio Application Insight obtiene los datos de forma automática o la forma más normal en la que nosotros le indicamos que aspectos debe de guardar para analizar posteriormente.

Para el primer caso bastaría con añadir en el siguiente código:

```
var appInsights = require("applicationinsights");
appInsights.setup("<instrumentation_key>").start();
```

En el segundo caso, tendríamos que añadir el siguiente código:

```
import appInsights = require("applicationinsights");
var client = appInsights.getClient();
client.trackEvent("custom event", {customProperty: "custom property value"});
client.trackException(new Error("handled exceptions can be logged with this method"));
client.trackMetric("custom metric", 3);
client.trackTrace("trace message");
```

Dependiendo del tipo de evento que invoquemos este se visualizará en un panel u otro de Application Insight.

Consideraciones a tener en cuenta

Este SDK por defecto hace peticiones REST a dos url:

- dc.services.visualstudio.com:443
- f5.services.visualstudio.com:443

Por lo que en caso de que en el entorno en el que lo estamos utilizando tenemos que tener en cuenta que estas direcciones no estén bloqueadas por ningún "firewall".

¿Cuándo utilizo el LOG de SPFX y cuando Application Insight?

Desde mi punto de visto se deben utilizar ambos, la solución que implementamos nosotros es crearnos una Interfaz LOG en la que tiene los métodos más comunes de un LOG por ejemplo:


```
interface ILog {
    Error(Source:string, Exception:any);
    Info(Source:string);
    Warm(Source:string);
    Verbose(Source:string);
}
```

Con esta interfaz, implementaremos una definición del LOG que consuma el Log de SPFx y posteriormente implementaremos una definición del LOG que tenga ambas. La implementación del LOG de SPFx la implementaremos la utilizaremos cuando nuestro desarrollo este en fase de desarrollo, ya que es mucho más sencillo para la persona que está desarrollando. Mientras que Application Insights lo vamos a utilizar cuando la aplicación ya esté en un entorno productivo para poder extraer información de lo que sucede cuando la aplicación la está utilizando otros usuarios.

Para hacer esta distinción utilizaremos la clase EnvironmentType que viene de serie, para saber cuándo estamos en el Workbench local o bien cuando estamos en un Tenant de Office 365. Para utilizar esta clase tendremos que añadir el siguiente código:

```
import {
    Environment,
    EnvironmentType
} from '@microsoft/sp-core-library';
```

Para saber en qué entorno esta tendremos que poner el siguiente código:

```
if (Environment.type === EnvironmentType.Local) {
```

¿Por qué motivo utilizar una Interfaz y no consumir directamente la clase?

En primer lugar, es por tener desacopladas las clases y no tener una dependencia de ninguna librería externa (lo cual en un futuro nos puede costar un quebradero de cabeza). Muchas veces hacemos uso de una librería de terceros y la tratamos como si fuera nuestra. Pero qué pasa si esta librería se deja de actualizar por su creador o bien encontramos otra librería que funciona mejor. La respuesta es clara, en la mayoría de casos continuamos dependiendo de la vieja clase hasta que es estrictamente necesario y cuando cambiamos proyecto ya actualizamos a la nueva clase que hemos decidido.

Resumen

Como estamos comprobando, este nuevo modelo de desarrollo a pesar de estar en una fase muy temprana en su evolución, es bastante completo y tiene muchas utilidades requeridas por los “Developers”. A todo esto, si se complementa con todos los servicios que ofrece el Cloud observamos que estamos en un tiempo muy divertido e interesante para los desarrolladores.

Uno de los puntos débiles a lo largo del tiempo, es el control de errores en la parte Front-End de la aplicación ahora con todas estas herramientas de las que disponemos NO hay excusa para no hacerlo.

ADRIÁN DIAZ CERVERA - ARCHITECT SOFTWARE LEAD AT ENCAMINA

MVP Office Development

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

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

adiaz@encamina.com @AdrianDiaz81

Emulando DocumentDB

Cuando desarrollamos aplicaciones que serán hospedadas en Azure y utilizarán sus servicios, son de mucha utilidad los emuladores. Estos nos permiten desarrollar en nuestro entorno local sin necesidad de tener que montar la infraestructura necesaria en Azure, así como no depender de tener conexión para realizar los desarrollos.

Uno de los últimos emuladores que nos ha proporcionado Microsoft es el emulador de DocumentDB, en las siguientes líneas vamos a ver cómo podemos instalarlo y utilizarlo.

Instalación

Lo primero que deberemos hacer es bajarnos el instalador (<https://aka.ms/documentdb-emulator>) e instalarlo. Para poderlo instalar es necesario que nuestro equipo cumpla las siguientes características:

- Software: Windows Server 2012 R2, Windows Server 2016 o Windows 10
- Hardware: 2 Gb de RAM y 10 Gb de espacio en el disco duro.

Una vez instalado lo podemos ejecutar:

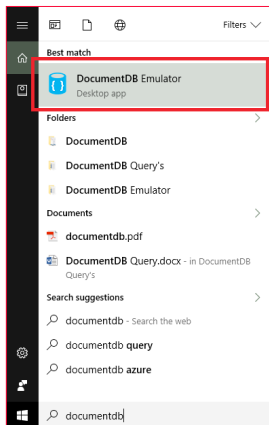


Imagen 1.- Búsqueda del emulador.



Figura 2.- Emulador ejecutándose.

“Emuladores nos permiten desarrollar en nuestro entorno local sin necesidad de tener que montar la infraestructura necesaria en Azure, así como no depender de tener conexión para realizar los desarrollos”

Administración

Cuando ejecutamos el emulador, automáticamente se nos abre el panel de control de DocumentDB en el navegador. Por defecto, la dirección del panel de control del emulador es https://localhost:8081/_explorer/index.html, donde podemos crear nuestras colecciones, hacer queries o bajarlos quickstarts.

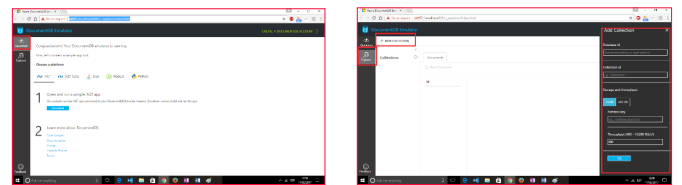


Imagen 3.- Quick Starts / Crear una colección.

También podemos administrar nuestro DocumentDB local desde la línea de comando. Para ello debemos abrir la línea de comando desde la carpeta donde tengamos instalado el emulador y escribir `DocumentDB.Emulator.exe` seguida de la opción que deseemos, para saber las opciones escribiremos `DocumentDB.Emulator.exe /?`

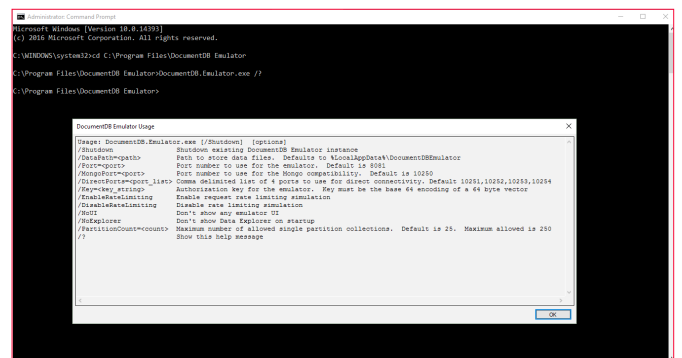


Imagen 4.- DocumentDB emulator command-line.

¿Qué podemos hacer?

- Crear documentos (recordemos que los documentos de DocumentDB son en formato JSON).
- Realizar query's sobre los documentos.
- Crear colecciones.
- Escalar colecciones.
- Ejecutar stored procedures y triggers.

¿Qué NO podemos hacer?

- Simular la replicación.
- Latencia de milisegundos de un solo dígito para lecturas/escrituras.

- Al no tener replicación no se nos permite seleccionar el nivel de consistencia.

Como podemos observar lo que podemos y no podemos hacer tiene su lógica dado que debemos tener en cuenta que estamos ejecutando nuestro DocumentDB sobre una máquina local.

Ahora bien, ¿Qué diferencias nos encontramos entre el emulador y el servicio de Azure?

Diferencias que nos encontramos entre el emulador y el servicio de Azure

- El emulador solo soporta una cuenta y una clave, además está no puede ser regenerada.
- El emulador no es un servicio escalable y no permite un gran número de colecciones, por defecto 25 colecciones o 1 colección con partición. El máximo son 250 colecciones o 10 colecciones particionadas.
- Como ya hemos comentado anteriormente, el emulador no nos permite simular los diferentes niveles de consistencia ni nos permite la replicación.
- El emulador puede no contener todas las últimas novedades que tenga el servicio de Azure.

En este punto ya tenemos una idea de cómo funciona, si bajamos uno de los quickstart, veremos cuál es el endpoint y la key de nuestro emulador.

En este caso utilizaremos el quickstart de .NET. Si vamos al web.config en la sección appSettings veremos la configuración de conexión a nuestro emulador:

```
<add key="endpoint" value="https://localhost:8081/" />
<add key="authKey" value="C2y6yDjf5/R+ob0N8A7Cgv30VRD-
JIWEHLm+4QDU5DE2nQ9nDuVTqobD4b8mGGyPMbIZnqyM-
sEcaGQy67XlW/Iw==" />
```

El ejemplo que nos baja es el típico ToDoList, si añadimos un elemento a nuestra lista veremos que se nos crea la colección y se inserta el documento en el emulador:

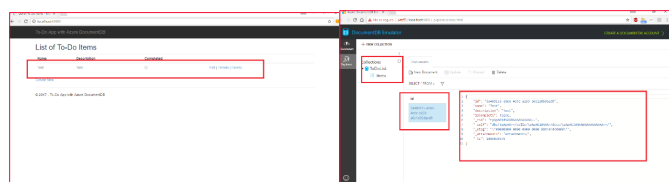


Imagen 5.- Utilizando el Emulador.

Ahora si quisiéramos usar el servicio real de DocumentDB, lo único que debemos hacer es crearlo en nuestra suscripción, cambiar los settings y listo.

```
<add key="endpoint" value="https://yourdocumentdb.docu-
ments.azure.com:443/" />
<add key="authKey" value="CdHe9nEjeIVLn5...yourkey" />
```

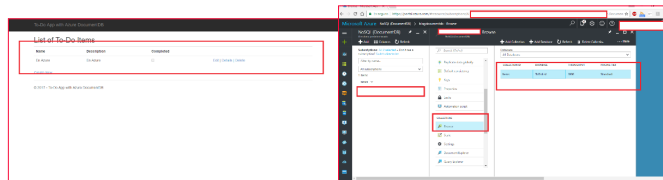


Imagen 6.- Haciendo uso real del servicio de DocumentDB.

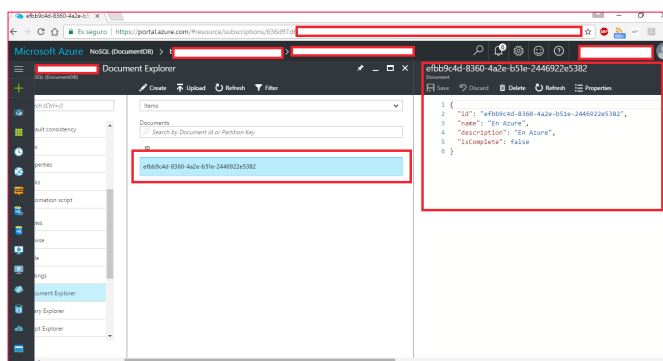


Imagen 7.- Uso del Document Explorer para DocumentDB.

Conclusiones

Utilizar el emulador es fácil y sencillo, y utilizarlo es tan similar al servicio que hace que su utilización se intuitiva si estás acostumbrado a trabajar con DocumentDB. Y en el caso de no haberlo utilizado, es una forma fácil y sencilla en adentrarse en este servicio y que el paso a Azure se transparente.

REFERENCIAS

<https://docs.microsoft.com/en-us/azure/documentdb/documentdb-nosql-local-emulator>

ROBERT BERMEJO

Arquitecto .Net & Azure lead en Sogeti España, SLU

roberto.bermejo@sogeti.com

@robertbermejo

www.robertbermejo.com

Introducción a FastTrack Network Analysis para Office 365

Para comenzar a utilizar FastTrack Network Analysis para Office 365 no tenéis más que acceder a la dirección Url <http://em1-fasttrack.cloudapp.net/o365nwtest>. Al navegar a esta Url os encontraréis con la primera sorpresa “inesperada” y es que FastTrack Network Analysis para Office 365 está disponible por medio de un applet de Java lo que en la práctica se traduce en que sólo podréis ejecutar la herramienta desde Internet Explorer o Mozilla FireFox. Si tratáis de ejecutarla desde Google Chrome o Microsoft Edge, simplemente no podréis.

- Para comenzar a utilizar FastTrack Network Analysis para Office 365, hay que indicar en primer lugar la Url del tenant de Office 365 al que nos vamos a conectar para realizar la evaluación de las características de nuestra red corporativa.

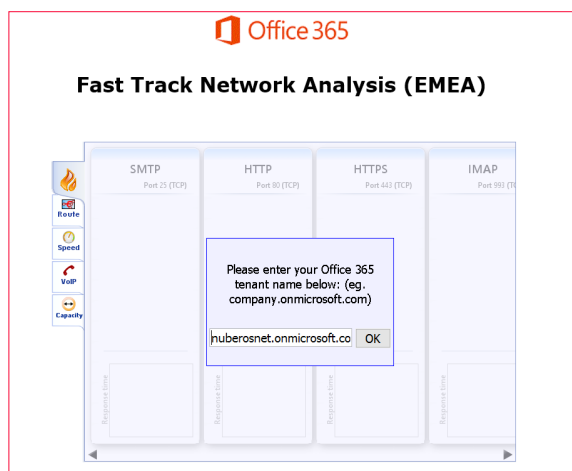


Imagen 1.- Configuración inicial de FastTrack Analysis Tool.

“...una herramienta proporcionada por el equipo de FastTrack de Microsoft que permite diagnosticar si las características de red de una organización son adecuadas para acceder a los servicios de Office 365...”

- La herramienta verifica en primer lugar que a nivel de puertos necesarios para conectarse a Office 365 todo está correcto. Por ejemplo, se comprueba si se permite, entre otros, la conexión a través de los puertos de HTTP / HTTPS (80 / 443), IMAP / POP (993 / 995). Para cada puerto se muestra el tiempo de respuesta.

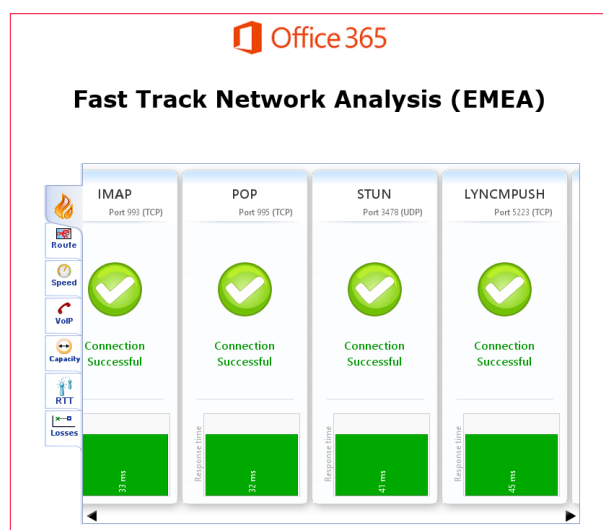


Imagen 2.- Verificación de conectividad a Office 365 a través de los puertos requeridos.

- A nivel de ruta seguida, FastTrack Network Analysis determina la ruta que siguen las peticiones que se realizan desde la red corporativa a Office 365 mostrando información detallada sobre los saltos realizados, % de paquetes perdidos en la conexión o si la resolución a nivel de DNS es adecuada.

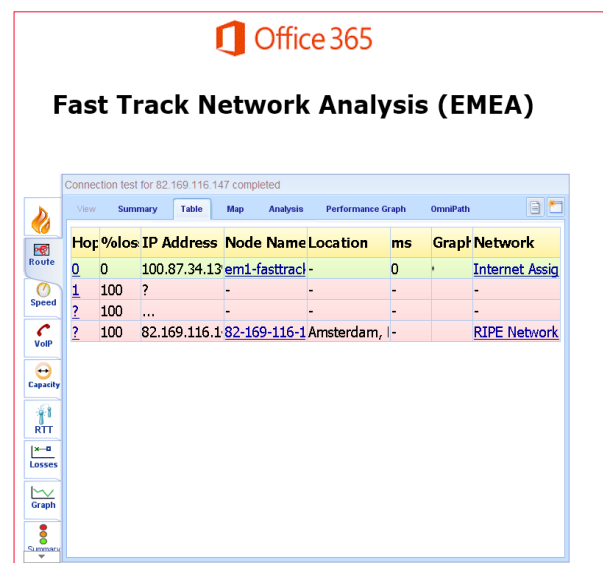


Imagen 3.- Análisis a nivel de ruta seguida a Office 365.

- La tercera pestaña de FastTrack Network Analysis muestra datos relativos a la velocidad de subida y de bajada de nuestra conexión a Internet, así como los valores de parámetros como el Round Trip Time (RTT), retardo máximo, conexiones usadas, etc.

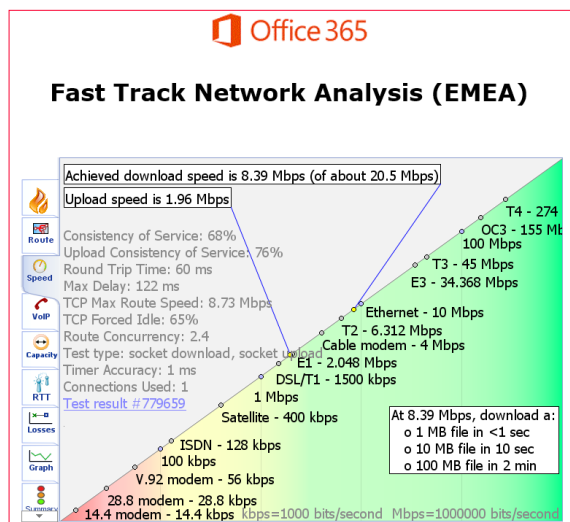


Imagen 4.- Velocidad de subida y de bajada.

- La sección VoIP permite determinar si nuestra red está preparada para en cuanto a Jitter y perdida de paquetes para comunicaciones de tipo voz sobre IP haciendo uso de Skype for Business.

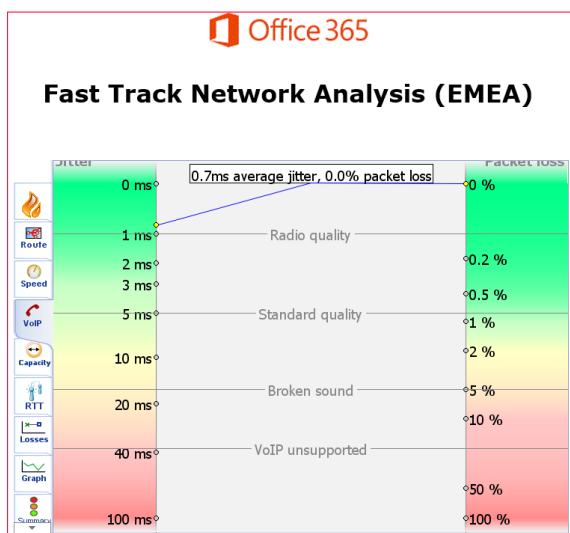


Imagen 5.- Sección VoIP.

- La capacidad de la red tanto en subida de información como en bajada se muestra en la sección de "Capacity".

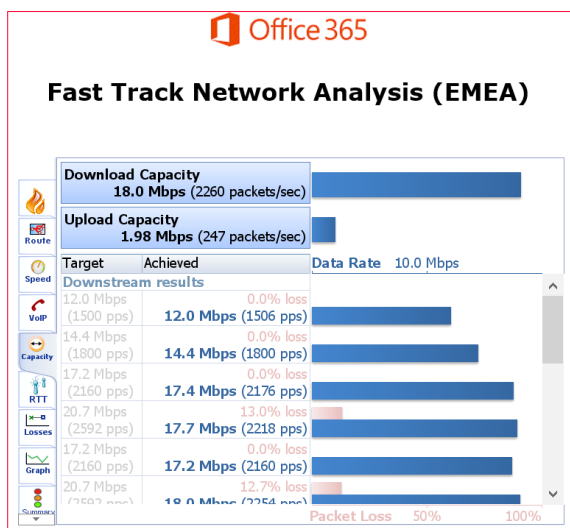


Imagen 6.- Resultados en la sección Capacity.

- La sección RTT muestra el valor medio de RTT y si dicho valor medio está en los parámetros esperados para tener una buena conectividad con Office 365.

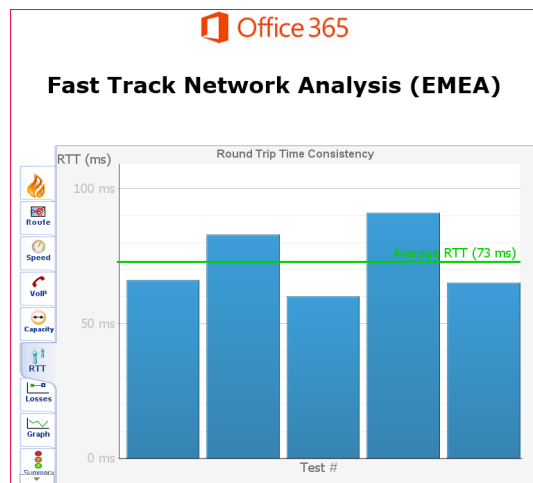


Imagen 7.- Valor medio de RTT.

- FastTrack Network Analysis permite obtener el % de paquetes perdidos tanto en subida como en bajada a través de los resultados que se muestran en la sección "Losses".

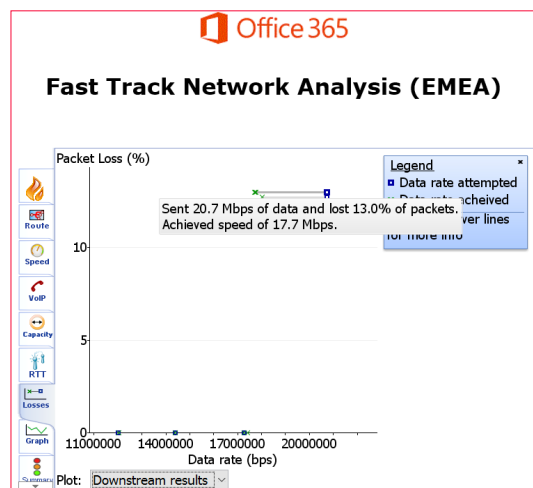


Imagen 8.- Resultados de paquetes perdidos en tráfico de bajada.

- En la pestaña "Graph" se muestran distintos gráficos como la velocidad de transferencia frente al retardo experimentado, el valor del Jitter frente al % de paquetes perdidos, etc.

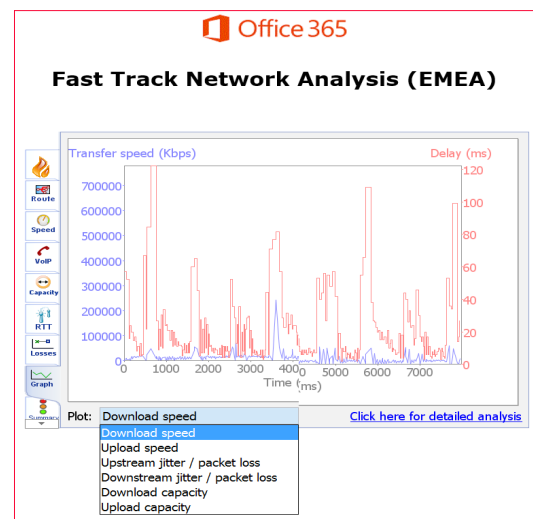


Imagen 9.- Sección "Graph".

- En la sección “Summary” se muestra un resumen de los resultados obtenidos con FastTrack Network Analysis. Por cada tipo de análisis realizado se muestra un enlace a la sección correspondiente y un semáforo que identifica si hay algún tipo de problema.

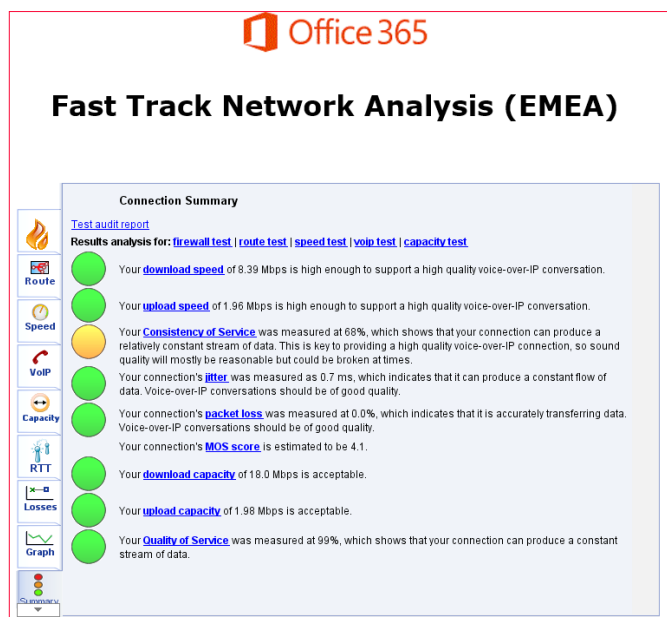


Imagen 10.- Resumen de resultados obtenidos.

- Finalmente, la sección “Advanced” muestra estadísticas avanzadas relativas a parámetros de la calidad de red para conectarse a Office 365.



Imagen 11.- Estadísticas avanzadas en el FastTrack Network Analysis.

Conclusiones

FastTrack Network Analysis es una herramienta proporcionada por el equipo de FastTrack de Microsoft que facilita el análisis de la conectividad de red y su idoneidad para el acceso a los servicios que conforman Office 365.

JUAN CARLOS GONZÁLEZ MARTÍN

Office Servers and Services MVP

Cloud & Productivity Advisor

jcgonzalezmartin1978@hotmail.com

@jcg1978 | <https://jcgonzalezmartin.wordpress.com/>

System Center: Una solución integral para la gestión de la nube privada

Muchos de nosotros hemos oído hablar en innumerables ocasiones a terceros de las maravillas de la gestión de sistemas basada en un marco de trabajo como ITIL, COBIT, u otro de los disponibles en el mercado actual. Se nos habla de mejoras sustanciales en el alineamiento del departamento de IT con el negocio, una mejora de la satisfacción de los clientes internos de la empresa y de una mejora del aprovechamiento de los recursos técnicos y de personal dentro del departamento técnico.

Está claro que cualquier coordinador de departamento o cualquier administrador de sistemas quisiera esto para sí mismo y su empresa, pero muchas veces, los que estamos más del lado técnico que del lado de la gestión no tenemos claro cómo llevar a cabo la adaptación de nuestra empresa y por tanto de nuestro departamento de IT, ni qué debemos hacer para implementar uno de estos Frameworks para el beneficio de nuestra empresa.

A lo largo del presente texto intentaremos transmitir al lector las nociones básicas sobre la gestión de sistemas para a continuación hablar sobre el marco de trabajo propuesto por Microsoft, uno de los principales fabricantes de software del mundo, el Microsoft Operations Framework (MOF) y terminaremos repasando las opciones que dicho fabricante nos da si queremos gestionar entornos de sistemas basados en Wintel o en entornos heterogéneos basados en sistemas Windows y UNIX.

¿Un marco de trabajo de un fabricante?

Microsoft Operations Framework (MOF) es por así decirlo una piel de ITIL, o al menos una expansión sobre el mismo, ya que uno de los equipos de trabajo de Microsoft partió de éste para desarrollar un marco de trabajo para la gestión que se adaptara mejor a sus productos.

Como su propio nombre indica, Microsoft Operations Framework es en realidad una guía de buenas prácticas para la operación de IT, por lo que no es tan extenso ni tan detallado como ITIL y no pretende sustituirlo, sino adaptarlo y expandirlo. Podemos decir también que ITIL es el modelo de referencia mientras que MOF es el modelo de implementación.

MOF, es en realidad una colección de recomendaciones, principios y modelos basada en un conjunto de artículos

y documentos de en los que Microsoft describe qué tenemos que hacer, cuándo y por qué.

La última versión a fecha de hoy es la v4.0 que se basa en ITIL 2011.

Pero, ¿Por qué debemos adoptar un marco de trabajo diseñado por Microsoft? Lo primero que debemos pensar es que está diseñado por Microsoft, pero no sólo es aplicable a sus sistemas o a infraestructuras diseñadas íntegramente con sistemas de dicho fabricante. Es más bien todo lo contrario, es adaptable a cualquier tipo de explotación, sea cual sea el sistema operativo que usen y sean cuales sean sus aplicaciones o incluso plataforma de trabajo.

MOF se basa en los principios de dos modelos, el modelo de equipo y el modelo de proceso. Debemos tener en cuenta que los modelos se dividen en tres categorías fundamentales: Personas, Procesos y Tecnología, división que ya conocemos por ITIL y que como ya sabemos, las categorías de personas y procesos se definen expresamente mientras que la tecnología no es independiente, sino que se extiende por MOF.

MOF es ideal para la optimización de procesos del departamento técnico de la empresa, ya que ayuda a identificar las personas y los roles que desempeñan, facilita la identificación de los riesgos y a controlarlos mediante planes de contingencia. Además ayuda con la estandarización de tareas, actividades y el establecimiento de procedimientos.

Vamos a darnos un pequeño paseo por este Framework para tener algunas nociones sobre el mismo y además intentar entender el porqué de su adaptación en los productos de los que hablaremos posteriormente.

Uno de los puntos más fuertes de este marco de trabajo es que tiene en cuenta la naturaleza cambiante y evolutiva de los departamentos de IT y nos ayuda a gestionar estos cambios a través de herramientas aplicables desde el primer momento y de manera progresiva, sin necesidad de implementar todas para que se logre el cambio, sino que se puede hacer paso a paso y únicamente hasta donde nos interese y a nuestro ritmo.

Otro de los puntos en los que radica el gran interés de estas recomendaciones es que a diferencia de otros marcos de trabajo, MOF contiene información que es aplicable directamente, es reutilizable entre departamentos y empresa y

además se puede adaptar según las necesidades de nuestra empresa.

El último punto por el que me parece de gran interés es porque MOF es compatible con otros estándares como el ISO-IEC 20.000, el framework de control COBIT y por supuesto con ITIL.

A continuación vamos a hablar un poco sobre la composición de MOF.

Este marco se basa en tres frases que son:

- Planificación (Plan).
- Entrega (Delivery).
- Operación (Operation).

Además se soportan sobre la capa de gestión (Manage) que da cobertura a las tres anteriores.



Imagen 1.- Las tres fases de M.O.F. 4.0.

Cada fase establece unos objetivos, metas y responsabilidades y se divide en “Service Management Functions” (SMF) que proveen los detalles de las actividades que se engloban dentro de cada una de las tres fases. En cada fase podemos encontrar los “Management Reviews” (MR) que se encargan de decidir en qué momento se puede dar una fase por concluida y con los test que establecen, nos permiten incrementar la calidad de nuestra gestión de la fase de operación del servicio. Los MR harían la función de objetivos de nivel de servicio (SLO) y nos permitirán comprobar en cada fase si se han cumplido para poder liberar la siguiente fase.

Una vez concluido todo el ciclo de planificación, entrega y operación se comienza un nuevo ciclo, también es posible que nos encontremos con equipos distintos que están trabajando simultáneamente en distintas fases.

Vamos a hablar brevemente de cada una de las fases para dar al lector una idea global de qué se hace en cada una de ellas y cómo funcionan.

En la Fase de Planificación obtenemos una guía de cómo planificar para optimizar la estrategia del servicio. Dentro de ésta tenemos cuatro SMF y dos MR. Del lado de las “Management Reviews” tenemos el alineamiento de servicio y el portfolio, mientras que del lado de las “Service Management Functions” tenemos el alineamiento del negocio, confiabilidad, políticas y administración financiera.

El alineamiento del negocio con IT estará compuesto de

una estrategia del servicio, un mapeo del servicio, gestión de la demanda, gestión de niveles de servicio (SLA) y gestión del portfolio. La confiabilidad incluye las gestiones de capacidad, continuidad, confidencialidad, integridad y disponibilidad. Las políticas incluyen seguridad, privacidad, uso apropiado de la información, gestión de socios y protección de activos. Por último, la administración financiera consiste en saber dónde se deben invertir los recursos.

En la Fase de Entrega se ayuda a los profesionales de IT a diseñar y entregar de forma más eficiente los servicios de infraestructura y los despliegues de productos. En este caso tenemos dos SMF y cinco MR. Los SMF son tener un plan de proyecto aprobado y tener preparado el lanzamiento, mientras que del lado de los MR tenemos la visión, el plan de proyecto, la construcción, la estabilización y el despliegue.

La visión consiste en tener claro lo que se quiere hacer. El plan de proyecto nos dice qué queremos hacer y cómo queremos hacerlo. La construcción del proyecto representa las tareas para llevar a cabo la ejecución del proyecto en sí. La estabilización consiste en dejar estable lo que se ha hecho hasta el momento, algo parecido a parametrizarlo correctamente. Por último, el despliegue consiste en realizar la propia entrega que dará por finalizada la fase de entrega.

La Fase de Operación ayuda a los técnicos de IT a operar, monitorizar y dar soporte a los servicios que se prestan al cliente, de manera que se cumpla con los acuerdos de nivel de servicio (SLA) pactados con el mismo. En esta fase tenemos un “Service Management Functions” y cuatro “Management Reviews”. El SMF consiste tener en todo momento el servicio con un nivel de salud conforme a los SLO. Entre los MR tenemos la operación, la monitorización del servicio, el servicio al cliente y la gestión de problemas.

La operación consiste básicamente en realizar las tareas necesarias para el correcto funcionamiento del servicio. La monitorización del servicio se encarga de llevar a cabo el seguimiento del mismo para validar que no se produce ninguna situación que ponga en peligro el correcto funcionamiento del servicio. El servicio al cliente incluye el Service Desk y la gestión de incidentes. Por último tenemos la gestión de problemas.

Tal como se ha dicho anteriormente, por encima de todo ello y en continuo contacto con todas las fases tenemos una Capa de Gestión. Esta capa establece el proceso de toma de decisiones, la gestión del cambio, la gestión del riesgo y los controles necesarios a lo largo de todo el ciclo de vida. En esta capa tenemos un “Service Management Functions” y tres “Management Reviews”. Por un lado tenemos las políticas y el control, mientras que por el otro tenemos la gestión del riesgo, de los acuerdos, de los cambios y configuraciones y del equipo de trabajo.

Todo esto lo podemos ver representado en la siguiente figura:



Imagen 2.- Visión general de M.O.F. 4.0.

No es objeto de este documento seguir profundizando en MOF por lo que para aquellos que tengan cierto interés en saber más sobre este marco de trabajo, pueden encontrar más información en la Web del proyecto y para los adictos a las certificaciones existe una en EXIN cuyo nombre es “Microsoft Operations Framework Foundation”.

¿Qué puede hacer system center por ti?

Creo que con lo que se ha expuesto hasta ahora en la sección de MOF se da una idea bastante general de cómo funciona esta piel de ITIL. El problema que tenemos es que seguimos sin saber cómo llevar esto de la teoría a la práctica. Pues bien, en el “IT Pro Quick Start Kit” de MOF 4.0 tenemos guías y esquemas de cómo podemos realizar la implementación.

Además, para realizar una gran parte de estas tareas tenemos algunas suites de trabajo, pero para mí la más completa y más sencilla de gestionar es Microsoft System Center, que el año pasado lanzó la versión 2016 con bastantes novedades para agilizar su gestión y facilitar la vida al personal de informática. Un punto a favor de esta suite es que dentro del citado “IT Pro Quick Start Kit” ya se hace referencia a la misma.

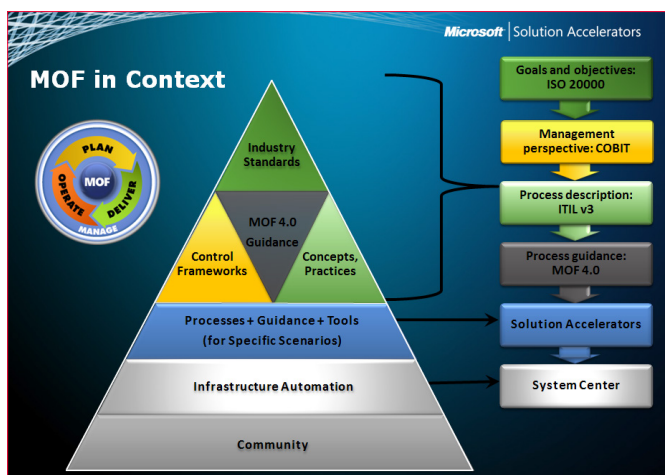


Imagen 3.- Contexto de M.O.F. 4.0.

Como siempre, debemos comenzar por una definición, pero claro, al igual que siempre, es difícil condensar en una sola frase o párrafo todo lo que queremos transmitir. Si tenemos que definir brevemente la suite Microsoft System Center podremos resumir su definición como una plataforma integrada para la administración, gestión y automatización de las tareas comunes del departamento de sistemas, lo que se viene a llamar herramienta de gestión de extremo a extremo. También podemos definirla como una herramienta para la adaptación de las mejores prácticas para la gestión del departamento de IT.

Para poder definir correctamente todo el producto vamos a desglosar uno a uno los productos que componen la suite para dar una visión global del producto.

Una solución completa para el service desk

Dentro de Microsoft System Center, Service Manager es la herramienta que el gigante de Redmond nos presenta como solución óptima para la gestión del Help Desk o mejor dicho del Service Desk ya que hace muchas más cosas de las que haría un simple Help Desk. Es en este producto en el que más me voy a extender al tratarse del más relacionado con el ITSM.

Service Manager es una herramienta completa que incluye una base de datos de gestión de la configuración (CMDB), control de versiones, control del cambio, gestión de la configuración y por supuesto procesos diversos para la resolución de incidentes, resolución de problemas.

Además, Microsoft System Center Service Manager proporciona capacidades de integración con otros sistemas de la explotación para asegurar una optimización de las funciones anteriormente descritas y para un mejor alineamiento con el negocio. Éste puede conectarse con otros miembros de la suite System Center como Operations Manager, Orchestrator, Configuration Manager o con sistemas propios de la infraestructura troncal como el Directorio Activo, el Servicio de correo, etc.

Entre las funcionalidades de integración y alineación con el negocio podemos destacar:

- Proporcionar una base de conocimiento real y útil: todos los usuarios, servidores, electrónica de red e incluso periféricos o incluso componentes quedan registrados en la CMDB para una mejor identificación de los mismos y una gestión de incidencias más ágil.
- Proporciona un punto de contacto único: Los usuarios dispondrán de un punto de contacto único para sus consultas y peticiones al departamento de IT, mejorando así la relación del departamento con sus clientes internos. En este portal de comunicación se pueden incluir otras formas de comunicación como mensajería instantánea o telefónica con los miembros de primer nivel.
- Proporciona reducción de costes: Aumentando la ca-

pacidad de respuesta a través de una herramienta central para gestión de todas las incidencias; conseguimos además aumentar la satisfacción del usuario.

- Proporciona automatización de procesos: La integración con el resto de herramientas de System Center nos permite la automatización de tareas relacionadas con la gestión de eventos.
- Proporciona un mejor alineamiento con el negocio: A través de la gestión de la configuración, de la gestión de riesgos, de la generación de informes, de la gestión de capacidades y escalabilidad.
- Proporciona gestión de los acuerdos de nivel de servicio (SLA): A través de herramientas integradas con el resto de la infraestructura, SCSM nos permite establecer objetivos de nivel de servicio (SLO), la revisión de la conformidad con los mismos y generación de informes asociados.

Un vigilante para nuestra explotación

Otro producto ofrecido por Microsoft System Center es Operations Manager, conocido comúnmente entre los administradores como SCOM. Es la solución de Microsoft para la monitorización integral de nuestra explotación. Esta solución proporciona monitorización para sistemas operativos como Microsoft Windows, GNU/Linux o UNIX, elementos de red como switches y routers de distintos fabricantes, bases de datos y una amplia gama de aplicaciones tanto de Microsoft como de otros fabricantes.

Otra de las grandes ventajas de SCOM es su capacidad de adaptación y modificación, permitiendo programar nuestros propios paquetes para la monitorización de aplicaciones diseñadas a medida para nuestra empresa. Además, nos facilita la interacción y adquisición de datos con otros gestores de monitorización como Nagios o Cacti.

El último punto a tener en cuenta es que Microsoft System Center Operations Manager es un producto con una alta capacidad de integración con el resto de programas de la suite, por lo que podrá configurarse para tomar datos sobre los elementos de configuración del Directorio Activo y de SCCM, generar incidencias en SCSM de manera automática cuando detecte eventos que puedan afectar al servicio o enviar correos usando el Exchange Server.

Provisión automatizada

El producto estrella dentro de la suite de gestión de Microsoft System Center es el famoso Configuration Manager, conocido por sus siglas como SCCM. Es una solución completa para la provisión de software, parches, actualizaciones e incluso para la instalación automática de sistemas operativos completos.

Por si fuera poco todo lo citado anteriormente, SCCM es un pilar fundamental para el resto de la suite ya que es capaz de obtener grandes cantidades de información sobre los elementos de configuración a través de los agentes insta-

lados en ellos. Sólo por citar un ejemplo, en un servidor, SCCM podrá sacar información sobre la memoria RAM, el microprocesador, las tarjetas de red, discos duros, etc.

Toda la información recopilada y generada por Configuration Manager se pasa al resto de productos que la usan para su propio funcionamiento. SCOM toma los datos recopilados para generar informes de disponibilidad mientras que SCSM la utiliza como elementos de configuración e información adicional para la CMDB mientras que VMM y CSCM utilizarán también las capacidades de provisión para el portal de gestión autoservicio del que se hablará en la parte dedicada a Virtual Machine Manager.

Automatización de procesos

Una de las tendencias que existe hoy en las empresas es la de automatizar todo lo que se pueda para reducir costes tanto de personal como de infraestructura y operación. Para esto, también tiene una solución Microsoft System Center, en este caso se trata de System Center Orchestrator, llamado previamente Opalis por la empresa que lo desarrolló en origen y a la cual se lo compró Microsoft.

Los administradores de sistemas y técnicos asociados al área de tecnologías de la información realizan a lo largo del día muchas tareas y procesos para mantener su entorno de trabajo y por tanto la empresa en correcto funcionamiento. Entre todas estas tareas tenemos un gran número de ellas que son tareas monótonas y repetitivas. Ejemplos de tareas comunes y repetitivas son dar acceso a los usuarios a un repositorio de datos, darles cuentas de correo, cuentas de usuario del dominio.

“algunas de las herramientas con las que afrontar las necesidades de gestión de nuestra infraestructura de la nube privada para además cumplir con un marco de trabajo alineado con la ISO 20.000, ITIL 2011 y con Microsoft Operations Framework”

Estas tareas pueden ser automatizadas a través de su subdivisión en tareas más pequeñas que su vez son automatizables. Orchestrator es una herramienta creada para el diseño de procesos automatizados basada en runbooks, con lo que estas tareas descompuestas pueden ser tomadas como procesos que se ejecutarán uno tras otro hasta lograr el resultado deseado.

Con la ayuda de System Center Orchestrator se pueden realizar las siguientes tareas:

- Automatizar gran parte de las operaciones del departamento de TI.
- Estandarizar los procesos operativos mediante la aplicación de las mejores prácticas.

- Automatizar procesos de manera independiente a la plataforma o al hardware que lo hospeda.
- Conectar distintos sistemas en entornos heterogéneos sin tener que preocuparse de bajar al nivel de los lenguajes de programación ni comandos complejos.

Salvaguarda de nuestros datos

Un elemento que no puede faltar dentro de una suite de gestión como Microsoft System Center es un método para realizar las copias de seguridad. En este caso la herramienta recibe el nombre de Data Protection Manager también conocido como DPM.

Es un programa de backup bastante sencillo, no comparable a otras herramientas como por ejemplo Veritas Net-Backup, aunque presenta las opciones necesarias para gestionar las copias de seguridad de manera eficiente. Es una solución más que valida para la copia de datos de repositorios, de bases de datos e incluso de los estaos de máquinas físicas y máquinas virtuales.

Es compatible con múltiples tipos de almacenamiento incluidas las cabinas de disco y las cabinas de cintas LTO por lo que es una solución más que viable para los planes de contingencia recomendados en las best practices de todas las metodologías de trabajo incluyendo ITIL, MOF 4.0 o las ISO-IEC 20.000 y 27000.

Gestión de nuestra nube privada

Todo lo que hemos visto hasta ahora de Microsoft System Center ha sido un conjunto de herramientas diseñadas para la gestión de una infraestructura informática de propósito general, pero hasta ahora no hemos visto nada que haya destacado por ser un producto que podamos decir que es un producto diseñado para la gestión de una nube privada, sino para cualquier infraestructura sin necesidad de utilizar ningún grado de virtualización.

Pues bien, ya no podremos decir eso, tan pronto como conozcamos la siguiente herramienta de Microsoft System Center. Virtual Machine Manager, conocido entre los administradores de sistemas como VMM por sus siglas es un producto diseñado íntegramente para la gestión de entornos virtualizados sin importar el grado de virtualización de los mismos ya que puede gestionar desde entornos poco virtualizados hasta entornos enteramente virtualizados.

Uno de los puntos que hacen interesante a SCVMM es su capacidad de gestionar máquinas virtuales, no sólo de Microsoft Hyper-V sino también de otros fabricantes como Citrix o VMWare que hasta ahora se han llevado la mayoría del mercado, aunque con las nuevas funcionalidades de Windows Server 2016 se espera que esta tendencia cambie.

El último punto por el que VMM es una muy buena herramienta de gestión y que no puede faltar en nuestra empresa para la administración de nuestra nube privada es por

el portal de gestión autoservicio. El portal de gestión de autoservicio trabaja junto a SCCM para dar una respuesta inmediata a las necesidades de los departamentos de la empresa. Permite administrar el número de máquinas virtuales destinadas a un proyecto, igual que la capacidad de memoria, procesador, red, disco, etc. que éstas destinan a servir a una aplicación, base de datos, etc.

¿Por qué necesito entonces system center?

Tras haber leído los anteriores apartados debería quedar más o menos claro, pero vamos a hacer un pequeño repaso, ya que se puede describir un resumen con una serie de razones por las que desplegar Microsoft System Center en nuestra empresa puede ser beneficioso para la organización. A continuación describiremos algunas de ellas:

- Se puede extender el servicio del Service Desk más allá del alcance actual minimizando la inversión.
- Facilita la simplificación de la gestión del Service Desk y de otras áreas de IT.
- Ayuda a automatizar tareas repetitivas para liberar de esta carga de trabajo al equipo técnico.
- Proporciona un sistema de apertura automática de incidencias que puede adelantar la resolución de las mismas.
- Se puede automatizar la generación de un amplio espectro de informes, incluyendo los informes de cumplimiento de nivel de servicio que tanto le gustan a nuestros CEO.
- Se mantiene un repositorio de información útil para el tratamiento de futuras incidencias.
- Permite crear y gestionar la base de conocimiento, la KEDB y la CMDB.
- Ayuda a elaborar y mantener planes de contingencia a través de la herramienta de backup.
- Permite la provisión de programas y actualizaciones de manera centralizada.
- Ayuda a mantener los sistemas vigilados con una monitorización de extremo a extremo.
- Proporciona herramientas para la gestión de las nubes de la empresa.
- Permite al usuario realizar solicitudes para sus necesidades a través del portal de autoservicio. Según la configuración, las solicitudes se tramitan de manera automática.
- Proporciona al usuario un punto único de contacto con el servicio de informática, lo que mejora la experiencia de usuario.

Un punto adicional que tiene suficiente entidad como para sacarlo fuera de esa numeración y darle un poco más de notoriedad es que Microsoft System Center es una solución completa para la gestión de nuestra explotación. Lo que se puede llamar una solución de extremo a extremo. Esta solución cubre todos los ámbitos tal como se puede ver en la siguiente imagen:

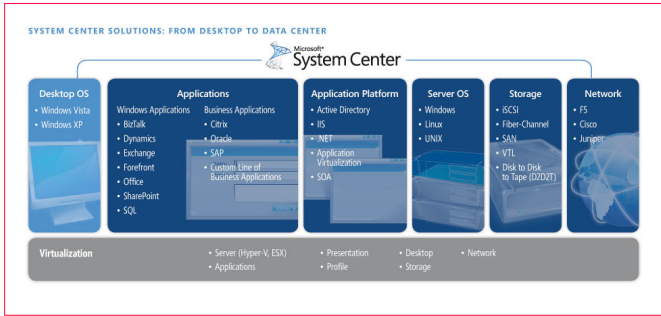


Imagen 4.- Solución de extremo a extremo.

Y el último punto a tener en cuenta es que toda la suite se integra con Microsoft Azure lo que nos brinda una escalabilidad sin límites.

Referencias

- [1] Página de la certificación “Microsoft Operations Framework Foundation”. <http://www.exin.com/ES/es/examen-es/&exam=microsoft-operations-framework-foundation>
- [2] Página del proyecto MOF 4.0 dentro de Microsoft. <http://technet.microsoft.com/en-us/solutionaccelerators/dd320379.aspx>
- [3] Página del producto Microsoft System Center. <http://www.microsoft.com/systemcenter/es/es/default.aspx> <https://www.microsoft.com/en-us/cloud-platform/system-center>

JUAN IGNACIO OLLER AZNAR

MVP Cloud and Datacenter Management

jioller@live.com

@jioller

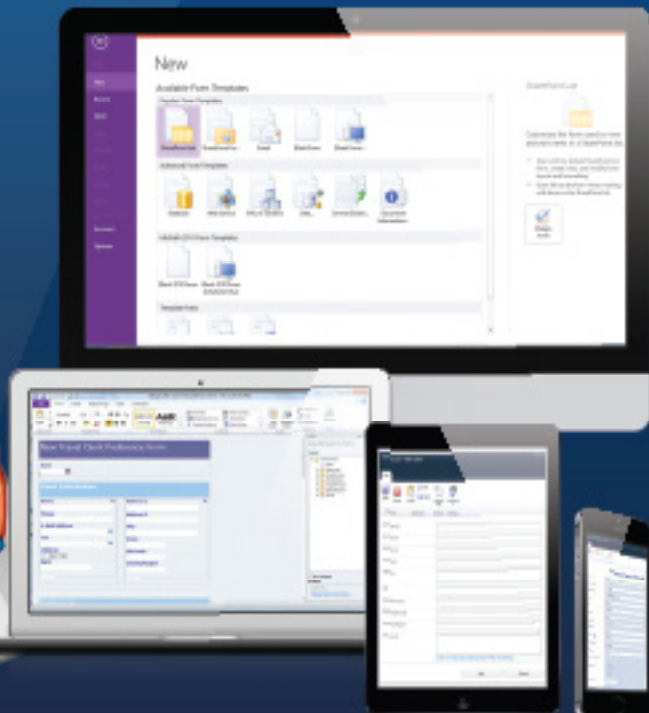
<http://blogs.itpro.es/jioller/>

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

KWizCom
KNOWLEDGE WORKER COMPONENTS

La MEJOR alternativa para InfoPath

**Ensaye los Formularios
de KWizCom Forms**



Destripando Modern Sites y Pages en SharePoint Online

Introducción a las “Modern Experiences”

Las “Modern experiences” es una nueva experiencia de usuario disponible actualmente a 3 niveles:

- 1.- Modern team sites.
- 2.- Modern list and library experiences.
- 3.- Modern site pages.

Esta nueva experiencia se caracteriza por ser mucho más amigable para el usuario, más sencilla para el autor, mucho más rápido, y lo que en principio es su principal característica: totalmente responsive, multi-dispositivo y multi-navegador.

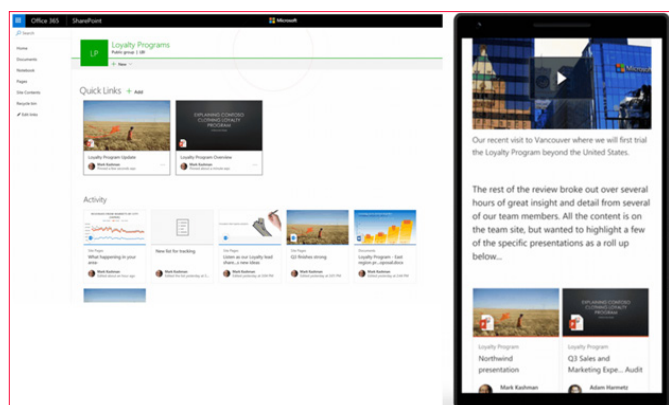


Imagen 1.- Sitio moderno de SharePoint Online.

Qué está soportado y qué no con los Modern Sites

Sinceramente, a día de hoy, los Modern Sites están bastante limitados en cuanto a posibilidad de personalizaciones. Poco sabemos sobre el roadmap, pero parece que Microsoft está trabajando en ofrecer alguna solución a las limitaciones actuales.

- Aplicar un Tema personalizado
- Aplicar un Tema Out of the Box
- Fields y Tipos de contenido
- Bibliotecas y listas
- Master Pages personalizadas
- Page Layouts personalizados
- User custom actions
- Embed Javascript custom
- Modern sub sites
- Características de Publishing
- Actualizar property bags
- Sandbox solutions

Imagen 2.- Posibilidades de personalización en sitios modernos.

Creando Modern Sites

A la hora de crear un Modern Sites tenemos varias opciones. Teniendo en cuenta que un Modern Site va (siempre) asociado a un grupo de Office 365, podemos crear un grupo de Office 365 desde la interfaz de Outlook web, tal y como vemos en la siguiente imagen:

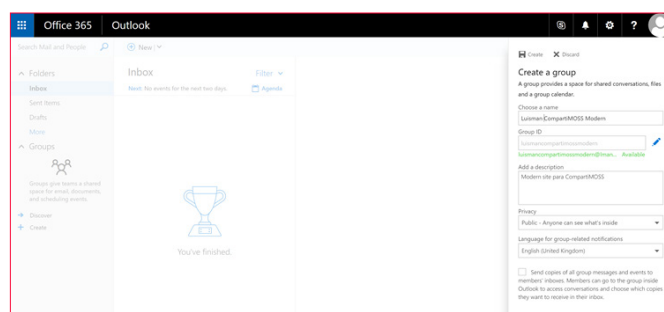


Imagen 3.- Creación de un Grupo de Office 365 desde OWA.

Una vez creado el grupo, podemos acceder al Modern site desde la opción “Site” del grupo.

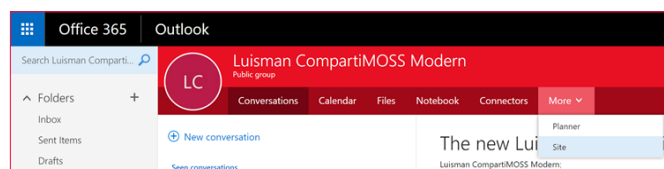


Imagen 4.- Acceso al sitio del Grupo creado.

También podemos crear un Modern site desde código, y aquí, de nuevo, el PnP nos ayuda con una serie de comandos PowerShell que nos lo ponen fácil:

```
### Crear Modern Site desde los comandos PowerShell de PnP ###
# Conectar a Azure AD
Connect-PnPMicrosoftGraph -Scopes "Group.ReadWrite.All", "User.Read.All"

# Imprimir en pantalla el AccessToken (no es requerido)
$accessToken = Get-PnPAccessToken

# Crear grupo de Office365
$group = New-PnPUnifiedGroup
-DisplayName "CompartiMOSSModernGroup"
-Description "Esto es un Modern site"
-Members "user@tenant.onmicrosoft.com"
-IsPrivate

# Conectar al nuevo Modern Site
Connect-PnPOnline -Url $group.SiteUrl
```

Creando Modern Pages

Una vez creado nuestro Modern Site, ya podemos crear Modern Pages desde la propia interfaz SharePoint y disfru-

tar de la nueva experiencia, mucho más rápida y sencilla.

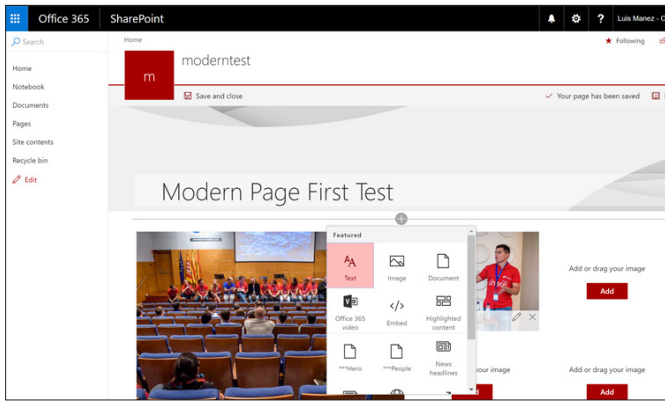


Imagen 5.- Ejemplo de Página moderna.

Las Modern pages, se almacenan en la biblioteca “Site pages”, como cualquier otra página de un Team Site. Sin embargo, las Modern pages tienen un tipo de contenido propio “Site page” y diferentes Fields.

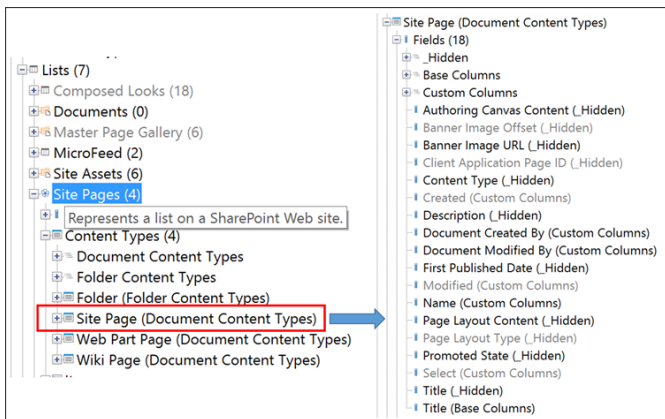


Imagen 6.- Análisis en detalle de una página moderna.

Además, una Modern Page tiene un “Page Layout Type” también diferente, en este caso “Article”.

“...la experiencia moderna se caracteriza por ser mucho más amigable para el usuario, más sencilla para el autor, mucho más rápido, y lo que en principio es su principal característica: totalmente responsive, multi-dispositivo y multi-navegador...”

Pero, ¿cómo almacena el contenido de una Modern Page?, pues toda la información de la página se guarda en un mismo Field, llamado "Authoring Canvas Content", y el contenido en este campo es puro HTML, donde cada Widget se

engloba en un tag DIV, con un atributo "data-sp-control-data".

```
<div>  
    <div data-sp-canvascontrol="" data-sp-controldata="%78%22controlType%22%5B%3,%  
        %22c4bd7b2f-7b6e-4599-8485-16504575f90%22,%22webPartData%22%5B;%78%22id%22%5  
        %22c4bd7b2f-7b6e-4599-8485-16504575f90%22,%22instanceId%22%5B;%221234f90e-dc4  
        %22"%Hero%22,%22description%22%5B;%22%5D"%Showing%22multiple%20pieces%20of%20con  
        %22serverProcessedContent%22%5B;%78%22htmlStrings%22%5B;%78%7D,%22imageSource  
        %22dataVersion%22%5B;%221.%0%22,%22properties">  
        <div data-sp-componentid="">c4bd7b2f-7b6e-4599-8485-16504575f90</div>  
        <div data-sp-htmlproperties=""></div>  
    </div>  
    <div data-sp-canvascontrol="" data-sp-controldata="%78%22controlType%22%5B%3,%  
        %22c70391ea-0b10-4ee9-b2b4-006d3fcad0c%22,%22webPartData%22%5B;%78%22id%22%5  
        %22c70391ea-0b10-4ee9-b2b4-006d3fcad0c%22,%22instanceId%22%5B;%22929ce0b321-96f  
        %22Quick%20links%22,%22description%22%5B;%22Create%20links%20to%20important%20C  
        %22serverProcessedContent%22%5B;%78%22htmlStrings%22%5B;%78%7D,%22imageSource  
        %22dataVersion%22%5B;%221.%0%22,%22properties%22%5B;%78%7D%7D,%22id%22%5B;%22  
        <div data-sp-componentid="">c70391ea-0b10-4ee9-b2b4-006d3fcad0c</div>  
        <div data-sp-htmlproperties=""></div>  
    </div>
```

Y es, en ese `"data-sp-controldata"` atributo, donde con cierto encoding, se almacena la información de cada widget, en formato JSON, por ejemplo, en la siguiente imagen, podemos ver un fragmento JSON del `"Hero"` widget:

```

"contentType": 3,
"displayMode": 1,
"webPartId": "c4bd7b2f-7b6e-4599-8485-16504575f590",
"webPartData": {
  "id": "c4bd7b2f-7b6e-4599-8485-16504575f590",
  "instanceId": "1234f90e-dc4e-4fdb-b56e-db216ae6088e",
  "title": "***Hero",
  "description": "***Showing multiple pieces of content with quick links.",
  "serverProcessedContent": {
    "htmlStrings": {},
    "imageSources": {},
    "links": {}
  },
  "dataVersion": "1.0",
  "properties": {
    "layout": 5,
    "content": [
      {
        "id": "3915dc75-6b75-468c-bde8-60abf3560978",
        "color": "red",
        "image": {
          "siteId": "b310ecb3-2f78-45b5-8487-7b169e9a6f39",
          "webId": "756b6690-e0f6-4aa1-b320-eca7e7e15964",
          "id": "73ce2d62-80b2-4629-bce1-69a046ea1e3c",
          "url": "/moderntest/SiteAssets/SitePages/Modern-Page-First-Test/314---SUG_CAT.jpg"
        },
        "link": "http://",
        "description": "",
        "title": "",
        "showDescription": false,
        "showTitle": false,
        "showImage": true,
        "alternateText": ""
      },
      {
        "id": "663fa0af-2329-41e1-9640-824f3b1212d4",
        "color": "red",

```

Resumen

Las Modern Experiences han llegado para quedarse. Ofrecen una experiencia de usuario mucho más agradable, rápida y multi-dispositivo, y aunque actualmente tiene muchas limitaciones a nivel de personalización, Microsoft sigue trabajando para ofrecer opciones y alternativas. Además, el PnP nos ofrece algunas utilidades para provisionar y trabajar con Modern sites.

LUIS MAÑEZ

SharePoint / Cloud Solutions Architect en ClearPeople LTD

@luismanez

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

SharePoint y Azure: Azure Content Delivery Network (CDN)

Que es y cómo funciona el Azure CDN

Azure Content Delivery Network proporciona los servidores para cachear información que no cambia frecuentemente. En este momento Azure CDN tiene servidores en 44 ciudades alrededor del mundo, incluyendo dos en Colombia (Medellín y Barranquilla), Buenos Aires, Quito, Rio de Janeiro, Sao Paulo y Madrid (la lista completa se puede encontrar en el sitio de Microsoft <https://docs.microsoft.com/en-us/azure/cdn/cdn-pop-locations>). Los servidores son de alto rendimiento y disponen de un ancho de banda sobredimensionado para poder soportar grandes cargas de trabajo.

CDN funciona de la misma manera que el cacheo local tradicionalmente utilizado por .NET Framework o por SharePoint internamente, es decir, mantiene en memoria archivos que se utilizan frecuentemente de tal forma que cuando un usuario requiere uno de ellos, lo puede acceder y enviar sin necesidad de utilizar IOs del disco duro o recuperarlos de las Bases de Datos. La ventaja de Azure CDN y sus servidores distribuidos globalmente es que la información se puede encontrar mucho más cerca del usuario final, disminuyendo el tiempo necesario para descargarla.

“Azure Content Delivery Network proporciona los servidores para cachear información que no cambia frecuentemente”

Azure CDN se utiliza también de la misma forma que el cacheo local tradicional: para mantener imágenes, hojas de estilo, documentos, archivos y/o scripts de cliente (JavaScript, por ejemplo). Adicionalmente, CDN ayuda a disminuir la carga de los sistemas, reduciendo la cantidad de información que se tiene que procesar (por ejemplo, en SharePoint, reduciendo el número de consultas a las Bases de Datos).

Azure CDN y SharePoint

Aunque SharePoint contiene nativamente diferentes mecanismos de cacheo, todos ocurren localmente en la granja misma, tanto para SharePoint Server On-Prem, como en SharePoint Online en Office 365. Esto ayuda a disminuir la

carga de las granjas, pero no disminuye el tiempo de respuesta de cara a los usuarios si la distancia a la granja es grande (granja de SharePoint y cliente en diferentes continentes, por ejemplo). Azure CDN ayuda a que los archivos que no cambian frecuentemente sean servidos mucho más rápido.

El siguiente ejemplo muestra cómo crear un Add-In Provider Hosted en Office 365 de SharePoint que cachea las hojas de estilo y archivos de JavaScript, aunque la utilización con SharePoint Server On-Prem es exactamente la misma.

Ejemplo de uso de Azure CDN con un AddIn de SharePoint Provider Hosted

- 1.- Crear la Aplicación Web en Azure para el AddIn:
 - Lóguese en el centro de Administración de Azure (<http://portal.azure.com>). Si no dispone de una cuenta, es posible crear una de prueba desde el mismo sitio.
 - Cree un Grupo de Recursos en Azure (llamada “CDNCompartiMOSS_RG” en el ejemplo) para contener tanto la Aplicación Web como el Perfil de CDN.
 - Cree una Web App en el Grupo de Recursos (llamada “CDNCompartiMOSSAddIn” en el ejemplo). En el sitio http://www.gavd.net/servers/sharepointv5/spsv5_item.aspx?top=art&itm=2008 se pueden encontrar instrucciones detalladas sobre cómo hacerlo.
- 2.- Cree el Add-In de SharePoint usando Visual Studio 2013 o 2015 (llamado “CDNAddIn” en el ejemplo). En el sitio http://www.gavd.net/servers/sharepointv5/spsv5_item.aspx?top=art&itm=2012 están las instrucciones sobre los pasos a seguir.
- 3.- Despliegue el Add-In en Azure. El sitio http://www.gavd.net/servers/sharepointv5/spsv5_item.aspx?top=art&itm=2015 indica cómo realizar la operación.
- 4.- Despliegue el Add-In en SharePoint Online. Las instrucciones sobre cómo hacerlo se pueden encontrar en el sitio http://www.gavd.net/servers/sharepointv5/spsv5_item.aspx?top=art&itm=2019. Desde este momento se dispone de un AddIn de SharePoint en Office 365 completamente funcional.
- 5.- Agregue un directorio “Styles” y una hoja de estilos en el proyecto Web del Add-In en Visual Studio. Agregue adicionalmente referencias en la pági-

na "Default.aspx" a uno de los scripts de JavaScript que ya existen y a la hoja de estilos acabada de crear. Publique de nuevo el Add-In en Azure siguiendo las mismas instrucciones dadas en el punto 4.

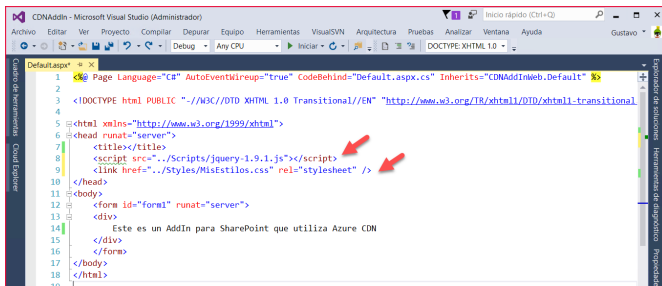


Imagen 1.- Código del Add-In en Visual Studio.

6.- Utilizando la administración de Azure, agregue un Perfil de CDN en el Grupo de Recursos (buscar por CDN en la ventana de "Add"). Asígnele un nombre ("CDNCompartiMOSS" en el ejemplo), seleccione la suscripción y el Grupo de Recursos (si no está seleccionado por defecto). Seleccione un nivel de precios. La creación puede durar hasta media hora. Un "Perfil" no es más que un contenedor para puntos de acceso (Endpoints) a CDN.

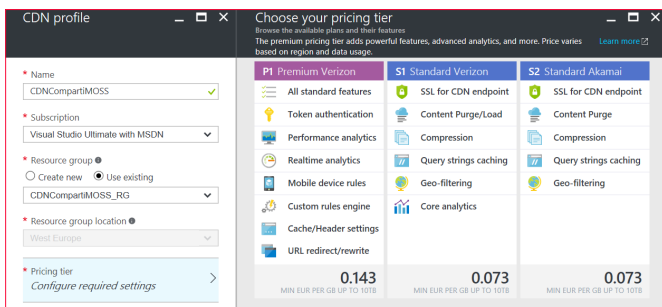


Imagen 2.- Creación del Perfil de CDN en Azure.

7.- Después de que el Perfil ha sido creado, añádele un "Endpoint" (use el botón de "+Endpoint" en el menú horizontal superior que aparece en la ventana de Azure cuando se selecciona el Perfil): defina el nombre ("CDNCompartiMOSEndpoint" en el ejemplo), "Web App" como Tipo de Origen, seleccione como "Origen hostname" el nombre de la Web App creada anteriormente (cdncompartimoss.azurewebsites.com). No modifique el "Origin path" ni el "Origin host header" y utilice los protocolos HTTP y HTTPS.

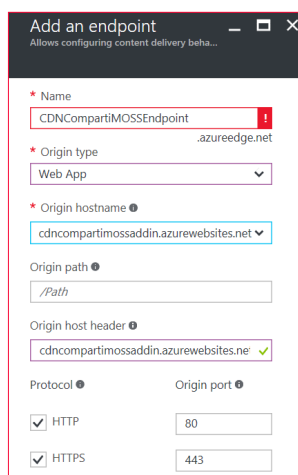


Imagen 3.- Creación del Endpoint de CDN en Azure.

8.- Cuando termina la creación del Endpoint, la conexión entre la Aplicación Web y el cache de CDN queda realizada por medio de la opción "Origin hostname" configurada al momento de creación. Si se hace ejecutar el Add-In de SharePoint, todos los archivos que utiliza son copiados a todos los servidores globales de Azure CDN automáticamente (la replicación global puede tomar unos 15 minutos). Si se selecciona el Endpoint en la pantalla de administración de Azure, bajo la sección de "Overview", en "Endpoint hostname" se puede ver el URL del CDN para el Add-In ("https://CDNCompartiMOSEndpoint.azureedge.net" en el ejemplo). Si utiliza el URL del directorio de los scripts junto con el URL del Endpoint en un navegador, puede comprobar que los archivos realmente han sido copiados, como muestra la imagen 4:

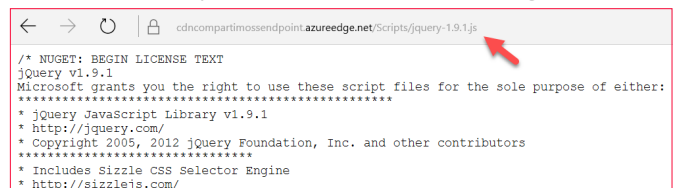


Imagen 4.- Archivo de JavaScript cacheado en el CDN de Azure.

9.- Cuando se traza de donde proceden los archivos al momento de ejecución del Add-In (con las herramientas de desarrollo de Edge, por ejemplo), se puede ver que todavía provienen de la granja de SharePoint, como muestra la Imagen 5:

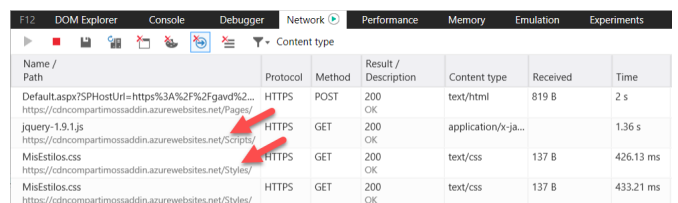


Imagen 5.- Procedencia de los archivos antes del cacheado en el CDN de Azure.

Para hacer que SharePoint utilice el CDN es necesario cambiar el URL de los archivos CSS y JS en default.aspx (en Visual Studio) y publicar la aplicación de nuevo. Utilice el URL base del Endpoint creado anteriormente:

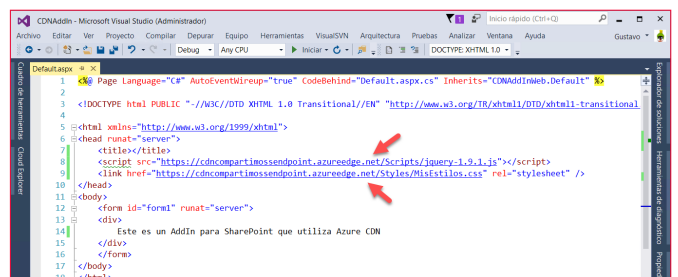


Imagen 6.- Cambio de URL para usar el CDN de Azure.

10.- Después de publicar el Add-In, se puede comprobar que los archivos provienen del cache de CDN:

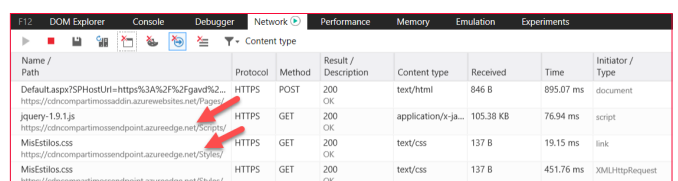


Imagen 7.- Procedencia de los archivos después del cacheado en el CDN de Azure.

Los tiempos de carga dependen totalmente de la localización del cliente con respecto a los centros de datos de Azure y Office 365. En el ejemplo mostrado, tanto el cliente como los centros de datos se encuentran en localidades muy cercanas, por lo que no se aprecia la diferencia. Si el cliente y el centro de datos de Office se encuentran en continentes diferentes, Azure CDN utilizará un cache cercano al cliente y las diferencias serán radicales.

Conclusión

Azure CDN (Content Delivery Network) es una opción de cacheo global que se puede utilizar para reemplazar el cacheo nativo de SharePoint. El cacheo puede aligerar la descarga de archivos que cambian muy poco en el tiempo, tales como hojas de estilo y scripts de JavaScript.

El precio de utilización de Azure CDN es bastante bajo y su configuración y utilización son muy sencillos. Fuera de las posibilidades técnicas mostradas en el ejemplo del artículo, es posible también configurar otros parámetros, tales

como el tiempo de cacheo y el cacheo por directorios específicos. Hay que tener siempre en cuenta que archivos en CDN son públicos y no es posible asegurarlos, por lo que los archivos cacheados no deben contener información sensible.

Una posibilidad muy interesante para explorar en el futuro, es la utilización de Azure CDN para contener las nuevas WebParts de Cliente de SharePoint basadas en el SharePoint Framework (SPFx), que está en el momento de publicación del artículo apenas en Release Candidate. Todas las WebParts de Cliente tienen que ser programadas en JavaScript, y una vez puestas en producción, los archivos que las componen no cambiarán rápidamente, por lo que forman un candidato ideal para Azure CDN.

GUSTAVO VELEZ

MVP Office Servers and Services

gustavo@gavd.net

<http://www.gavd.net>



May, 20th 2017

SharePoint Saturday Madrid

No te lo puedes perder el evento técnico de SharePoint del año.
Si quieres participar como ponente, patrocinador o asistente

<http://www.spsevents.org/city/madrid/madrid2017>

@sps_madrid

Gobernabilidad en Power BI

En el presente artículo describiremos varios enfoques en cuanto a definiciones en materia de Gobernabilidad BI. Hoy en día brindando servicios consultivos me ha sucedido que me pregunten acerca de cuál es el criterio de despliegue, ambientes, y decisiones que se deben tomar al momento de distribuir nuestros reportes y paneles.

“...cuál es el criterio de despliegue, ambientes, y decisiones que se deben tomar al momento de distribuir nuestros reportes y paneles...”

Algunos temas que debemos abordar como primer paso en esta instancia:

Desarrollando una Cultura de Analytics

1.- Midiendo el Éxito: Adopción de Usuarios.

Hoy en día tenemos variantes para la medición de éxito y uso. Por ejemplo, a través de registros de auditoría que nos proveen los Enterprise Gateway instalados para la actualización de Datasets. Power BI actúa además como una herramienta de modelado que permite hacer un trackeo de uso sobre reportes, paneles, y orígenes de datos. Por lo tanto, un incremento del índice de adopción puede ser tangible y evidenciar un camino crítico para generar valor al negocio.

2.- Sponsoreo Corporativo.

Es fundamental contar con un sponsor corporativo que entienda los beneficios de una cultura que se mueve en base a la información y que promueva una visión de aprendizaje en diferentes niveles de la organización. Dicho sponsor además puede ser fundamental al momento de definir un plan de adopción a nivel corporativo.

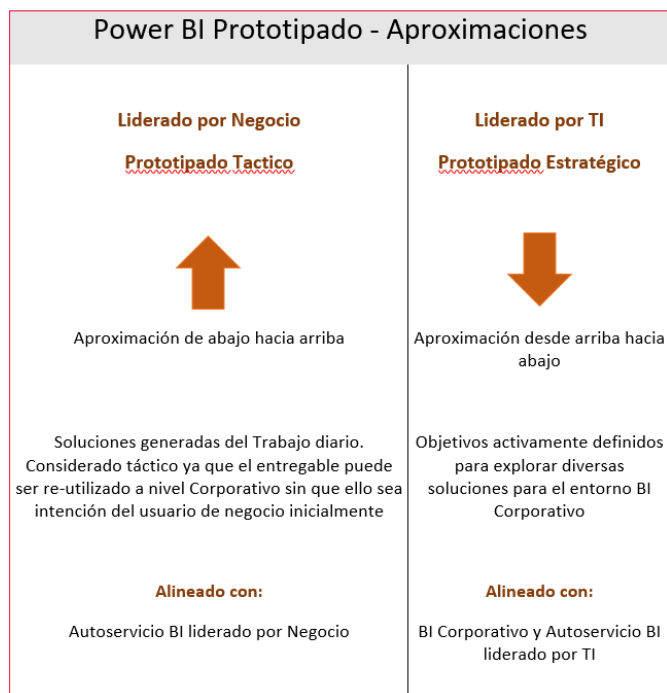
3.- Campeones de Power BI.

Más allá del sponsor corporativo, el rol de Campeón Power BI es fundamental al momento de ayudar a evangelizar la visión y la generación de reportes y paneles. Los mismos pueden ser definidos a partir de

cualquier rol de la compañía. Se supone busquemos personas con un alto nivel de especialización en el uso de Excel, y que deseen colaborar con el Sponsor Corporativo para plantear la mejor estrategia de implementación de Power BI en la organización.



Planificación de uso de Power BI como herramienta de Prototipado



Fases del Delivery – Liderado por el Negocio / Autoservicio BI



Fase 1 – Comenzar con un caso de negocio relativamente simple, pero con alto impacto en el negocio. Esto ofrece una gran oportunidad de aprendizaje con altas chances de éxito. Como resultado de esta fase el Campeón Power BI debe identificar escenarios de reportabilidad con baja complejidad, pero con gran impacto de negocio. Esos son los escenarios candidatos de prototipados en la próxima fase de Delivery

Fase 2 – Implementación de Reportes seleccionados en Fase 1 por parte del Campeón Power BI. En esta fase se puede incluir solo generación del reporte o también extracción de datos, estandarización, modelado y cálculos. Recibir feedback inmediato es fundamental en esta instancia por parte de colegas. Se debe aclarar que esto no va a formar parte de un proceso formal de Delivery de Reportes.

Fase 3 – Publicación y Monitoreo. Los miembros de este WorkSpace se recomiendan que sean usuarios con permisos solo de lectura. Una vez publicado el reporte existen determinadas actividades a realizar: creación de dashboard, establecer refresco de datos, verificación de seguridad y colaboración, creación de content pack, documentación en base a estándares. (se recomienda en esta instancia el uso de One Drive for Business o SharePoint Online para evitar pérdida de datos y mantener versionado).

Es bueno a medida que el reporte/dashboard crece en popularidad que el Campeón Power BI tenga presente la auditoría de uso.

Fase 4 – Soporte / Entrenamiento / Expansión. Expansión de visión: Compartir experiencias. Generar un sitio en la intranet o vía Newsletter. Gamification. Power BI Training.

Consolidación de infraestructura. Power BI reportes de producción. Generar un logo que ayude a identificar aquellos reportes que ya son productivos. Analizar expandir reporte utilizando SQL Server Services. Uso de Cortana Analytics.

Fases del Delivery – Liderado por TI / Autoservicio BI



Fase 1 – IT compila requerimientos, mediante entrevistas. Objetivo: Descubrir requerimientos de alto nivel. Guía para Prototipado (manejo de metodologías ágiles recomendado).

Fase 2 – Prototipado Estratégico: Etapa fundamental ya que es muy seguido que se da el desconocimiento de información necesaria por parte de usuarios de negocio. Armar un Prototipado un set de datos de ejemplo, mockups, herramientas simples a manejar en Power BI. Fundamental el manejo de Agenda y Objetivos en estas sesiones. Utilizar ventana Query en Power BI para mostrarle al usuario transformación y limpieza de datos.

Sesiones individuales para:

- Definir orígenes de datos, estructura, complejidad.
- Interfaz semántica (nombres de campos, jerarquías, agrupación de datos).

- Cálculos y reglas de Negocio.
- Estructura de Reportes.
- Interactividad necesaria en reportes (drill downs).

Fase 3 – Blueprinting. Definiciones:

- Orígenes de datos.
- Modelo de datos (completo o parcial). Incluye convenciones de nombres.
- Reglas de negocio, cálculos y jerarquías.
- Definición de manera adecuada de visualizar la información.
- Casos de Uso que pueden influir en temas de seguridad o decisiones de implementación.
- Captación de necesidades específicas como exportación, drill-down, interactividad, refrescos automáticos.

Fase 4 – Desarrollo. Definiciones:

- Trabajar con iteraciones.
- Realizar puestas en producción de pequeños componentes de la solución.

Fase 5 – Soporte / Entrenamiento / Expansión

- Es fundamental en esta etapa el monitoreo para me-

joras en temas de Performance, Seguridad y detectar elementos del entorno que son críticos.

- Elementos fundamentales en Training:
 - Diccionario de términos de información.
 - FAQ's.
 - Videos cortos de entrenamiento.

Conclusiones

Como conclusión, es importante aclarar que estos dos enfoques mencionados anteriormente pueden ser personalizados y deberían serlo para adaptarlo a la dinámica, visión y cultura general de la organización. Por otro lado, este concepto requiere de reuniones de Kick Off para definir no solo el roadmap que se desee trazar en gobernabilidad sino también para definir los roles involucrados y que también mencionamos al principio del presente artículo.

Este artículo pretende ser un punto de partida que ayude a modo de guía en la definición de Gobierno BI.

GASTÓN CRUZ

Business Intelligence Technical Manager | Arkano Software

Entrevista Luis Valencia

Mi nombre es Luis Valencia, soy Ingeniero de Sistemas y vivo en Medellín – Colombia.

Soy Microsoft Certified Professional desde el año 2003.

Me especializo en SharePoint, Office 365, Azure y Full Stack.



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

Cuando era niño iba a trabajar con mi papa, él era ingeniero electrónico, y en su oficina me acuerdo que compraron un Macintosh, era un computador de 14 pulgadas, y era a Color, creo que eso fue antes de los 90s, el computador tenía mouse y todo, en esa época eso era una maravilla. El computador era para los diseñadores que hacían planos en alguna herramienta, tal vez AutoCAD, no me acuerdo, y a mí me dejaban utilizarlo, luego mi papa compro un 386, con 40Mhz de velocidad, la RAM no me acuerdo cuanto era, en todo caso yo era feliz en esa época, aprendí dBASE, WordStar, Wordpress, QPRO.

Mi computador fue mejorando, pasé por 486, Pentium I, II, y mi primer trabajo fue también con mi papa haciendo Macros en QPRO y un editor de texto, creo que se llamaba WordPerfect. Prácticamente todo mi bachillerato trabajaba durante las vacaciones con mi papa en el computador en su oficina, luego decidí estudiar ingeniería de sistemas, simplemente porque me gustaban los computadores, en esa época no sabía nada de programación, y bueno entre a la ingeniería y resulta que me quedo gustando mucho más de lo que esperaba, y bueno aquí estoy 15 años de experiencia más tarde y con una excelente carrera que me ha dado todas las satisfacciones profesionales que cualquier persona puede desear.

Una vez termine la carrera mi primera experiencia fue programando en ASP Clásico con VBScript, ¡qué tiempos aquellos!, afortunadamente la tecnología fue mejorando y hoy en día las herramientas nos facilitan la vida.

¿Cuáles son tus principales actividades tecnológicas hoy en día?

Soy el Director de Tecnología de una empresa mediana en Colombia, se llama Software Estratégico. Tenemos aproximadamente 120 personas en Medellín, Bogotá y Nearshoring en Florida (Estados Unidos). Como Director de Tecnología manejo el área de Investigación y Desarrollo, visionando los proyectos, productos y servicios que podemos implementar. También tengo a mi cargo el área de arquitectura donde debo hacer el seguimiento técnico de todos los proyectos, revisiones par, mejores prácticas y obviamente acompañamiento a los desarrolladores para implementar mejores productos.

Mi vida profesional esta descrita en 4 tecnologías principales, SharePoint, Azure, IoT y FullStack, sé que es demasiado, y no me las se todas en Azure, pero ahí estoy todos los días aprendiendo y todos los días aportándole a la comunidad lo que más pueda.

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

Salir con mis amigos a tomar unas cervezas de vez en cuando, y disfrutar con la familia de algunos ratos libres.

¿Cuáles son tus hobbies?

Cuando puedo viajo, cuando no puedo, utilizo mi tiempo libre viendo series en Netflix, soy un adicto a esto.

¿Cuál es tu visión de futuro en la

tecnología de acá a los próximos años?

Definitivamente el Internet de las Cosas y BigData están estrechamente relacionados, veremos como el Internet de la Cosas produce Petabytes de información que nos ayude a tomar mejores decisiones con Machine Learning a hacer algoritmos y programas autómatas, que puedan aprender de nuestro entorno de nuestro día a día, y mejorar nuestra calidad de vida a través de dispositivos, herra-

mientas, Smart cars, etc. Muchas de las cosas que hoy hacen personas muy pronto serán automatizadas por robots, o herramientas, es un poco triste porque puede generar desempleo, pero desde el punto de vista técnico es algo realmente emocionante.

LUIS VALENCIA | OFFICE SERVERS & SERVICES MVP

le.valencia@outlook.com

@levalencia

<http://www.luisevalencia.com>

¿Conoces nuestras mini guías?



<http://www.compartimoss.com/guias>

Api Management y Logic Apps como Bus de Integración

Es cada vez más normal, que nos encontremos casos de integración entre múltiples sistemas que poco tienen que ver unos con otros, y tendamos a exponer servicios web o APIs como si no hubiera un mañana.

Esta situación lleva a tener que programar integraciones de punto a punto entre los distintos servicios, dificultando mucho el mantenimiento y la evolución de todos estos sistemas. Sin contar que por momentos el inventariado de todos estos servicios, viene a ser prácticamente imposible.

Si ya es bastante conocido el poder de “Integración” de Azure Logic Apps entre sistemas, que decir que, si lo sumamos a Azure Api Management, podemos tener un Bus de Integración muy potente gracias a las funcionalidades que veremos a continuación.

¿Por qué lo definimos como Bus de integración?

Me he permitido el lujo de referirme siempre al matrimonio de conveniencia entre Logic App y Api Management como una solución válida para implementar un “Bus de Integración”, huyendo un poco de decir bus de servicios. El motivo no es otro que Azure ya proporciona el servicio Service Bus (MaaS), que evidentemente está pensado para interconectar distintos sistemas, desacoplar sistemas y distribuir mensajes y transacciones haciendo uso de un complejo sistema de Colas.

La principal diferencia la podemos encontrar en que con Azure Logic App podemos integrar sistemas de una forma más sencilla sin recurrir tanto a desarrollos a medida, gracias al diseñador de procesos.

Si además envolvemos con Api Management nuestras APIs y nuestros procesos tenemos un sistema 100% fiable, seguro y sobretodo muy mantenible.

Con todo esto decimos que esta peculiar pareja de servicios, nos permiten construir un bus de integración muy potente.

Diseña procesos con Logic App

La idea de incluir Logic App, es el concepto integración con el menor esfuerzo posible. Para ello podemos crearnos

desde el portal de Azure una Logic App y definirlas desde el diseñador un pequeño proceso de integración para combinar por ejemplo un par de APIs que tengamos implementadas.

Como un ejemplo base podemos definir el siguiente flujo de trabajo:

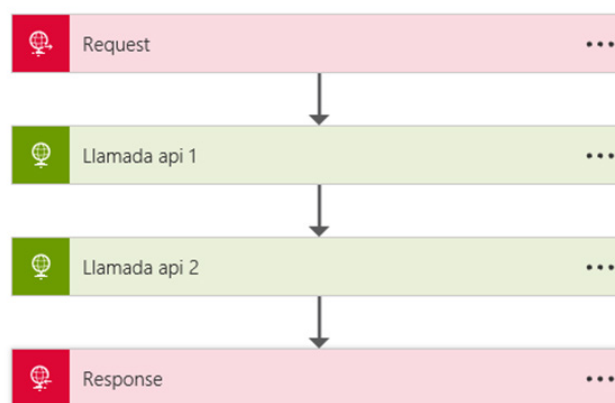


Imagen 1.- Diseñando procesos.

Si analizamos el proceso la primera acción es un request, que nos permite definir un punto de entrada a nuestro proceso, y exponer una URL a la cual podemos realizar la petición de arranque, en logic app esto se denomina “trigger o desencadenador”.

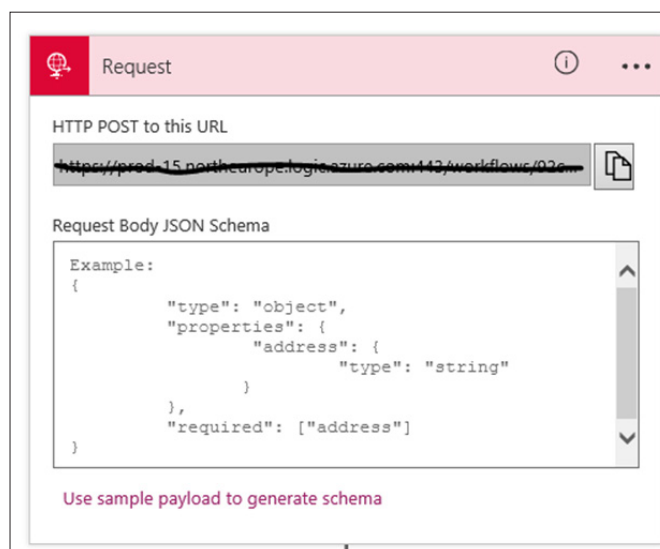


Imagen 2.- Desencadenador en Logic App.

Después debemos incluir las llamadas a nuestras APIs, que por ahora las insertaremos como una petición HTTP. Logic

App nos permite hacer una petición http del tipo que necesitamos, añadir las cabeceras necesarias (por ejemplo, si necesitamos algún parámetro de autenticación), y añadir el body necesario para realizar la petición. Además de que Logic App permite realizar la petición con múltiples protocolos de autenticación como podría ser OAUTH de Azure Active Directory.

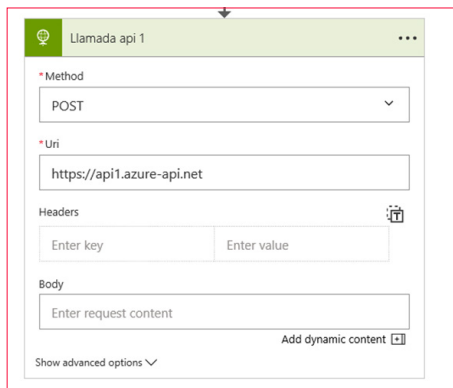


Imagen 3.- Peticiones http con Logic App.

El paso clave dentro del proceso, es unificar el resultado de ambas llamadas a las API, y para ello debemos hacer uso de la vista código que nos proporciona Logic App (por simplificar el proceso). Lo que necesitamos es editar la caja de salida o Response que hemos añadido como punto de salida de nuestro proceso.

Para ello necesitamos añadir al body, la estructura de salida JSON que esperamos devolver, y además mapear a los campos del JSON, los resultados de las llamadas a ambas APIs de los pasos anteriores, para que quede algo parecido a la imagen siguiente.



Imagen 4.- Unificar en el body varios resultados.

La forma de conseguir este mapeo, es cruzar desde la vista código la salida de API1 y API2 con los campos del JSON definido como salida. Para ello añadimos el siguiente código:



Imagen 5.- Vista código en acción.

La instrucción clave es "@body('Nombre caja consulta api')['Campo a mapear']".

Con este simple código conseguimos devolver de forma unificada, los resultados de la llamada a dos servicios, sin la necesidad de que estos tengan integración ninguna, simplemente necesitamos conocer el modelo del resultado que nos devuelven para poder hacer bien la construcción de la salida.

Controla tus APIs, ¿Por qué api Management?

Hasta ahora con el proceso de Logic App que hemos desarrollado hemos conseguido integrar dos APIs, pero podríamos si quisiéramos, integrar un CRM con un SharePoint, o un Directorio Activo con una tabla SQL. Pero ¿Es suficiente esto para definir Logic App como un bus de integración completo?

Respondiéndome a mí mismo, diré que no, y por ello vamos a añadir otro servicio PaaS como es Azure Api Management.

Api Management nos va a permitir publicar nuestros servicios y APIs de una forma directa con la idea de que estos servicios se puedan consumir por usuarios internos o externo, y además nos aporta:

- Securización y control de acceso a nuestros servicios y APIs.
- Permite escalar verticalmente, para adaptarse al tráfico de llamadas.
- Monitoriza las ejecuciones y los errores.
- Nos aporta un portal del desarrollador para testear nuestras APIs.

Registrar un servicio en API Management

Para empezar a trabajar con API Management deberemos crearnos en Azure un servicio de este tipo y accederemos a un amplio menú de funcionalidad que nos aporta el producto. El primer paso será empezar a registrar nuestros servicios, de modo que podamos empezar a consumirlos.

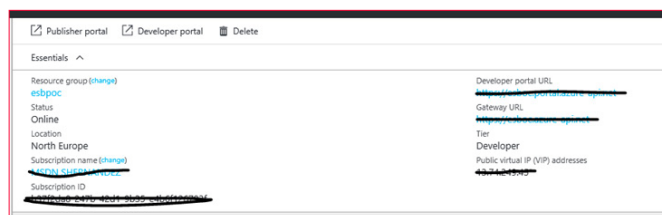


Imagen 6.- Portal de azure Api Management.

Desde el menú de configuración de Api Management en el portal de Azure ya podemos empezar a ver el concepto "Portal de Publicación", y "Portal de Desarrollador".

Desde el portal de publicación digamos que vamos a poder

administrar el servicio al completo, y el portal del desarrollador está pensado para los “usuarios con rol desarrollador” que no son otros que con los que hemos compartido un servicio o producto y desde este portal podrán aceptar esta invitación y empezar hacer uso del servicio compartido.

Si accedemos al portal de la publicación, deberemos acceder al apartado APIs, en el cual por defecto veremos creado un par de ejemplos como vemos en la imagen.

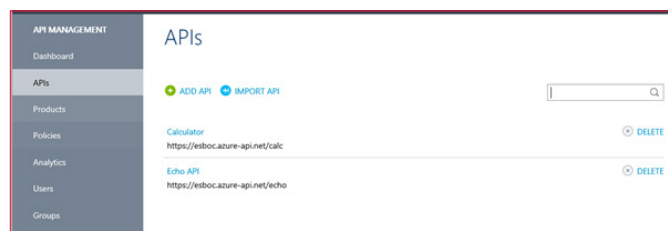


Imagen 7.- Control de Api en api management.

En este punto tenemos dos opciones, añadir una API o servicio de forma manual, o bien crearlo importando la definición de un servicio ya existente.

Para ello necesitaremos en el caso de ser un servicio web pues el contrato con el fichero WSDL o si por ejemplo es una API REST, necesitaremos el JSON de definición que nos proporciona Swagger.

Deberemos especificar la url para nuestro servicio, que acabará teniendo una forma parecida a “URL_API_MANAGEMENT/URL_NUEVA_API”

Decir por último que API Management trabaja con “Productos”, que podemos definir y asignarles un plan de uso (límite en el uso de llamadas), y a estos productos les asignaremos las APIS o servicios que nosotros elijamos.

Lo adelanto para que nos suene el concepto, ya que cuando demos acceso a un usuario a nuestros servicios, realmente le estamos compartiendo el producto al completo.

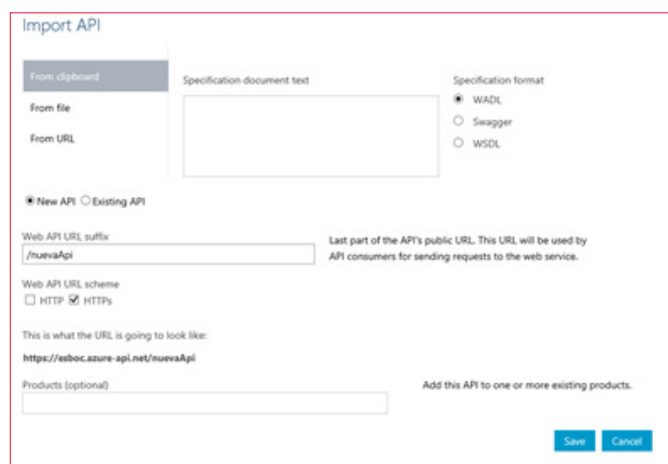


Imagen 8.- Importando una API.

Una vez creada registrada la API o servicio, ya podríamos empezar hacer uso desde la nueva URL que nos ha generado el producto.

Securiza tus servicios

Este punto es bastante interesante, ya que además de la posible autenticación que tenga cada uno de nuestros servicios, Api Management permite aplicar diferentes medidas de seguridad para acceder a los productos.

Para añadir seguridad a una API, deberemos acceder a la pestaña Seguridad del servicio y aquí podemos añadir la autenticación que necesitemos como una básica por HTTP o por protocolo Oauth 2.0 si es lo que necesitamos.

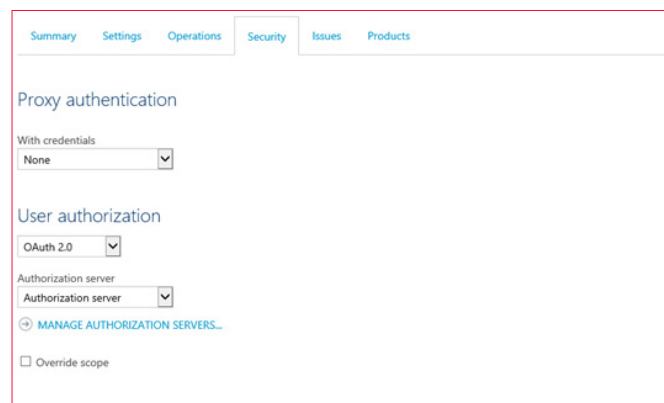


Imagen 9.- Securización en api management.

Si vamos a utilizar Oauth 2.0, necesitaremos añadir un servidor de autenticación válido previamente a configurarlo en la API, y para ello necesitaremos acceder a la opción Seguridad de todo el servicio de API Management.

Para ello deberemos proporcionar los datos de nuestro servicio de Azure AD, que podemos obtener desde el portal antiguo de Azure accediendo al servicio de Azure AD. (Proceso 100% documentado en los manuales de Azure).

Una vez configuremos nuestra API con autenticación, recordar que al hacer las llamadas pertinentes desde fuera o desde, por ejemplo, una Logic App, necesitaremos proporcionar las credenciales o el token de sesión en función del tipo de autenticación seleccionada. Recordar que hemos visto que Logic App soporta autenticación por Oauth, por lo que bastaría con configurar correctamente la llamada.

Facilita el acceso a los usuarios / portal del desarrollador

Un punto que para mi gusto distingue API Management de otras soluciones, es la facilidad para dar acceso a nuestros consumidores a una API. Para ello debemos saber que el proceso que usa para compartir un API, es el concepto suscripción, podremos invitar a un usuario a utilizar un servicio y este recibirá una invitación de suscripción al producto y si lo acepta podrá empezar a hacer uso de él.

Debemos recordar en este punto el concepto “Producto” que hemos nombrado anteriormente. Para ver el funcionamiento vamos a crear un nuevo producto, y lo vamos a sumar a los dos que viene por defecto “Starter” y “Unlimited”.

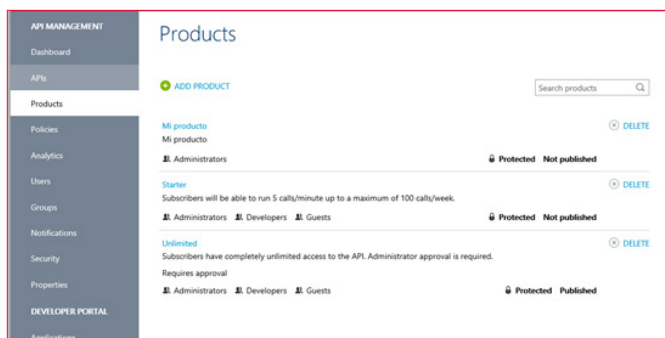


Imagen 10.- Productos.

Una vez creado podemos añadir tantas APIs como necesitemos, eso sí que estén previamente registradas.

El producto lo podemos hacer público, con suscripción, y con suscripción con aprobación de un responsable.

Una vez tenemos algún API registrada podemos publicar el producto, para que puede empezar a consumirse.

Por defecto en el servicio tendremos tres tipos de usuario Administradores, Desarrolladores e Invitados, pero podremos crear tantos tipos o grupos como necesitemos.

Los usuarios siempre pertenecerán a un grupo, y se podrán añadir usuarios desde diferentes fuentes:

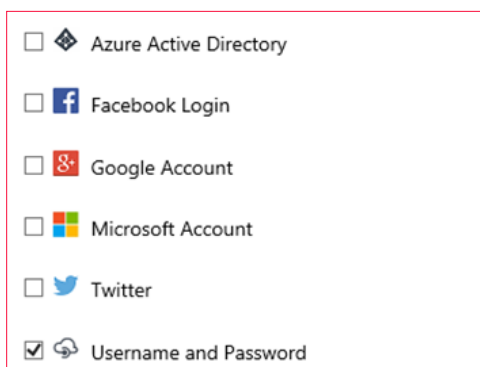


Imagen 11.- Proveedores de cuentas.

Una vez tenemos los usuarios dados de alta en nuestro servicio, y les hemos añadido al grupo que nos interesa, vamos a acceder al perfil del usuario para poder suscribirlo a un producto en concreto. Debemos asegurarnos de que el producto está visible para el grupo al que pertenece el usuario, es decir si queremos suscribir a Sergio Hernández y este está en el grupo desarrolladores, el producto que hemos creado debe tener visibilidad para este grupo (Product -> Visibility).

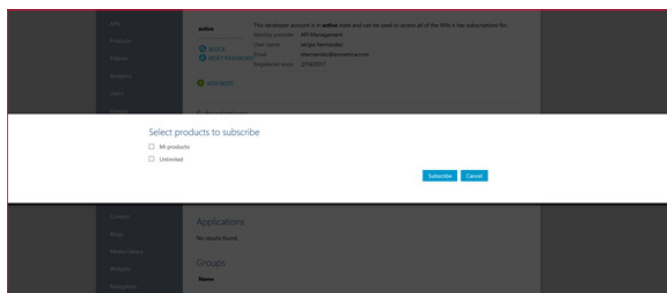


Imagen 12.- Visibilidad de un producto.

Una vez se ha suscrito al usuario en el producto, ya podría acceder al portal del desarrollador y comprobar que tiene acceso 100% al api y podría hacer uso del servicio.

Nosotros como proveedores le daremos acceso al portal del desarrollador, y el usuario con sus credenciales privadas, podrá acceder a los servicios y productos que le hemos compartido.

“Es una solución en la Nube de Microsoft que contiene una colección de servicios para administrar y proteger nuestra infraestructura OnPremise, en la nube o si tenemos una infraestructura híbrida”

Aun así, el usuario siempre recibirá a su cuenta de correo electrónico un mail con la confirmación de la suscripción al producto.

Nota importante el usuario deberá aceptar la suscripción para poder empezar a hacer uso de los servicios del producto.

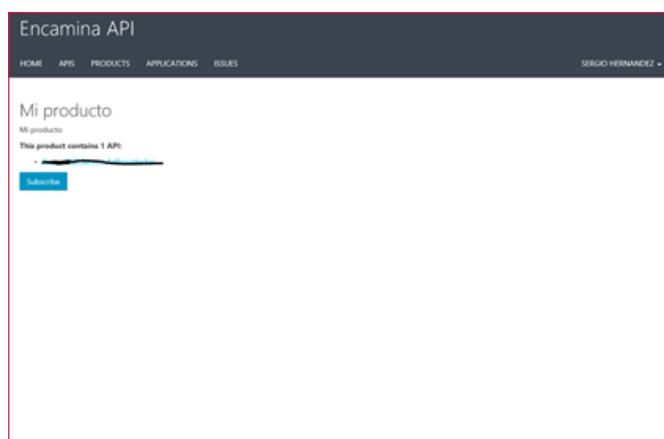


Imagen 13.- Aceptando la suscripción.

Por último y no menos importante, el usuario tendrá asignada una serie de claves propias, o claves de suscripción, que deberá incluir en las llamadas a los servicios para poder consumirlos. Esta seguridad se sumaría a las ya existentes en el API Management y en las propias API's de Origen.

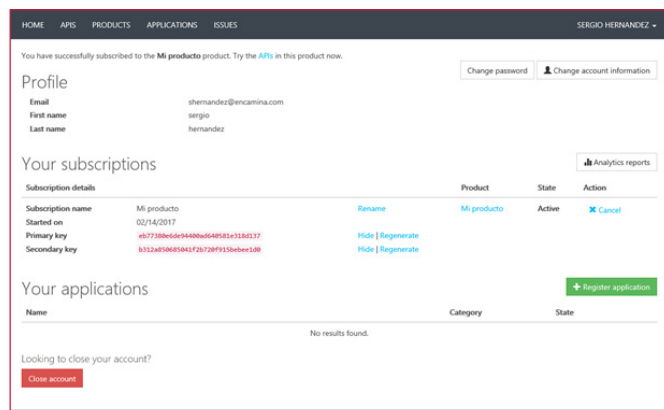


Imagen 14.- Claves de usuario por suscripción.

Esta clave se añadiría en las cabeceras de petición a los ser-

vicios como vemos en el siguiente ejemplo en c#.

```
var client = new HttpClient();
var queryString = HttpUtility.ParseQueryString(string.Empty);

// Request headers
client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key", "{subscription key}");
```

Monitorización y rendimiento

Con Api Management podremos monitorizar y controlar las ejecuciones y los errores, que se produzcan.

En la página principal del portal de publicación encontramos un dashboard general con toda la información en cuanto al uso, rendimientos, usuarios suscritos a los productos, etc.

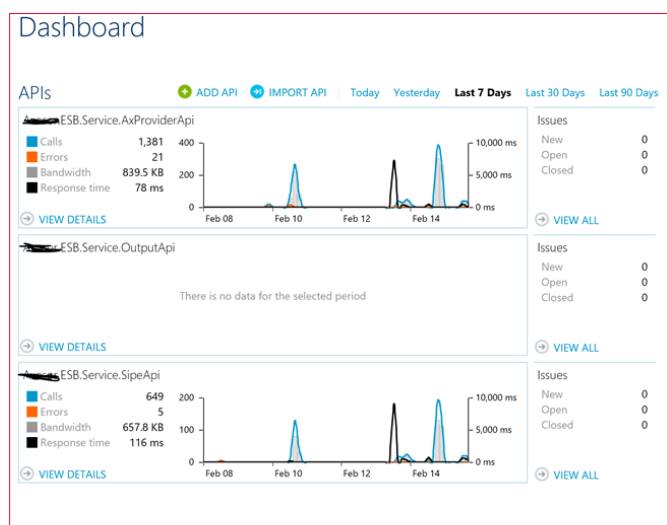
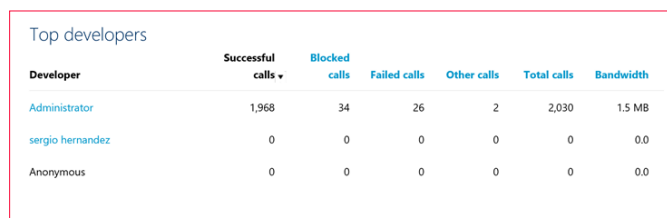


Imagen 15.- Monitor de rendimiento.

Así mismo podremos ir navegando por los distintos servicios y productos para focalizar la información de estos informes para un producto o servicio en concreto.

Podemos obtener datos muy interesantes como tiempos de respuesta, número de llamadas, número de llamadas acabadas en error..., que nos pueden entre otras cosas permitir ver el consumo de nuestros servicios, y poder plantear tendencias de uso y en función de esto estudiar políticas de escalado.

Además, tenemos una utilidad de análisis en el propio portal que amplía aún más toda esta información, bajando incluso al nivel de usuario o de suscripciones.



Developer	Successful calls	Blocked calls	Failed calls	Other calls	Total calls	Bandwidth
Administrator	1,968	34	26	2	2,030	1.5 MB
sergio hernandez	0	0	0	0	0	0.0
Anonymous	0	0	0	0	0	0.0

Imagen 16.- Estadísticas por usuario.

En la pestaña de uso podremos incluso ver desde donde se realizan las peticiones a los productos. Por ejemplo, en mi grafica vemos que he desplegado servicios en varias regiones distintas.

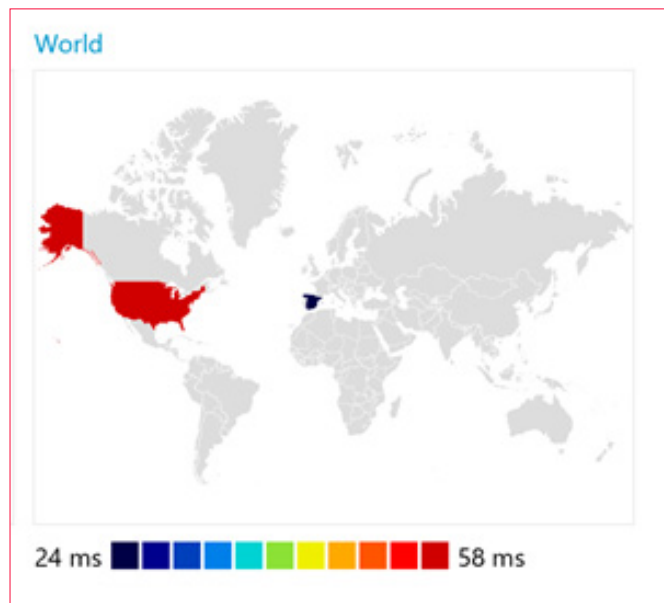


Imagen 17.- Localización de las llamadas

En resumen, podemos monitorizar, y controlar la salud de nuestros servicios y productos, así como controlar picos de actividad.

Volviendo al diseñador de procesos de Logic App

Por concluir, decir que una vez que se han registrado los servicios como API en API Management, podemos beneficiarnos de ello también en el diseño de los procesos en Logic App

Como tenemos definidas nuestras API's, incluyendo métodos, parámetros, protocolos de autenticación, cabeceras, cuerpos de los mensajes..., con esta información Logic App nos expone una serie de acciones que nos facilitan un poco más si cabe las consultas.

En vez de realizar peticiones HTTP podemos tender a usar la acción de API Management, y directamente llamar a los servicios expuestos por el producto. Que decir que haciendo así las peticiones nos aseguramos de que sea la propia API Management la que gestione las peticiones, monitoree y exponga los criterios de seguridad en cuanto a suscripciones (claves).

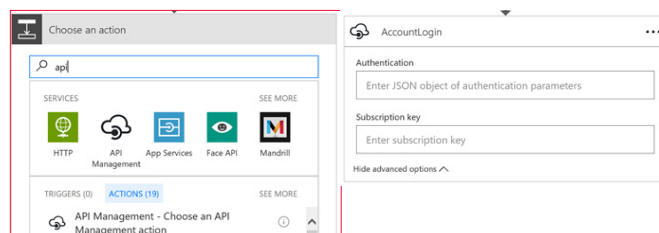


Imagen 18.- Acciones en Logic APP para Api management

Conclusiones

Una vez más los servicios PaaS de Azure, nos demuestran mucho con muy poco, es decir parece que por separado

son procesos muy completos pero que no terminan de representar un producto, en cambio combinándolos volvemos a la misma conclusión de siempre. Podemos construir sistemas muy complejos y encima beneficiarnos de todas las ventajas de este tipos de servicios (pago por uso, SLA garantizado del 99% por Microsoft, reducción de costes en implementación...).

En este caso hemos dado con un bus de integración completo, escalable, que podemos hacer crecer de forma casi infinita, porque ¿No es cierto que si añadimos Azure Function podemos integrar con casi cualquier cosa y sin necesidad de tener el coste de gobernar una infraestructura

compleja?

Como dice el refrán, “La unión hace la fuerza”, y llevado a Azure no es poner servicios por ponerlos, pero haciendo una combinación sensata de funcionalidad conseguimos sistemas antes impensables sobre todo si nos referimos a costes y tiempos de desarrollo.

SERGIO HERNANDEZ MANCEBO

Principal Team Leader at ENCAMINA

@shmancebo



NO TE PUEDES PERDER EL MAYOR EVENTO
DE COMUNIDAD SOBRE
MICROSOFT AZURE

2017
Global Azure
BOOTCAMP

Madrid
Sábado 22 de abril de 2017

2016
www.azurebootcamp.es
BOOTCAMP

@gwab_es

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.



Temperatura en tiempo real, aplicación web con Raspberry, Azure IoT Hub y SignalR

El IoT (Internet de las cosas) está explotando y todo el mundo está hablando de ello. Las fuentes de Twitter están zumbando, y los mensajes de blog están inundados de información. Sin embargo, para muchos de nosotros simples mortales, parece algo de otro mundo.

En este post, voy a tratar de hacer que sea fácil para que usted pueda entender los conceptos básicos y la tecnología detrás de las escenas. También crearé una aplicación web de muestra totalmente funcional, donde podrá ver el resultado final y cómo se conectan todas las piezas.

Entonces, ¿Qué es el IoT? Probablemente encontrará diferentes explicaciones, pero para mí, la mejor explicación es:

El IoT se compone de cosas o dispositivos que están conectados a una red; Producen o reciben datos, y luego realizan acciones basadas en él.

Tomemos un sensor de temperatura como un simple ejemplo. En este escenario, este sensor en particular envía datos a un sitio web en tiempo real. Está conectado a un refrigerador en una farmacia, que por ley necesita tener temperatura controlada. En lugar de emplear a una persona para ir a la nevera cada 30 minutos para tomar nota de la temperatura actual, anotarla manualmente en un pedazo de papel, y luego guardar la evidencia de que la temperatura se ha escrito por 24 horas, el sensor lo toma automáticamente.

El IOT permite nuevos escenarios empresariales, nuevos productos, nuevos servicios y -como el que voy a mostrar- hay cientos que ya están en producción. Y no es algo de otro mundo.

¿Qué necesitará para completar esto?

Lo primero que debes tener es la cosa, o dispositivo. En mi caso tengo una Raspberry PI 3. Se trata de una computadora de \$ 39.99 dólares a la que podemos conectar dispositivos y luego crear aplicaciones que interactúan con esos dispositivos o sensores. Aquí también tengo un sensor BMP180, que se puede encontrar en Amazon por \$ 9 o \$ 10. Este sensor mide la temperatura en tiempo real. Más adelante aprenderá cómo mostrarlo en tiempo real en una aplicación web.

La Raspberry PI 3 viene por defecto con Raspbian instalado. Se trata de un sistema operativo basado en Linux y Debian, sin embargo, se puede instalar Windows IoT Core. Se trata de una versión de huella mínima del sistema operativo. Lo asombroso de este sistema operativo es que puedes crear aplicaciones con las herramientas que ya conoces y amas. Sí, Universal Windows Plataforma Apps se ejecutará en IoT Core, así como Windows 8 o Windows 10, con el mismo lenguaje de programación que ya conoces, C#.

Hay APIs por ahí para crear aplicaciones en JavaScript o en otros idiomas, pero por simplicidad sólo mostraré el código C#.

Partes móviles de nuestra solución

La PI APP

En nuestra solución, en Visual Studio, tengo un proyecto en Universal Windows App. Esta aplicación tiene sólo cuatro trabajos: se conectará al sensor, leerá la temperatura del sensor, lo mostrará en la pantalla y luego la enviará al concentrador de IoT.

La aplicación en este dispositivo nunca se conectará en tiempo real con una base de datos. Estos sensores pueden leer cientos de lecturas por segundo, por lo que dependiendo de cómo codifique o configure su aplicación, si la inserta directamente en una base de datos, la aplicación probablemente no escalará bien.

IoT Hub

El IoT Hub es un sistema de mensajería construido sobre Azure Event Hub. Es básicamente un sistema que es capaz de recibir millones de mensajes por segundo, que luego podemos procesar en un momento posterior. Por lo tanto, si tenemos 10.000 sensores de temperatura, el envío de temperatura cada segundo, podría recibir 10.000 mensajes por segundo, y eso no es mucho para este servicio Azure. He copiado y pegado las capacidades de este [artículo de Microsoft Azure](#). Azure IoT Hub:

- Proporciona múltiples opciones de comunicación de dispositivo a la nube y de la nube a dispositivo, incluyendo mensajería unidireccional, transferencia de archivos y métodos de solicitud y respuesta.
- Proporciona un almacenamiento que se puede consultar para los metadatos del dispositivo y la información

de estado sincronizada.

- Permite comunicaciones seguras y control de acceso utilizando claves de seguridad por dispositivo o certificados X.509.
- Proporciona un amplio monitoreo de la conectividad de dispositivos y eventos de administración de identidad de dispositivos.
- Incluye bibliotecas de dispositivos para los idiomas y plataformas más populares.

Azure Web Job

Bien, ahora tenemos nuestra aplicación IoT Core enviando mensajes al concentrador de IoT en tiempo real, ¿qué hacemos a continuación? Debemos procesarlos sin conexión a través de un proceso diferente -en este caso, un trabajo de Web de Azure.

Un trabajo de Azure Web es sólo un proceso de fondo que puede escribirse en C#, Powershell y en diferentes idiomas. Estos viven junto a una aplicación web y puede desarrollar cualquier lógica de negocio que desee, con el fin de procesar las cosas en el fondo.

Mi trabajo particular de Web de Azure se conecta al IoT Hub y toma los últimos mensajes en el IoT, luego inserta un nuevo registro en SQL Server (también en Azure) y marca los mensajes o el lote de mensajes procesados.

SQL Server

Como con casi cualquier otra aplicación, necesitamos una manera de almacenar nuestra información, porque el dispositivo pertenece a una empresa y a un usuario. Aunque hemos utilizado SQL Server para una base de datos relacional, esto no es obligatorio. La capa puede ser cambiada por alguna base de datos relacional si lo desea, pero la capa de la aplicación web cambiará en este caso, por supuesto.

Web App

La aplicación web es sólo una aplicación ASP.NET MVC, la que ya conoces y amas. Sin embargo, enchufamos SignalR. Como muchos de ustedes ya saben, SignalR es un marco de comunicación en tiempo real que le permitirá presionar los cambios que ocurren en el lado del servidor a la parte delantera sin respaldo de correos, o los tiempos de espera establecidos por JavaScript.

SignalR se basa en diferentes tecnologías construidas en los navegadores modernos para permitir esta comunicación en tiempo real. En nuestro caso, tenemos un panel que muestra la temperatura en tiempo real, el retraso es sólo lo que se necesita entre la aplicación PI para enviar el mensaje, el trabajo web para procesar el mensaje e inserta en una base de datos y luego lee el dato de la tabla correspondiente.

Cuando lo probamos, el retraso no fue más de 1-2 segundos, lo cual está bien para nuestro escenario. Un refrigerador no cambiará la temperatura drásticamente en ese

período de tiempo.

Creación del IoT Hub

Para que el código siguiente funcione, necesitamos crear un IoT Hub en el Portal Azure. No hay ciencia en este paso. Sólo tienes que hacer clic en el signo más, encontrar IoT Hub... y eso es todo.

Copia y pega la URL (debe ser algo como esto: myiothub.azuredevices.net) - lo necesitará más tarde.

Encontrará instrucciones sobre esto en la primera parte de este post:

<https://azure.microsoft.com/en-us/documentation/articles/iot-hub-csharp-csharp-getstarted/>

Raspberry PI App (Plataforma Universal Windows)

Voy a asumir que usted ya ha instalado la última versión de Windows IoT Core en su Raspberry PI. Si no es aquí un enlace que explica cómo hacerlo:

<https://developer.microsoft.com/enus/windows/iot/Docs/Get-Started/rpi3/sdcard/insider/GetStartedStep1.htm>

Una vez instalado Windows IoT Core, puede ejecutar Visual Studio y crear una aplicación UWP con C#, luego revise el código. Hay varias piezas de código relevante. Al final de la publicación encontrarás un enlace a mi repositorio de GitHub donde se publica el código completo, pero solo en este post solo muestro las piezas relevantes.

Cuando se inicia la aplicación, hay que inicializar el sensor. El sensor es BMP180 y lee la temperatura. Debe crear un temporizador con este código. El temporizador tiene un intervalo asignado, y una devolución de llamada, lo que significa que un determinado método se ejecutará cada número de milisegundos, de acuerdo con lo que el usuario ha configurado.

Puede encontrar la clase de sensor Bmp180 en este proyecto:

<https://www.hackster.io/acegrc/bmp180-windows-iot-d14b76>

```
private async void InitializeSensors()
{
    string calibrationData;
    try
    {
        _bmp180 = new Bmp180Sensor();
        await _bmp180.InitializeAsync();
        calibrationData = _bmp180.CalibrationData.ToString(); //
    Retorna una cadena que representa al objeto actual.
    if (_periodicTimer == null)
    {
        _periodicTimer = new Timer(this.TimerCallback, null, 0,
        readingInterval);
    }
    }
    catch (Exception ex)
    {
        calibrationData = "Error de dispositivo! " + ex.Message;
    }
}
```

Una vez inicializado el dispositivo, cree el código para la

devolución de llamada del temporizador. Este código sólo lee los datos del sensor y, a continuación, crea un mensaje JSON que se enviará al IoT Hub.

```
public async void TimerCallback(object state)
{
    string temperatureText;
    try
    {
        var sensorData = await _bmp180.GetSensorDataAsync(
            Bmp180AccuracyMode.UltraHighResolution);
        temperatureText = sensorData.Temperature.ToString("");
        temperatureText += "°C";

        var temperatureDataPoint = new
        {
            deviceKey = deviceKey,
            deviceName = deviceName,
            temperatura = sensorData.Temperature,
            fecha = DateTime.Now
        };
        //// actualizaciones de la interfaz de usuario... deben ser
        invocados en el subproceso de interfaz de usuario
        var task = this.Dispatcher.RunAsync(CoreDispatcherPri-
            ority.Normal, 0 =>
        {
            temperatura.Text = temperatureText;
        });
        var messageString = JsonConvert.SerializeObject(tem-
            peratureDataPoint);
        var message = new Microsoft.Azure.Devices.Client.Mes-
            sage(Encoding.ASCII.GetBytes(messageString));
        message.Properties["Ambiente"] = ambiente;
        await deviceClient.SendEventAsync(message);
        autoResetEvent.Set();
    }
    catch (Exception ex)
    {
        var a = ex.StackTrace;
        temperatureText = "Sensor Error: " + ex.Message;
    }
}
```

El cliente del dispositivo tiene que ser inicializado con la cadena de conexión del IoT Hub y la clave del dispositivo, como esto.

```
deviceClient = DeviceClient.Create(iotHubUri, new DeviceAu-
    thenticationWithRegistrySymmetricKey(deviceName, device-
    Key));
```

Azure Web Job

Después de que nuestros sensores estén leyendo datos y enviando mensajes al IoT Hub, necesitamos procesar esos mensajes e insertarlos en la base de datos. Nuestro trabajo web es solo una aplicación de consola C# que lee continuamente desde el Hub e inserta en la base de datos.

Nuestro trabajo web de Azure utiliza EventProcessorHost. Esta es una clase proporcionada por el Azure SDK que permite controlar la recepción de los mensajes y también la creación de punto de control.

¿Qué es un puesto de control? Es sólo una forma de controlar o tomar una instantánea de cuándo se procesan los mensajes, de modo que cuando la próxima vez que se ejecute el trabajo, no vuelva a leer los mismos mensajes.

Para que funcione, necesitamos una cuenta de almacenamiento de Azure. Luego en el archivo de configuración para

la aplicación de consola, tenemos que introducir la cadena de conexión del IoT Hub. Tenga en cuenta que debe extraerlo de la pestaña Mensajería en el portal azure: es lo que ellos llaman el punto final compatible con Event Hub.

Mi app.config se parece a esto:

```
<appSettings>
  <add key="EventHubConnectionString"
    value="Endpoint=sb://xuzthub-ns-tempsense-69811-
    eb157a3ed0.servicebus.windows.net/;SharedA-
    ccessKeyName=iothubowner;SharedAccessKey=mykey" />
  <add key="EventHubName" value="myeventhubname" />
  <add key="StorageAccountName" value="mystorageac-
    countname" />
  <add key="StorageAccountKey" value="mystorageaccount-
    key/" />
  <add key="TamanoLoteMensajes" value="100" />
</appSettings>
<connectionStrings>
  <add name="TempsenseConnection"
    connectionString="Server=192.168.100.3\SQL2014;Initial Cata-
    log=myDB;Persist Security
    Info=False;User ID=myusername;Password=mypassword;"
    providerName="System.Data.SqlClient"
  />
```

En Azure, los trabajos en la web tienen disparadores. Estos disparadores significan que ciertas funciones se ejecutan cuando sucede algo, como cuando un archivo se deja caer en el almacenamiento Azure o se recibe un mensaje en una cola. En mi caso, he utilizado un cronjob disparador, lo que significa que la función se ejecute cada minuto o, sin embargo, a menudo.

```
public static void CronJob([TimerTrigger("*/1 * * * *")] TimerInfo
    timer)
{
    try
    {
        Console.WriteLine(String.Format("Inicio lectura men-
            sajes : {0}", DateTime.Now.ToString()));
        string storageConnectionString = string.Format("De-
            faultEndpointsProtocol=https;AccountName={0};Account-
            Key={1}",
            Configuracion.StorageAccountName, Configur-
            acion.
            StorageAccountKey);
        string _guid = Guid.NewGuid().ToString();
        string eventProcessorHostName = _guid;
        EventProcessorHost eventProcessorHost = new Event-
            ProcessorHost(
                eventProcessorHostName,
                Configuracion.EventHubName,
                EventHubConsumerGroup.Default-
                GroupName,
                Configuracion.EventHubConnec-
                tionString,
                storageConnectionString);
        Console.WriteLine("Registering EventProcessor...");
        var options = new EventProcessorOptions();
        options.ExceptionReceived += (sender, e) => { Console.
            WriteLine(e.Exception); };
        eventProcessorHost.RegisterEventProcessorAsync(op-
            tions).Wait();
        eventProcessorHost.UnregisterEventProcessorAsync().Wait();
        Console.WriteLine(String.Format("Fin lectura mensajes :
            {0}", DateTime.Now.ToString()));
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

La clase de procesador de eventos, que es realmente responsable de abrir y cerrar conexiones, también es respon-

sable de recibir los mensajes del IoT Hub.

```
namespace ActiveSense.Tempsense.Receptor
{
    public class SimpleEventProcessor : IEventProcessor
    {
        Stopwatch checkpointStopWatch;
        int messageCount = 0;
        async Task IEventProcessor.CloseAsync(PartitionContext context, CloseReason reason)
        {
            Console.WriteLine("Processor Shutting Down. Partition {0}, Reason:{1}.", context.Lease.PartitionId, reason);
            if (reason == CloseReason.Shutdown)
            {
                await context.CheckpointAsync();
            }
        }
        Task IEventProcessor.OpenAsync(PartitionContext context)
        {
            Console.WriteLine("SimpleEventProcessor initialized. Partition:{0},offset:{1}", context.Lease.PartitionId, context.Lease.Offset);
            this.checkpointStopWatch = new Stopwatch();
            this.checkpointStopWatch.Start();
            return Task.FromResult(null);
        }
        public Task ProcessEventsAsync(PartitionContext context, IEnumerable messages)
        {
            List medidas = new List();
            foreach (EventData eventData in messages)
            {
                string strConn = string.Format(ConfigurationManager.ConnectionStrings["myConnectionString"].ConnectionString, eventData.Properties["Ambiente"]);
                messageCount++;
                string data = Encoding.UTF8.GetString(eventData.GetBytes());
                JObject o = JObject.Parse(data);
                var deviceKey = int.Parse(o["deviceKey"].ToString());
                using (ActiveSenseContext db = new ActiveSenseContext(strConn))
                {
                    try
                    {
                        var disp = db.Dispositivos.Where(p => p.DispositivoID == deviceKey);
                        if (disp.ToList().Count > 0)
                        {
                            ActiveSense.mynm.modelo.Modelo.Medida medida = new ActiveSense.mynm.modelo.Modelo.Medida()
                            {
                                DispositivoID = disp.FirstOrDefault().DispositivoID,
                                Valor = decimal.Parse(o["valor"].ToString()),
                                FechaHora = Convert.ToDateTime(o["fecha"].ToString());
                            };
                            Console.WriteLine(string.Format("Message received. Partition:{0}, Data:{1}(2)", context.Lease.PartitionId, data, eventData.EnqueueTimeUtc));
                            db.Medidas.Add(medida);
                            db.SaveChanges();
                        }
                        else
                        {
                            Console.WriteLine(string.Format("Device Key not found in database:{0}, Message:{1}", o["deviceKey"].ToString(), o));
                        }
                    }
                    catch (Exception ex)
                    {
                        Console.WriteLine(string.Format("Mensaje tuvo error:{0},{1}", ex.Message, data));
                    }
                }
            }
            //if (messageCount > Configuracion.TamanoLoteMensajes)
            context.CheckpointAsync();
            return Task.FromResult(null);
        }
    }
}
```

Bueno, ahora tiene nuestros datos en la base de datos, nuestro sensor está enviando mensajes al IoT Hub, y nuestro webjob de Azure los está leyendo en tiempo real e insertándolos fila por fila en nuestra tabla de medidas.

Aplicación web

La aplicación web es una simple aplicación ASP.NET MVC que muestra datos en tiempo real. La parte agradable de toda la demostración aquí es la lectura en tiempo real, para la cual utilizamos SignalR.

SignalR es un framework que nos proporciona las utilidades que necesitamos para actualizar nuestra página en tiempo real cuando algo sucede en el servidor. En nuestro caso tenemos un temporizador que, basado en un intervalo predefinido, va a la base de datos, toma la última temperatura del dispositivo, y luego lo envía a los clientes (los navegadores), y luego con JavaScript refrescará al usuario Interfaz en tiempo real. La mejor manera de demostrar que esto funciona es sólo para poner un dedo en el sensor, y luego verá la temperatura disparar.

Para fines de demostración todas las temperaturas son en grados Celsius.

El código que implementa este panel está usando SignalR. Para esto tenemos que crear un Hub que está abajo:

```
public class TemperatureHub : Hub
{
    // Create the instance of ChartDataUpdate
    private readonly LastTemperatureUpdate _instance;
    public TemperatureHub() : this(LastTemperatureUpdate.Instance) {}
    public TemperatureHub(LastTemperatureUpdate instance)
    {
        _instance = instance;
    }
    public void InitTempData(string data = "")
    {
        _instance.idUsuario = data;
        _instance.GetTempData();
    }

    public void GetTempData()
    {
        _timer = new Timer(TempTimerCallback, null, _updateInterval, _updateInterval);
    }
    private void TempTimerCallback(object state)
    {
        if (_sendingLastTemperature)
        {
            return;
        }
        lock (_tempUpdateLock)
        {
            if (!_sendingLastTemperature)
            {
                _sendingLastTemperature = true;
                SendLastTemperature();
                _sendingLastTemperature = false;
            }
        }
    }

    private void SendLastTemperature()
    {
        tempUpdate.TakeLastTemp(this.idUsuario);
        GetAllClients().All.UpdateTemperature(tempUpdate);
    }
}
```

Por lo tanto, tenemos un ChartHub - que es un componente SignalR que centraliza la comunicación con los clientes (navegadores) - y en el inicializador del concentrador definimos un temporizador con un método de devolución de llamada Interval. Cuando se ejecuta ese método, lee la última temperatura de la base de datos. Luego con SignalR API envía la nueva información (la última temperatura de cada sensor) a todos los clientes (los navegadores abiertos en la vista del tablero).

El código de abajo es la vista del tablero de mandos, que en realidad es responsable de que el mensaje sea enviado por el servidor y luego tomar esos datos y actualizar la interfaz de usuario. Por motivos de simplicidad no pegaré todo el JS y HTML, solo la pieza que recibe los datos. El resto del código está en el repositorio de GITHUB.

```
<script>
$.ready(function () {
  $(function () {
    $("#ContenedorPreload").show();
    $("#content").hide();
    //Create the Hub
    var temperatureHub = $.connection.temperatureHub;
    //Call InitChartData
    $.connection.hub.start().done(function () {
      temperatureHub.server.initTempData($("#usery").
    val());
  });
  temperatureHub.client.updateTemperature = function
  (tempUpdate) {
    UpdateTemperature(tempUpdate);
  };
});
});
</script>
```

"...este sensor en particular envía datos a un sitio web en tiempo real y está conectado a un refrigerador en una farmacia, que por ley necesita tener temperatura controlada..."

Y el resultado final es un tablero que muestra la temperatura en tiempo real, como muestra esta captura de pantalla:

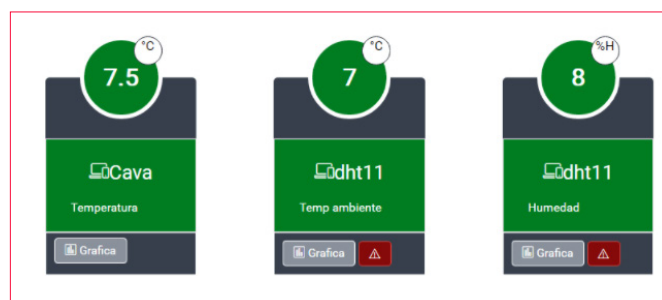


Imagen 1.- Tablero mostrando la información recogida.

El código completo de este proyecto está en este repo de GitHub.

<https://github.com/levalencia/IoTRealTimeSensorWebApp>

Agradecimientos

Mis agradecimientos a Gina Ospina que me ha ayudado con la traducción de este artículo desde el original publicado en el blog de MVP Award.

LUIS ESTEBAN VALENCIA

Office Servers & Services MVP

@levalencia

Cloud PBX ¿Cuál es la solución adecuada para mí?

La voz empresarial con Microsoft es totalmente basada en Voz sobre IP (VoIP por sus siglas en ingles) y se utiliza mediante solución de software, con el cliente de Skype for Business (anteriormente Lync, anteriormente OCS, anteriormente LCS y mejor aquí nos detenemos). Esta solución puede aplicarse 100% OnPremises, ofreciendo una gama amplia de funcionalidades tales como Grupos de Respuesta, Parqueo de Llamadas, etc.

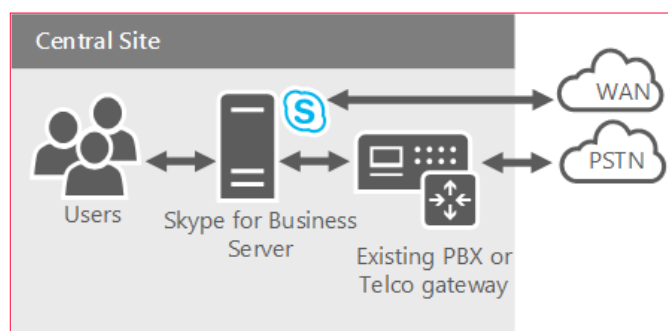


Imagen 1.- Configuración de Voz Empresarial OnPremises.

Muy bonito, pero ¿en donde entra la oferta de Cloud PBX? La oferta de CLOUD PBX, y remarco la palabra cloud, porque como imaginan es solamente para cuando los usuarios se encuentran en la nube. Y como con la mayoría de los productos Microsoft tenemos diferentes opciones para elegir.

“...La voz empresarial con Microsoft es totalmente basada en Voz sobre IP y se utiliza mediante solución de software, con el cliente de Skype for Business...”

Cloud PBX con Servicio de Llamadas a la PSTN (PSTN Calling). Solución 100% en la nube, los usuarios están hospedados en la nube al igual que es servicio de PSTN. Cuando se adquiere este servicio (ya sea mediante licencia E5 o E3 + Add on) Microsoft será nuestro proveedor de servicios de telefonía. Y solamente debemos elegir los números que necesitamos dentro de una lista disponible para cada país, asignarlos a nuestros usuarios y serán capaces de realizar y recibir llamadas desde su cliente de Skype for Business.

Cloud PBX con conectividad local a la PSTN (PSTN on-premises connectivity). Con esta opción las llamadas son realizadas y recibidas a través de nuestro proveedor de servicio

de telefonía localmente. Aunque los usuarios viven en la nube, todo el tráfico de voz se enruta para hacer uso de la solución de telefonía actual (PBX, Gateway, SBC o Direct SIP). Para ello también es necesario contar con alguno de los siguientes.

- Skype for Business Server existente. Para poder conectar los usuarios de la nube a la infraestructura de voz local, necesitamos un punto de conexión entre ambos. Si ya se cuenta con una implementación de Skype for Business Server, y se está realizando la migración de los usuarios a la nube, podemos ocupar esta infraestructura para que sirva como “puente” entre ambos servicios.
- Skype for Business Cloud Connector Edition. Caso contrario, no tenemos una implementación de Skype for Business local, pero necesitamos hacer uso de los servicios de PSTN locales. Es suficiente con hacer uso del Cloud Conector Edition, que no es más que los servicios básicos necesarios para poder funcionar como “puente” entre la nube y el servicio local, en lugar de realizar una implementación completa de Skype for Business Server, CCE (su abreviatura) es un conjunto de máquinas virtuales con los requerimientos mínimos para realizar la conexión.

NOTA: Esta solución es solo para organizaciones que no cuenta con alguna implementación existente de Lync o Skype for Business.

***Al día de publicación de este artículo no está soportado aún no está soportado la coexistencia entre CCE y alguna implementación de Lync o Skype for Business.**

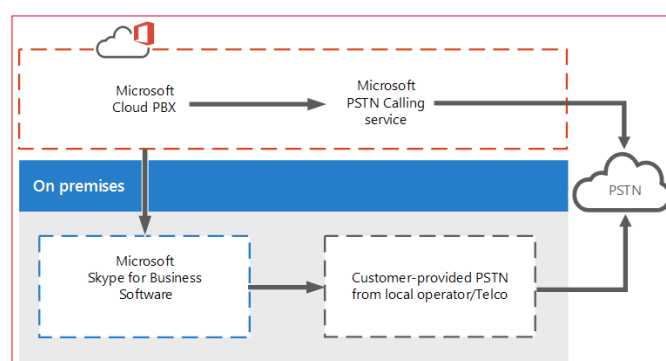


Imagen 2.- Soluciones Cloud PBX.

Ya que tenemos todos los escenarios definidos, ¿cómo elegir cual es el mejor para mí? Para esto podemos basarnos

en el siguiente diagrama.

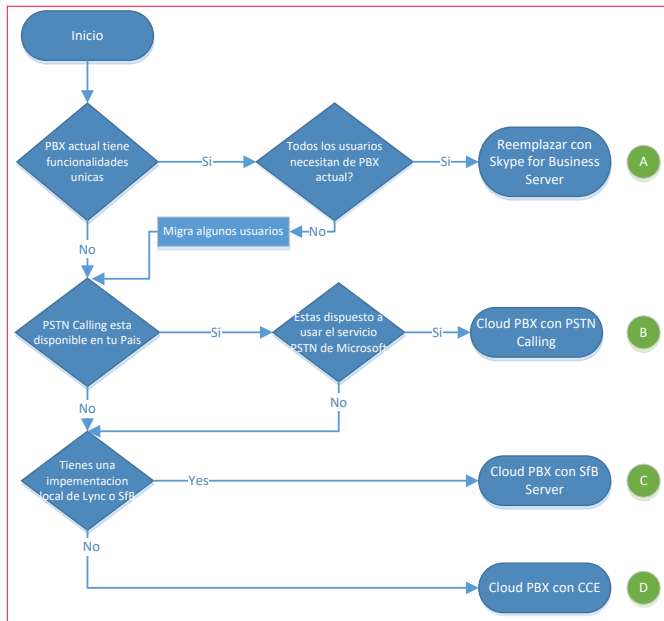


Imagen 3.- Proceso para seleccionar el escenario adecuado.

Si eres parte de escenario A. Aun tienes servicios que necesitas utilizar que solo una solución local puede ofrecerte y por lo tanto la nube no es una solución adecuada para ti. También un punto determinante que les recomiendo que revisen es los casos de uso que necesitan, algunas de las funcionalidades de Cloud PBX se encuentran aún en preview o dentro de Road Map, tales como Auto Attendant, Grupos de Respuesta o el soporte para teléfonos de Área común. Entonces analicen con suficiente tiempo todos los escenarios donde utilizan telefonía y busquen si Cloud PBX puede ofrecerlo. En caso contrario vean si Skype for Business Server lo puede ofrecer, y planeen reemplazar su PBX actual con Skype for Business Server.

Si después de seguir el diagrama quedas en el escenario B. ¡Felicidades! Formas parte de los afortunados que tienen el privilegio de utilizar el servicio de PSTN de Microsoft. Actualmente este servicio solo se encuentra disponible para los siguientes países. Francia, Puerto Rico, España, Estados Unidos y Reino Unido. Si, por el contrario, estás fuera de esta lista, ya sabes que el escenario B no está hecho para ti.

Aquí es interesante resaltar que puedes tener a algunos usuarios en la nube con funcionalidad “limitada” que ofrece PSTN Calling, y otros usuarios localmente utilizando los servicios avanzados con tu solución de PSTN on prem. (Configuración Híbrida)

Aquellos que terminaron en el escenario C, por alguna razón no están convencidos de utilizar PSTN Calling con Microsoft como proveedor, o porque no está disponible en sus países (la mayoría de nosotros) y cuentan actualmente con una infraestructura de Lync o Skype for Business Server. Para ustedes ya tienen la mayoría de trabajo realizado. Recomiendo asegurarse que tengan actualizados sus servidores para que no exista algún problema, y que tengan adicional al Front End Server, los roles de Mediation Server (puede ser colocado en el Front End) y Edge Server. Basta con que inicien la configuración de su ambiente híbrido para mover a los usuarios que necesitan a la nube, y utilizar la implementación actual para la PSTN.

Y para aquellos con el escenario D como resultado, por alguna u otra razón no pueden hacer uso de la oferta de PSTN de Microsoft, y no tienen una solución actual de Skype for Business. No se preocupen, para no invertir en una solución completa la cual no van a necesitar al 100, ustedes pueden implementar Cloud PBX con CCE. La serie de máquinas virtuales diseñadas por Microsoft solamente para esta solución. Necesitan contar con Hyper V para poder configurar estas máquinas (por el momento es el único Hypervisor soportado) y realizar su ambiente híbrido. Dentro de estas máquinas se encuentran los siguientes roles. AD, Mediation, Edge y CMS.

Espero que con esto puedan tener una visión más clara sobre qué solución es adecuada de acuerdo al escenario que tienen, cabe mencionar que una empresa puede tener PSTN Calling en un sitio y Cloud PBX con PSTN Connectivity en otros, todo se adapta a las necesidades de la empresa.

RODOLFO CASTRO AGUILAR
 MVP Office Server and Services
 @ucblogmx
<http://ucblogmx.com>

Integración de SharePoint con Power BI

Con la inserción definitiva de Power BI en el mercado y el crecimiento continuo de la misma, es cada vez más común para las empresas en diferentes verticales el realizar el cambio lógico de tecnología y pasarse a esta herramienta en pleno crecimiento, la cual en el ámbito de Business Intelligence, integración de datos, y tecnologías Microsoft, Power BI es el abanderado.

El crecimiento, como en todos los casos, genera nuevas posibilidades y necesidades a ser resueltas, y he aquí donde los expertos en tecnologías Microsoft deben interceder para volver realidad las necesidades presentadas. En base a esto, el mayor requerimiento, en lo que es la era de la información y comunicación en la que nos encontramos, es la integración, confidencialidad y rapidez de respuesta ante el cambio. Tres hitos difíciles de alcanzar, pero imperativos de cumplir y en este escenario no fueron ajenos estas necesidades.

Situación de Negocio

Una importante empresa relacionada a la minería y abastecimiento de maquinaria industrial con un fuerte enlace con tecnologías Microsoft y en especial con la utilización de SharePoint como principal herramienta de almacenamiento de datos corporativos, de visualización e integración de distintas componentes sensibles de la empresa.

En los últimos meses ha empezado a utilizar Power BI como herramienta de reportería y visualización de información de forma dinámica, logrando una gran aceptación dentro de la cúpula gerencial y expertos de las diversas áreas, generando la migración de reportes legados de Reporting Services a Power BI, así como múltiples ramificaciones que debido a las limitaciones de las herramientas utilizadas, resultaban inviables de realizar.

Situación Actual vs Deseada

Con el crecimiento presentado y utilizando a SharePoint como principal plataforma de visualización de los reportes, se recurrió a la necesidad de visualizar reportes de Power BI embebidos en la plataforma, pero con la particularidad de que el acceso a los reportes no fuera de forma pública y que el acceso a los mismos no pueda ser compartido fuera del ámbito de SharePoint.

A su vez, en la reforma estructural a nivel tecnológico que

está inmersa la empresa, se encuentra la migración de la plataforma SharePoint on-premises a su versión online, por lo que la solución debe contemplar la necesidad de poder ser utilizada indistintamente en ambas versiones.

Análisis

Los requerimientos presentados, constaban de dos grandes hitos, integración de Power BI con SharePoint y tipo de paquete de SharePoint a ser utilizado para contemplar la múltiple plataforma.

Dentro de las integraciones de Power BI, ya sea con SharePoint o cualquier otra plataforma o sitio web se encuentran tres variantes:

- Embeber un reporte en un iframe (situación actual).
- Utilización de la REST API de Power BI.
- Power BI Embedded.

Embeber un reporte en un iframe

La primera de las posibilidades permite de forma rápida el compartir cualquier reporte en un iframe, algo que resulta atractivo debido a la rapidez de integración pero tiene la gran desventaja de que para poder visualizar un reporte el mismo debe ser publicado de forma pública.

Esto implica que cualquier usuario interno o externo con la URL del reporte podría acceder al mismo, brecha de seguridad con posible mitigación, si se considera que la única puerta de entrada a los reportes se realizará a través de SharePoint y su estructura de seguridad inherente.

“...el mayor requerimiento, en lo que es la era de la información y comunicación en la que nos encontramos, es la integración, confidencialidad y rapidez de respuesta ante el cambio...”

De todas formas, la gran desventaja no solo radicó en lo antes mencionado, sino que se encontró en la estructura del iframe con el reporte embebido que presentaba Power BI. En el iframe se presenta la opción de compartir el reporte, es decir se puede obtener la URL del mismo, por lo

que la posibilidad de mitigar la brecha de seguridad resulta inviable a través de este approach.

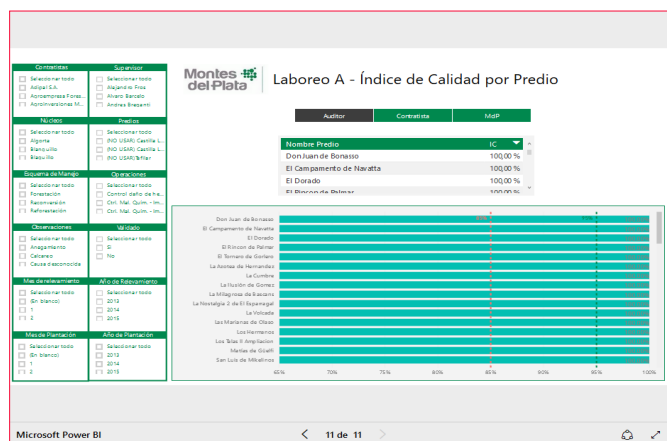


Imagen 1.- Opción de compartir un informe de Power BI.

Utilización de la REST API de Power BI

La segunda opción era la utilización de la API REST de Power BI, con este approach se cumplen las restricciones de seguridad y las presentadas por Power BI, dado que el reporte no debe ser publicado de forma pública y se accede con la autenticación de usuario/contraseña o con el token del usuario, pero lamentablemente tiene dos limitantes, las cuales son la autenticación automática y la duración del token.

La autenticación automática, es decir el ingreso de credenciales y posterior obtención del token de acceso se debe realizar desde el servidor, dado que la API REST valida que la solicitud provenga de una aplicación registrada y dada la restricción de que la solución debe ser apta para SharePoint Online esto no era posible. A su vez, uno de los requerimientos subyacentes, era la necesidad de que el usuario no tuviera que ingresar las credenciales de la cuenta de Power BI, dado que la mayoría de los usuarios solo consumirán la información, por lo que no conocerán las credenciales de Power BI.

El otro talón de Aquiles de esta posibilidad es el tiempo de vida del token (aproximadamente una hora), lo que imposibilitaba la utilización de la solución en monitores de visualización estáticos, es decir, monitores cuya función es mostrar todo el día determinados reportes y tableros de control.

Power BI Embedded

Como opción adicional tenemos Power BI Embedded. Esta solución, en comparación a las otras conlleva un tiempo de desarrollo y aplicación mayores e involucra otros servicios de Microsoft, lo que genera gastos extras para el cliente, pero que ofrece toda la potencialidad de Power BI y el manejo inteligente de acceso al mismo.

La pregunta que surge es como implementarlo, y el gran secreto de su funcionamiento es abstraerse de todo y dejar que Azure haga el manejo de los accesos, la generación de tokens y la seguridad inherente a la aplicación, lo que con-

lleva a que Power BI únicamente se centre en hacer lo que mejor sabe hacer, presentar la información y mantenerla actualizada.

Básicamente para generar un reporte embebido con Power BI embedded, debemos contar con las siguientes suscripciones:

- Cuenta de Power BI Pro.
- Cuenta asociada de Azure.

El siguiente paso es crear un Workspace Collection y un Workspace de Power BI dedicado para la solución. Como último paso será registrar los reportes en Azure, esto registra los reportes en el Workspace de PowerBI, permitiendo integrarlos con los disponibles en Power BI Web, pero con la particularidad de que Azure podrá ofrecer un token a dicho reporte, de forma de poder controlar el acceso.

La integración entre Azure y Power BI radica en lo siguiente:

- 1.- Azure provee un Access Key necesario para la solitud del token (este Access Key estará contenido en la solución de SharePoint).

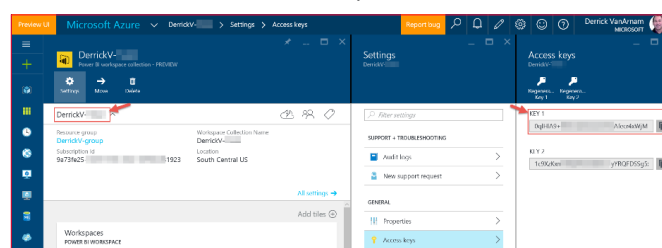


Imagen 2.- Access Key en Azure.

- 2.- El segundo paso es generar el token, el cual tendrá un tiempo de expiración establecido:

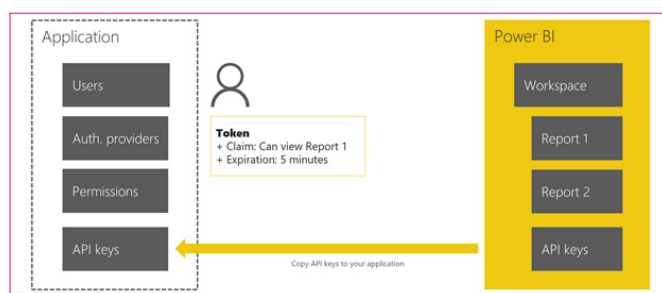


Imagen 3.- Generación del token.

- 3.- El token es firmado con el Access Key provisto por Azure y almacenado en la aplicación.

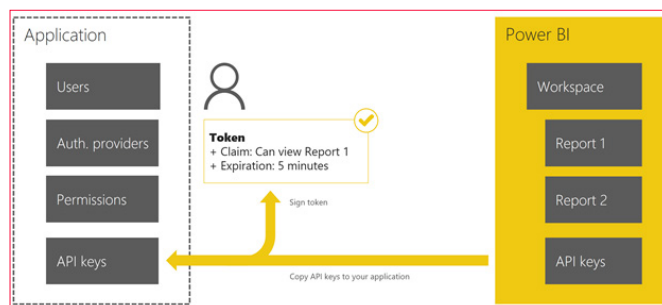


Imagen 4.- Firmado del token.

- 4.- El siguiente paso es la solicitud de visualización del reporte (a través del código del mismo, el cual

es provisto por Azure al subirlo al Workspace) con el token generado

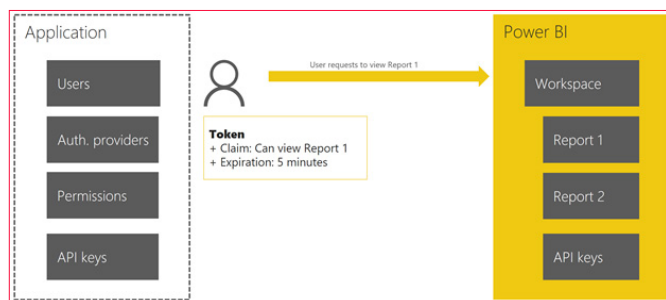


Imagen 5.- Solicitud de la visualización del informe.

5.- Azure, a través del Workspace de Power BI verifica el token y el reporte solicitado.



Imagen 6.- Verificación del token.

6.- Como último paso el reporte es enviado al usuario.

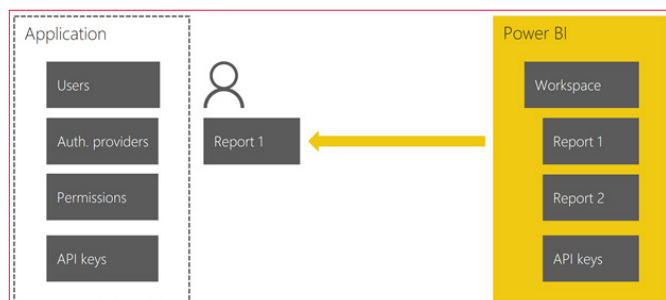


Imagen 7.- Envío del informe al usuario.

El token generado tendrá un tiempo de expiración análogo al de la API REST, pero a diferencia de la misma, no es necesario realizar una autenticación del usuario, ya que teniendo el Access Key del Workspace y generando JSON Web Tokens, se puede actualizar de forma automática el token, sin necesidad de contar con un servidor.

Comparación

Luego del análisis de cada una de las características de las opciones disponibles y cruzándolas con los requerimientos solicitados, tenemos lo siguiente:

REQUERIMIENTO	IFRAME EMBEBIDO	API REST	POWER BI EMBEDDED
SEGURIDAD	No	Sí	Sí
REFRESCO DE DATOS	Sí	Sí	Sí
RENOVACIÓN DE ACCESO AUTOMÁTICA	No	No	Sí
SHAREPOINT ONLINE	Sí	No	Sí
SHAREPOINT ON-PRE-MISES	Sí	Sí	Sí

Una vez definida la forma de integración de SharePoint con PowerBI, se debió definir la mejor opción de desarrollo para la solución.

Bajo circunstancias normales, la mejor y única opción viable para realizar lo solicitado, de forma de contemplar los requerimientos solicitados, resumiendo: estuviera disponible para SharePoint on-premises y Online, y que fuera una solución cerrada, es decir, un paquete, sería el desarrollo de una AppPart. En este caso aplica a su vez Microsoft SharePoint Framework (en modo Preview) ofreciéndonos una forma diferente de desarrollo.

Esta nueva herramienta de Microsoft, cambia el paradigma tradicional y elimina por completo la dependencia con el servidor, transformando un sistema pensado como cliente –servidor a cliente-cliente, ofreciéndonos una integración inmediata con plataforma de desarrollo liviano como es Visual Studio Code y proveyendo de ambientes de prueba totalmente disponibles (sin configuraciones de por medio) en SharePoint Online. (Si le interesa informarse más acerca de esta nueva herramienta, los invitamos a que busquen los artículos relacionados dentro de nuestras publicaciones)

Debido al estado de preview con el que cuenta SharePoint Framework, es que únicamente se pudo hacer una prueba de concepto (la cual la presentaremos en un futuro artículo) con esta nueva herramienta, teniendo que recurrir a la generación de una AppPart.

Conclusión

El resultado final fue la generación de una AppPart, en la cual se debe especificar el Access token del Workspace de PowerBI y el código referente al reporte a visualizar, esta solución contempla los requerimientos solicitados y nos permite manejar el acceso a información sensible de forma ordenada y jerarquizada.

BRUNO TORTELORO

Software Developer en Arkano Software

Introducing Microsoft PowerApps

Welcome to an article where we will see Microsoft's new product, "PowerApps". With just the launch, it has provided an amazing response within our technology organization as it provided an efficient way to create an app or create a flow. Microsoft PowerApps is so promising in its initial phase that once you start working on it, it's hard to get your hands off.

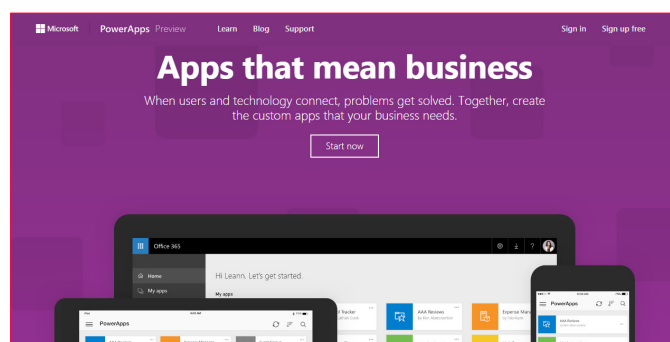


Figure 1.- PowerApps portal main page.

The PowerApps Portal

Create an App in PowerApps is a very straightforward process:

- Use this link to sign in with your organization's Office 365 ID or with your window's ID.
- Once you have been logged in, you will arrive to the portal.

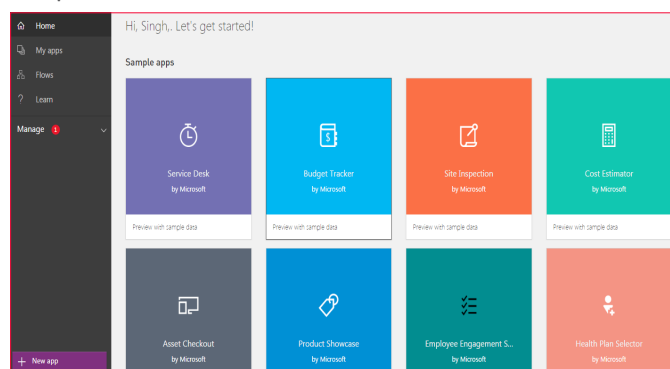


Figure 2.- PowerApps Portal.

"...PowerApps provee la respuesta dentro de nuestra organización ofreciendo una manera eficiente para crear una App o un Flujo..."

- The left tool bar provides you various options to build or explore.

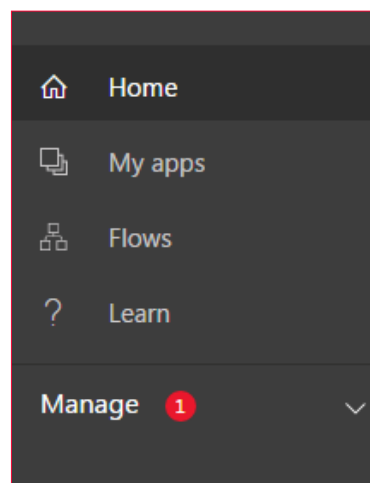


Figure 3.- Left tool bar in the PowerApps Portal.

Flows

You can create automated workflows between your apps and services to get notifications or to synchronize files, collect data and many more functionalities. Flows will provide the workflows capabilities which we have been creating through SharePoint Designer or through other 3rd part tool son SharePoint.

Now Flow will be a part of them providing automatically solutions to all the task or process. You can set up a flow that will manage email flows based on different criteria or conditions.

Whether it's a complex multistep process or a simple one-step task, flows will improve the effectiveness, productivity, and performance of the process assisting all end users to get benefitted.

Let's see how we can create one:

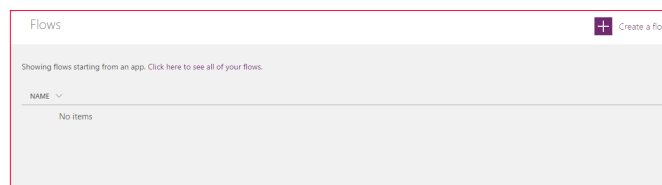


Figure 4.- Flows section in the Portal.

- Just on one click, you are ready to create a flow.
- Click on it.

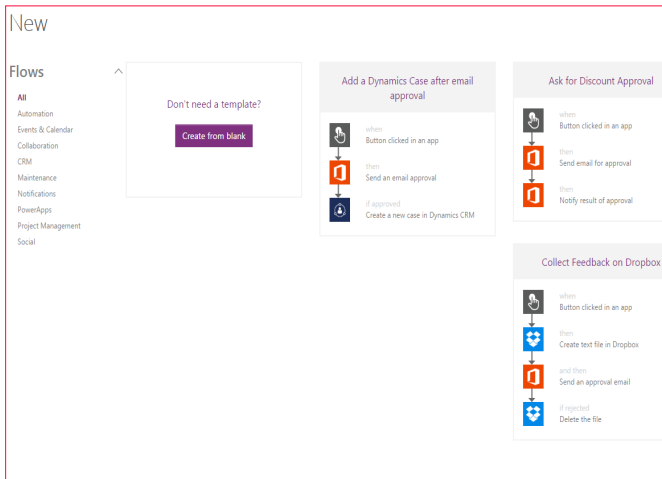


Figure 5.- Templates to create a Flow.

- You can either create a blank flow or choose from any of the available templates as you can see on Figure 5.
- We will see a demo soon in my next article so stay tuned.

Apps

Another amazing thing which PowerApps offer is creation of Apps in just few simple clicks which are compatible to all devices and browsers. As PowerApps says, Innovation just got easier. Here you can create an app while connecting to different apps which you are already using without even writing a line of code and publishing them on all devices to be used.

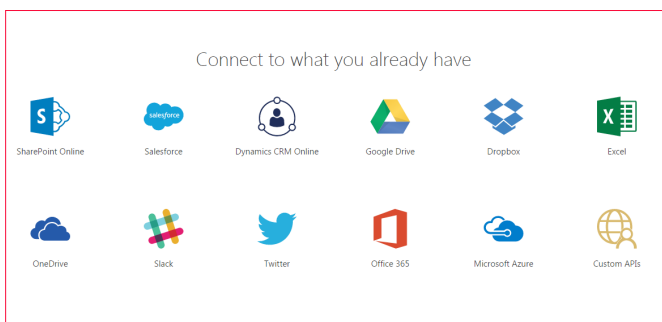


Figure 6.- Some of the services that can be used in a PowerApp.

You can connect your app from a various everyday social or custom apps and use them in your process of development your own PowerApps. You can also connect to existing data sources and service like Excel files, SharePoint lists, CRM records, your custom APIs, your drop box or any such to collaborate on your app.

The best part is you use a designer to customize or build an app that uses device capabilities, including cameras, GPS, and pen control without writing any line code.

To create your first App:

- Click on "My Apps".

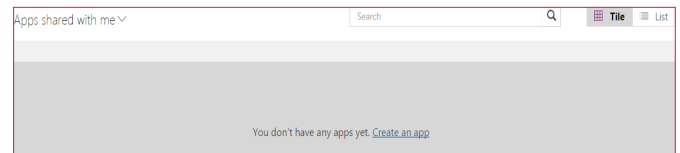


Figure 7.- Creating an App in PowerApps.

- Choose if you want the designer to be installed on your machine or you want to use a web designer on your browser.

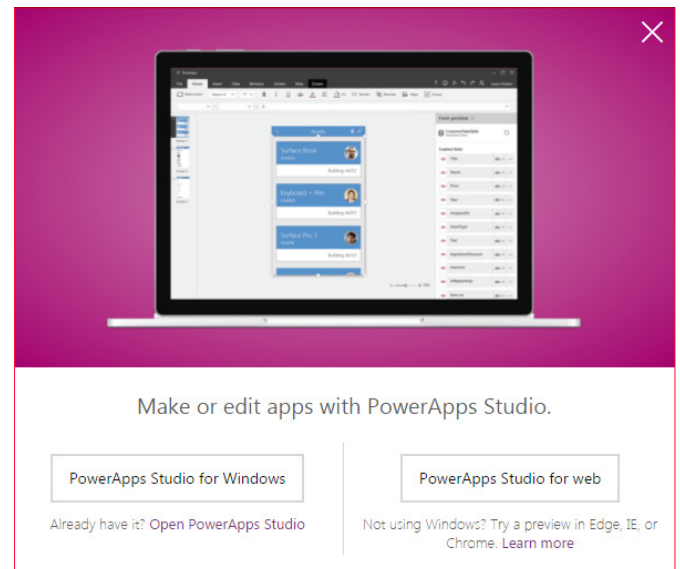


Figure 8.- Designer experiences for PowerApps.

- Once you have your studio ready, you will see the screen below.

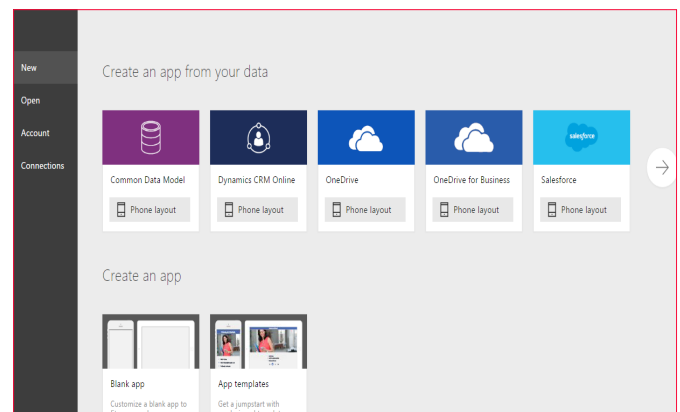


Figure 9.- PowerApps Designer

- You can either start using an application to connect to your app or create a blank app.
- We will see a demo soon in my next article so stay tuned.

In this article we saw our Microsoft PowerApps and Flows easing the lives of the customers and soon we will see their demo.

Until then, keep reading & keep learning!

MANPREET SINGH

SharePoint Consultant | Office Servers & Services MVP

Microsoft Operations Management Suite (OMS)

Este Artículo trata de hacer una breve introducción a OMS que es y cómo realizar los primeros pasos en OMS, preparar nuestro entorno y como desplegar el agente.

¿Qué es OMS?

Es una solución en la Nube de Microsoft que contiene una colección de servicios para administrar y proteger nuestra infraestructura OnPremise, en la nube o si tenemos una infraestructura híbrida. OMS se puede ejecutar como servicio independiente o si tenemos en nuestra empresa desplegado System Center 2012 R2 Operations Manager o System Center 2016 Operations Manager, podemos extender SCOM y conectar con OMS. Hay que dejar claro que OMS no es un reemplazo de System Center Operation Manager (SCOM).

También si tenemos desplegado en nuestra empresa System center 2016 Configuration Manager también podemos conectar SCCM 2016 con OMS e importar la configuración de nuestras colecciones.

OMS se divide en mayoritariamente en cuatro categorías o Características.

Insights & Analytics.

Nos ayuda a reunir, correlacionar, buscar y actuar sobre los datos de registro (Logs). Nos da conocimientos operacionales en tiempo real, utilizando una búsqueda integrada, Dashboards de actividad, etc. También nos da soluciones desarrolladas por Microsoft que tenemos disponibles mediante la Galería.

El Análisis de registro nos permite primero a recoger todos los datos de la máquina OnPremise y la infraestructura en la nube y luego realizar consultas completas de los datos de nuestro entorno a través de las búsquedas inteligentes, análisis y visualización. Análisis del registro de OMS nos permite interactuar rápidamente con datos de la máquina en tiempo real y con el histórico para desarrollar ideas y en las tomas de decisiones.

Automation & Control.

Nos ayuda a organizar y automatizar operaciones comple-

jas y repetitivas. Con la introducción de los trabajos híbridos de los runbooks. OMS amplía las capacidades desde la nube a nuestro centro de datos OnPremise. Puede automatizar tareas manuales, repetitivas, propensos a errores en los entornos de Data Center o en la nube. También podemos implementar procesos administrativos que consisten en poder realizar múltiples tareas sin la intervención humana.

Protection & Recovery.

Nos ayuda a proteger y ampliar nuestro centro de datos más fácilmente implementar estrategias de backup y disaster recovery. Las soluciones de backup y recuperación son impulsadas por Azure Backup y Azure Site Recovery. Las capacidades de replicación de Azure Site recovery ayudan a proteger las aplicaciones críticas y ampliar nuestro centro de datos a Azure. También nos permite recuperación, desarrollo y pruebas, así como migrar aplicaciones a Azure. Azure Backup es una solución escalable que protege nuestros datos de uso y se conserva durante años, sin cualquier inversión de capital y con mínimos costos de operación.

Security & Compliance.

Nos ayuda a identificar, evaluar y mitigar los riesgos de seguridad de nuestra infraestructura. Nos ayuda mejorar la protección de las cargas de trabajo, servidores, evaluaciones de actualización de sistema, auditoría, evaluación de malware y análisis de incumplimiento. Security And Compliance nos permite identificar actualizaciones de sistema que nos faltan y el estado de malware. También puedes acumular eventos de seguridad y realizar una auditoría forense y de análisis de incumplimiento. Puede reconocer amenazas tempranas y mejorar el cumplimiento. También puede capturar y examinar la comunicación de red para determinar si hay que cualquier comunicación que ocurre con direcciones IP maliciosas.

Primeros pasos con OMS

Ya sabemos que es OMS y sus características o categorías principales ahora vamos hacer los primeros pasos con OMS y para empezar vamos a trabajar con una cuenta gratuita.

- Entramos al sitio web de OMS para activar una trial de OMS: <https://www.microsoft.com/es-es/cloud-platform/operations-management-suite-trial>
- Iniciamos una sesión con una cuenta de Microsoft.



Imagen 1.- Inicio de sesión en OMS.

- Nos avisa que no tenemos área de trabajo. Un área de trabajo es la forma de agrupar, organizar por ejemplo hacemos un área de trabajo por producción otra área de trabajo de Pre-producción etc. Ya sabemos lo que es un área de trabajo así que vamos a crearla ponemos la información que nos pide y hacemos clic en crear.

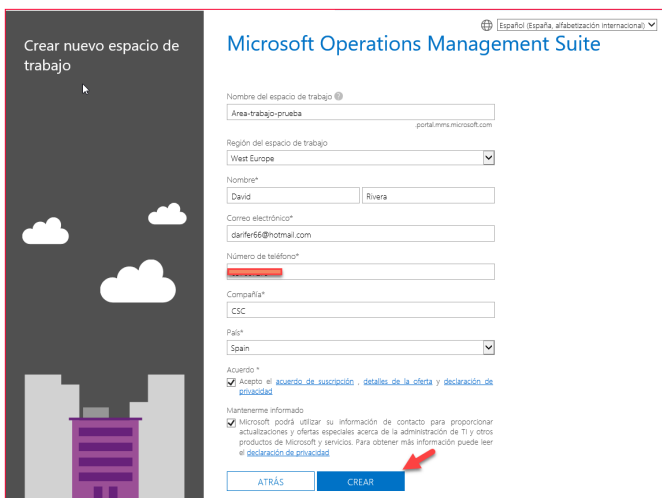


Imagen 2.- Definición de un área de trabajo en OMS.

- Como yo no tenemos una suscripción de Azure y es necesaria para OMS le damos crear nuevo rellenamos toda la información. Hacemos clic en vinculo para vincular nuestra área de trabajo con la suscripción de Azure.

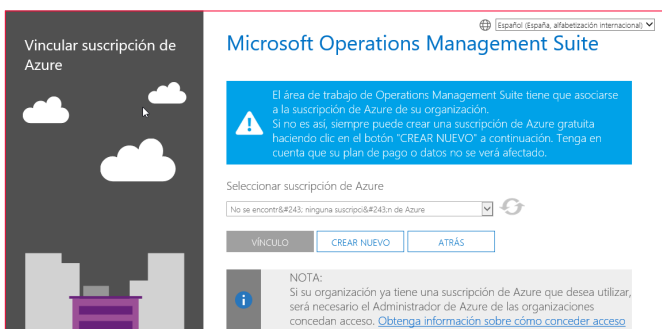


Imagen 3.- Creación de la suscripción de Azure para OMS.

- Ya tenemos nuestra área de trabajo creada:

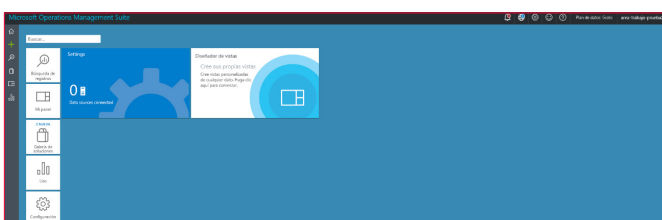


Imagen 4.- Área de Trabajo creada.

- Si hacemos clic en la galería de las soluciones veremos las cuatro características o categorías principales y las demás soluciones que tenemos disponibles. Estas soluciones se van ampliando ya que OMS avanza mucho.

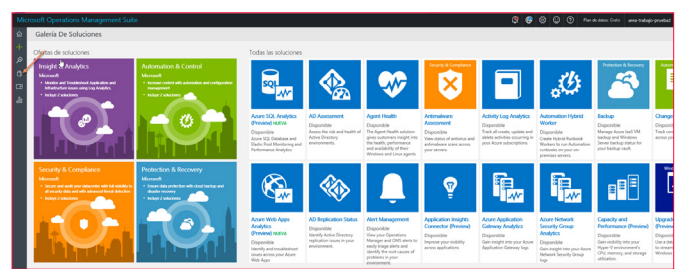


Imagen 5.- Galería de soluciones.

Instalación del agente de OMS.

Vamos a ver como se instala un agente en un servidor Windows:

- Desde la pantalla principal hacemos clic en el icono de "Home" y después clic en configuración.

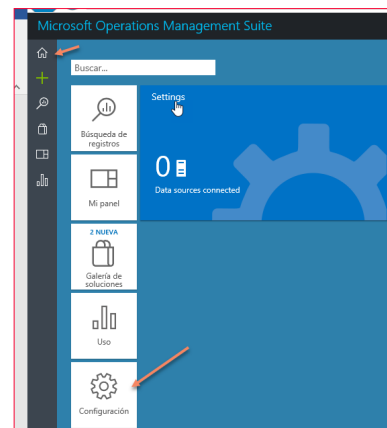


Imagen 6.- Instalación del agente de OMS.

- Después hacemos clic en Connected Sources y vemos la siguiente ventana.

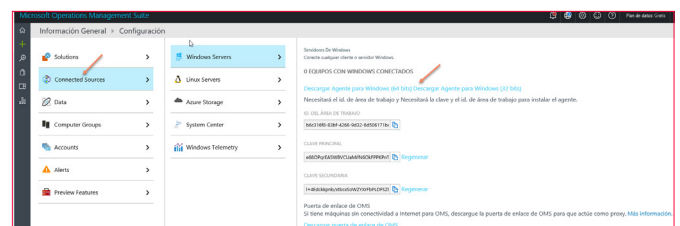


Imagen 7.- Detalle de fuentes conectadas en OMS.

- Hacemos clic en Descargar agente para Windows, lo descargamos y lo ejecutamos.

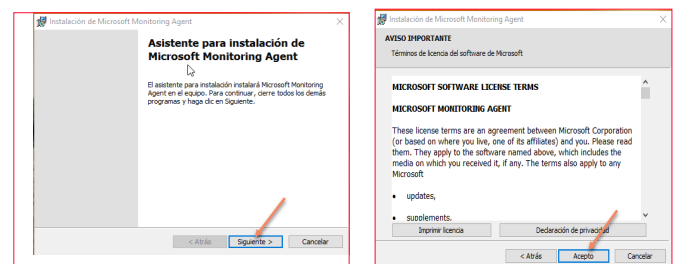


Imagen 8.- Inicio de la instalación del agente de OMS.

- Es muy sencillo seguimos las indicaciones de la imagen y en menos de 5 minutos tenemos instalado el agente de OMS.

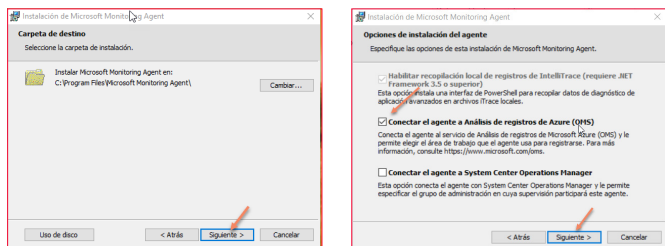


Imagen 9.- Configuración de la instalación del agente.

- En la siguiente ventana nos pide Id de área de trabajo y la clave de área. Esta información la sacamos desde la ventana donde hemos descargado el agente. Copiamos la información y la pegamos y hacemos clic en siguiente.

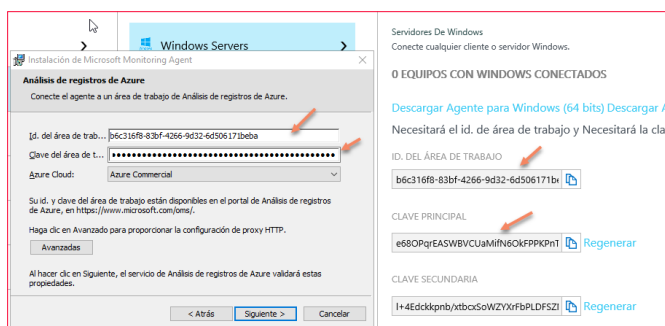


Imagen 10.- Configuración del ID y Clave del área de trabajo.

- En la siguiente ventana hacemos clic en instalar.

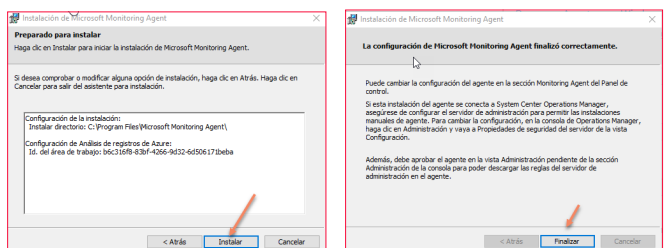


Imagen 11.- Finalización de la instalación del agente de OMS.

- Hacemos clic en finalizar y como se ha visto, la instalación es un proceso sencillo que lleva menos de 5 minutos. Para finalizar, esperaremos unos minutos y ya tendremos el servidor conectado a OMS

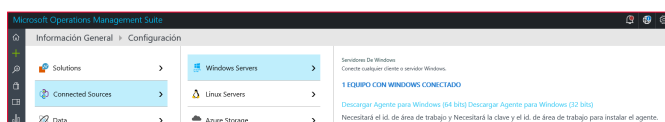


Imagen 12.- Servidor conectado a OMS

“Es una solución en la Nube de Microsoft que contiene una colección de servicios para administrar y proteger nuestra infraestructura OnPremise, en la nube o si tenemos una infraestructura híbrida”

Configuración de la extensión de SCOM a OMS

Para configurar la extensión de SCOM a OMS:

- Abrimos la consola de SCOM vamos al nodo de administración hacemos clic en Operations management Suite hacemos clic en Register with Operations Management Suite.

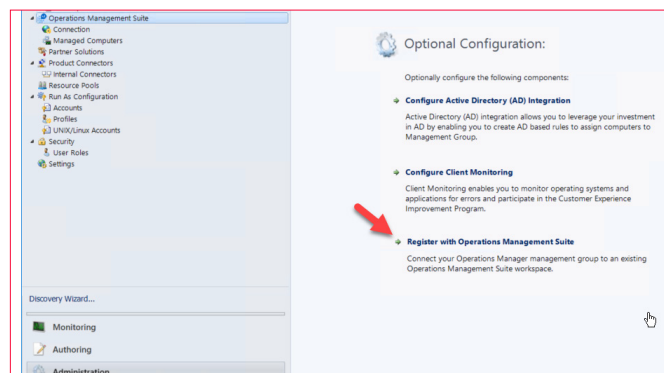


Imagen 13.- Registro de OMS en SCOM.

- Nos pedirá que seleccionemos el espacio de trabajo. Seleccionamos el espacio de trabajo creado en el punto anterior.

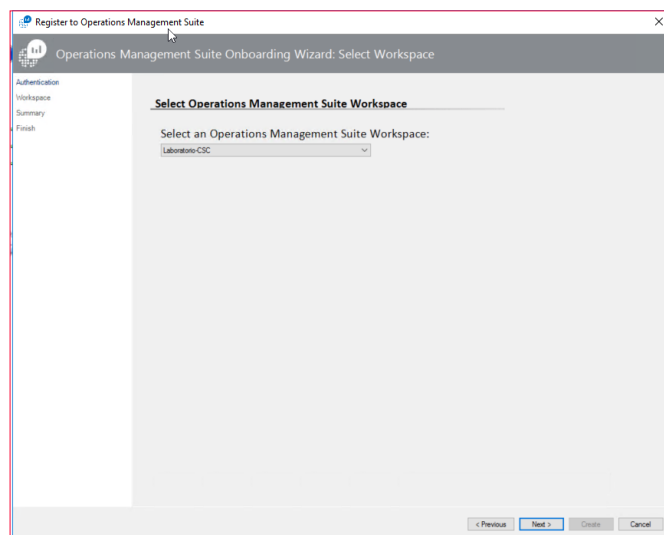


Imagen 14.- Selección del espacio de trabajo.

- En esta ventana nos pide la cuenta de Microsoft con la cual administramos OMS. La introducimos y hacemos clic en Sign in.

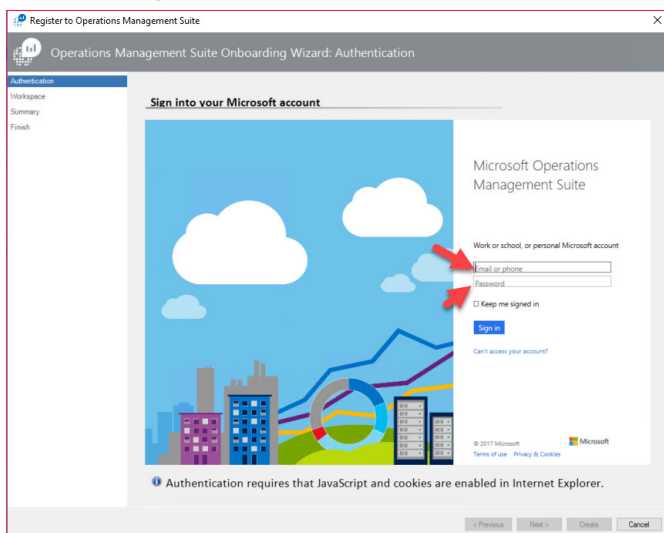


Imagen 15.- Inicio de sesión en OMS.

- En esta Ventana nos muestra la información de la cuenta que hemos utilizado y el espacio de trabajo que hemos utilizado para la conexión de SCOM con OMS. Hacemos clic en Close y ya tenemos conectado SCOM con OMS.

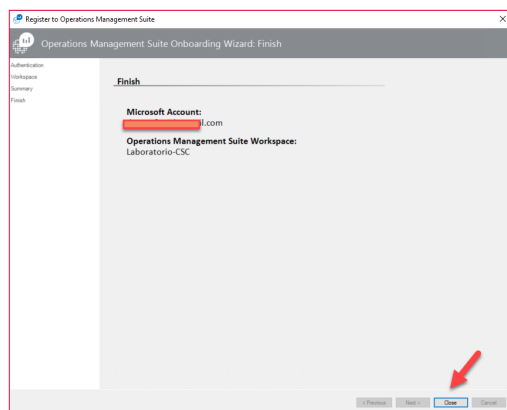


Imagen 16.- Fin de la conexión de SCOM con OMS.

Conclusiones

En este artículo se ha detallado que es OMS y los elementos que lo componen, se ha realizado una introducción a la creación de espacios de trabajo para OMS, se ha realizado una instalación del agente de OMS y finalmente se ha extendido System Center Operation Manager a OMS.

DAVID RIVERA

Microsoft MVP Cloud & Datacenter Management

@darifer66

En **encamina** buscamos:

- ★ Desarrolladores .NET
- ★ Desarrolladores Dynamics 365
- ★ Consultores Office 365
- ★ Consultores CRM
- ★ Consultores de Azure

Si tú también **piensas en colores**



¡ Queremos tu talento !
rrhh@encamina.com

encamina

@encamina f ENCAMINA in ENCAMINA

Integrando Bot Framework y Microsoft Teams

En el pasado número 29 de CompartiMOSS salió un artículo introductorio de Bot Framework, escrito por Santiago Porras. En este artículo voy a centrarme en como añadir lógica al bot, en este caso para poder integrarlo en Microsoft Teams. Para ver una descripción de Teams, os recomiendo el artículo de Miguel Tabera en el CompartiMOSS nº 30.

El escenario

Lo que quiero conseguir con el bot es que busque dentro de mi entorno de Microsoft Teams todos aquellos documentos en los que pueda estar interesado. Para facilitar el escenario, voy a construir un bot mínimo que va a buscar los documentos que están en los Teams del usuario, buscando por las palabras que introduzca el usuario en su conversación.

La lógica del bot

El bot no deja de ser una aplicación Web API, con unas acciones concretas mapeadas a los verbos HTTP. La más importante es la operación POST, con la siguiente firma:

```
public async Task<HttpResponseMessage> Post([FromBody]
Activity activity)
```

Aquí se nos pasa la conversación como parámetro llamado activity. Dentro del objeto activity tenemos acceso al texto del mensaje (propiedad Text) y al usuario originador del mensaje (propiedad From).

Para responder al mensaje, usaremos un ConnectorClient, inicializándolo con el parámetro ServiceUrl del propio mensaje. De esta manera responderemos al mismo canal que está usando el usuario. Cabe recordar que un mismo bot puede interactuar mediante varios canales, como pueden ser Skype, chat, Teams, email, Facebook y muchos más.

Con este código tendríamos un bot mínimo que respondería al usuario en modo "eco" para repetir el texto introducido previamente:

```
ConnectorClient connector = new ConnectorClient(new Uri(ac-
tivity.ServiceUrl));
Activity reply = parentActivity.CreateReply($"Hola, has dicho
{activity.Text}");
await connector.Conversations.ReplyToActivityAsync(reply);
```

Integrando Microsoft Teams

Ahora tenemos un bot sencillo. Lo que nos falta es integrarlo con Microsoft Teams.

Hemos dicho que un bot es una aplicación web. Por lo tanto, para interactuar con Microsoft Teams tenemos que usar la Graph API de Microsoft. Voy a dar por supuesto que ya conocéis Graph API (y si no, en CompartiMOSS nº 25 Rubén Toribio os lo explica).

En concreto, usaré la API de grupos de Graph para poder encontrar los Teams que tiene un usuario. Todos los sitios de Teams son grupos de Office 365, así que si enumero los grupos de un usuario ya tendría acceso a ellos.

Para acceder a los grupos de un usuario tenemos dos vías. Una es a partir del endpoint "/me" donde cada usuario tiene enumerados los recursos de Office 365 a los que tiene acceso. Sin embargo, nuestro bot no tiene las credenciales del usuario así que dentro de "me" no encontrará nada.

Nuestro bot es una aplicación de servidor con su propia identidad y código secreto de autenticación, que usa la librería ADAL.NET para autenticarse contra Office 365. Notad que para que el bot puede enumerar los usuarios y acceder a los ficheros de grupos, hay que darle permisos a la aplicación de acceder a los recursos. Esto se hace en la pantalla de configuración de la aplicación en Azure Active Directory.

Los permisos exactos que necesita son: leer las colecciones de sitio del tenant, leer todos los grupos, leer los datos del directorio y leer los perfiles completos de usuarios (todos dentro de Microsoft Graph).

The screenshot shows the 'single sign-on' configuration page in the Azure AD portal. It displays the 'permissions to other application' section, which is expanded to show a list of permissions for the 'Microsoft Graph' application. The permissions are as follows:

Application	Application Permissions	Delegated Permissions
Microsoft Graph	4	0
Windows Azure Active Directory	1	4
Office 365 Exchange Online	2	1
Office 365 SharePoint Online	2	1

The 'permissions to other application' section is expanded, showing a list of permissions for the 'Microsoft Graph' application. The permissions are as follows:

- ☐ Read and write calendars in all mailboxes
- ☐ Read contacts in all mailboxes
- ☐ Read and write contacts in all mailboxes
- ☒ Read all groups
- ☐ Read and write all groups
- ☒ Read directory data
- ☐ Read and write directory data
- ☐ Read and write devices
- ☒ Read all users' full profiles
- ☐ Read and write all users' full profiles
- ☐ Read and write all user mailbox settings (previous versions)

Imagen 1.- Configuración de los permisos requeridos para el Bot en Azure AD.

Una vez que el bot está autenticado, tengo que buscar en Graph API los datos del usuario que está conversando con el bot. Recordad que tenemos el usuario en la propiedad From del parámetro activity en el código del bot. Si el usuario es un usuario de Office 365, su identidad (parámetro From.Id) serán tres cadenas de caracteres alfanuméricos separadas por dos puntos (:). Una es el tenant, otra la cadena "orgid" y la tercera es el ID del usuario. Nos interesa la última cadena, la del identificador. Este identificador lo podemos usar en Graph API para obtener el usuario.

```
var userId = activity.From.Id;
var userIdSegments = userId.Split(':');
var userIdGuid = Guid.Parse(userIdSegments[userIdSegments.Length - 1]);
```

Cuidado en este punto, porque según parece, la forma del ID de usuario puede cambiar hasta que Bot Framework salga del preview. El equipo del producto dice que habrá una manera diferente de obtener el usuario dentro de Office 365, probablemente con una propiedad adicional de la Activity que recibe el bot.

Graph API

En vez de interactuar con Graph API usando llamadas REST, aprovecharé la librería de .NET que existe (como otras tantas para diferentes lenguajes y plataformas). La base de la librería .NET es el objeto GraphServiceClient. Para crear este objeto le tenemos que pasar un autenticador, que en el caso del bot es una autenticación contra Azure AD mediante ID y secreto de la aplicación.

```
public class AzureAuthenticationProvider : IAuthenticationProvider
{
    public async Task AuthenticateRequestAsync(HttpRequestMessage request)
    {
        string clientId = "<ID CLIENTE>";
        string clientSecret = "<SECRETO DE LA APP>";

        AuthenticationContext authContext = new AuthenticationContext("https://login.windows.net/<TENANT>/oauth2/token");

        ClientCredential creds = new ClientCredential(clientId, clientSecret);

        AuthenticationResult authResult = await authContext.AcquireTokenAsync("https://graph.microsoft.com/", creds);

        request.Headers.Add("Authorization", "Bearer " + authResult.AccessToken);
    }
}

private GraphServiceClient GetAuthenticatedClient()
{
    GraphServiceClient graphClient = new GraphServiceClient(new AzureAuthenticationProvider());
    return graphClient;
}

Una vez que ya tenemos el cliente, buscamos el usuario, sus grupos y los ficheros de esos grupos:
var graphClient = GetAuthenticatedClient();
var users = await graphClient.Users.Request().GetAsync();
var user = users
    .Where(u => u.Id == userIdGuid.ToString())
    .SingleOrDefault();
var groups = await graphClient.Users[user.Id].MemberOf.Request().GetAsync();
```

```
var searchResults = new List<DriveItem>();
foreach (var group in groups.Where(g => g.ODataType == ("#microsoft.graph.group")))
{
    var drive = await graphClient.Groups[group.Id].Drive.Request().GetAsync();
    var files = await graphClient.Drives[drive.Id].Search(searchTerm).Request().GetAsync();
    searchResults.AddRange(files);
}
return searchResults;
```

En este código pasan muchas cosas. Primero buscamos los usuarios del tenant (operación Users) y en concreto el usuario con el ID que corresponda al userId que extraemos del mensaje recibido por el bot.

Luego obtenemos los grupos (de Office 365 y de seguridad) a los que pertenece el usuario, usando la operación MemberOf. Hay que hacer un poco de filtrado para quedarnos sólo con los grupos del tipo "Microsoft.Graph.Group" y no los grupos de seguridad de Active Directory.

Cada grupo de Office 365 a la vez tiene un "Drive", es decir un repositorio de ficheros. En realidad, es una biblioteca de documentos de SharePoint Online asociado al grupo. Al obtener el drive, hacemos una búsqueda dentro del drive (operación Search). Vamos guardando todos los resultados de búsqueda en una lista porque no hay una API única para buscar en todos los grupos.

Los resultados de búsqueda son instancias de la clase DriveItem, que contienen el nombre del fichero, su URL, fecha y otros metadatos.

Embellaciendo el bot con Cards

Nuestro bot es totalmente funcional pero queremos darle un toque de interfaz de usuario más bonita. Para ello disponemos de una librería llamada Cards, parte de Bot Framework. Con Cards se pueden hacer diálogos y "tarjetas" de contenido para todos los canales de comunicación.

En este caso usaré una ThumbnailCard. Dentro de la misma pondré el nombre del fichero, el autor y el botón para abrirlo.

Thumbnail Card

The Thumbnail card is a multipurpose card; it primarily hosts a single small image, a button, and a "tap action", along with text content to display on the card.

Property	Description
Title	Title of card
Subtitle	Link for the title
Text	Text of the card
Images[]	For a hero card, a single image is supported
Buttons[]	Hero cards support one or more buttons
Tap	An action to take when tapping on the card

El código que lo hace es el siguiente. Le paso como parámetro la lista de los resultados encontrados y la conversación del usuario, para poder contestar a ella.

```
private Activity GetFormattedSearchResultsReply(List<DriveItem> searchResults, Activity parentActivity)
{
    Activity reply = parentActivity.CreateReply("I couldn't find any documents matching that, sorry.");

    if (searchResults.Count > 0)
    {
        reply.Text = ($"I found {searchResults.Count} documents matching your request.");
        reply.Attachments = new List<Microsoft.Bot.Connector.Attachment>();

        foreach (DriveItem driveItem in searchResults)
        {
            List<CardAction> cardButtons = new List<CardAction>();
            List<CardImage> cardImages = new List<CardImage>();

            CardAction plButton = new CardAction()
            {
                Value = driveItem.WebUrl,
                Type = "openUrl",
                Title = "Open document"
            };
            cardButtons.Add(plButton);
            ThumbnailCard plCard = new ThumbnailCard()
            {
                Title = driveItem.Name,
                Subtitle = driveItem.CreatedBy.User.DisplayName,
                Images = cardImages,
                Buttons = cardButtons
            };
            Microsoft.Bot.Connector.Attachment plAttachment =
            plCard.ToAttachment();
            reply.Attachments.Add(plAttachment);
        }
    }
    return reply;
}
```

De esta forma el resultado del bot en el emulador se ve más agradable que un simple texto.

“El bot no deja de ser una aplicación Web API, con unas acciones concretas mapeadas a los verbos HTTP”

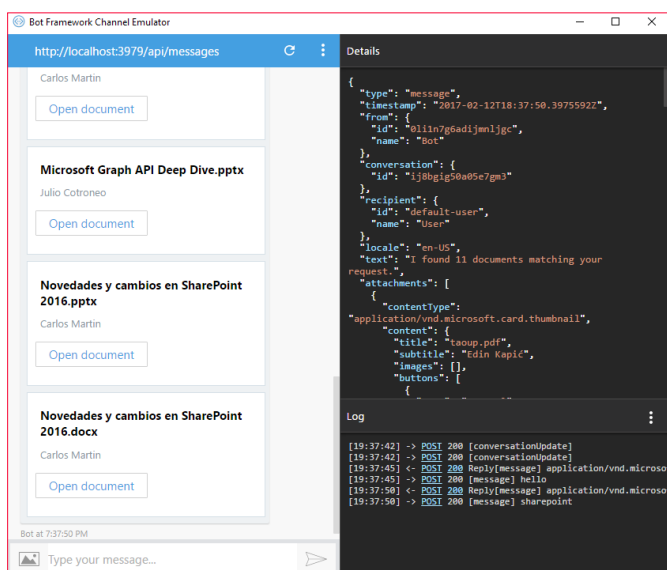


Imagen 2.- Experiencia de usuario del bot con los cambios aplicados.

Para publicar el bot en Microsoft Teams, tenemos que habilitar la carga de bots no firmados. Para ello dentro de la

administración de Teams en el portal de Administración de Office 365 hay que habilitar la opción de “Enable side-loading of external bots”.

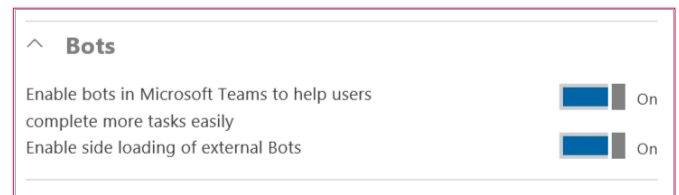


Imagen 3.- Habilitando Bots no firmados en Teams

Los bots que se pueden integrar dentro de Teams tienen que estar publicados en Bot Dashboard, tal como explica Santiago en su artículo. Del dashboard de nuestro bot publicado hay que copiar el “Microsoft App ID”.

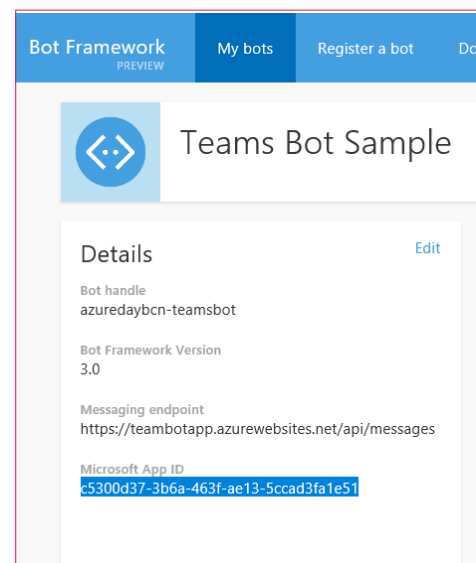


Imagen 4.- App ID del Bot.

Entonces, dentro de nuestro Team, añadimos un nuevo usuario al Team con la opción “Add members” y pegamos este ID. Teams resolverá el ID como la identidad del bot:

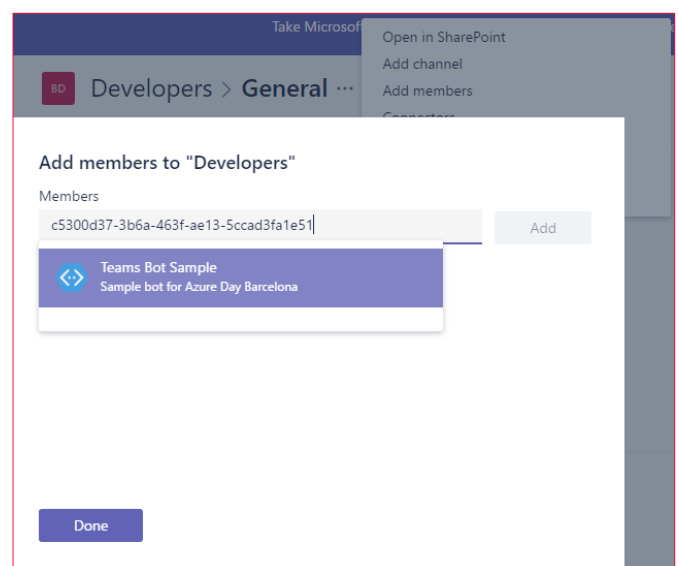


Imagen 5.- Añadiendo el Bot a Teams.

Ya podemos interactuar con el bot, mediante conversación 1:1 o en el canal del equipo usando el prefijo @ como si fuera un integrante más. Una vez desplegado y añadido en Teams, el bot está funcionando como se espera:

un diálogo con el usuario para pedir la autenticación contra Office 365.

¿Se os ocurre alguna idea más?

El código de ejemplo del bot os lo podéis bajar de mi OneDrive. Tendréis que sustituir los IDs y secretos de la aplicación por vuestros propios.

Referencias

- Código de ejemplo del bot: <https://1drv.ms/u/s!AjFh3Us-no9YnsaB0-dix9oqR-ym9DA>

EDIN KAPIC

Arquitecto senior de SharePoint y Team Lead en Sogeti

Email: edin.kapic@sogeti.com

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

Blog: www.edinkapic.com

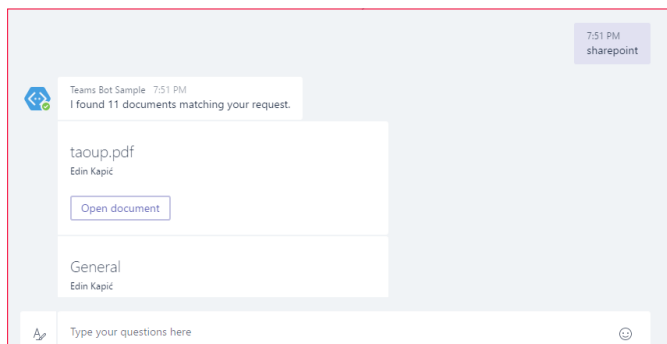


Imagen 6.- Bot en funcionamiento.

Ideas para extender este bot

Este mínimo bot se puede extender de varias maneras. Se me ocurren dos:

- Utilizar el procesamiento de lenguaje LUIS de Cognitive Services para reconocer entidades y órdenes a partir del lenguaje natural.
- En vez de utilizar las credenciales de la aplicación, abrir



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.



Activación de las opciones de guardado de plantilla en SharePoint Online

En SharePoint desde la versión 2007 existe la posibilidad de guardar Sitios, listas o bibliotecas de documentos como plantillas para su posterior reutilización.

Esto nos permite crear rápidamente nuevos sitios o listas con la misma estructura e incluso el mismo contenido que el sitio original que se guardó como plantilla y es una buena opción para crear rápidamente funcionalidad nueva con la misma estructura que la existente lo que nos permite guardar la homogeneidad.

En SharePoint Online, sin embargo, os podéis encontrar con que estas opciones no estén disponibles por defecto.

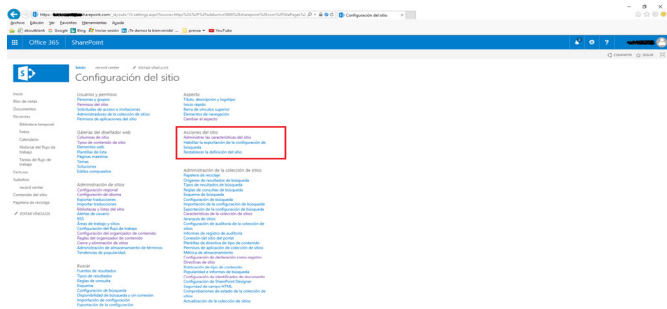


Imagen 1.- Ejemplo de sitio de SharePoint Online en el que no aparece la opción de guardar sitio como plantilla.

Para conseguir que la opción de guardar como plantilla aparezca hay que realizar los siguientes pasos:

- En la colección de sitios donde queremos activar el guardado como plantilla ir a Configuración de sitio.
- En la sección administración de la colección de sitios ir a administrar las características del sitio.

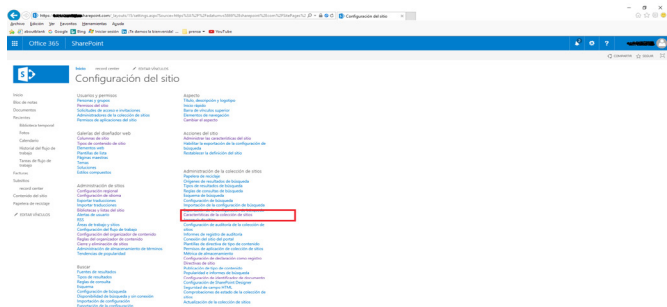


Imagen 2.- Acceso a la sección de Características de la colección de sitios.

- Ir a Características de la colección de sitios y desactivar la característica: Infraestructura de publicación de SharePoint Server. Si nuestro sitio debe utilizar esta característica, no podremos guardarlo como plantilla

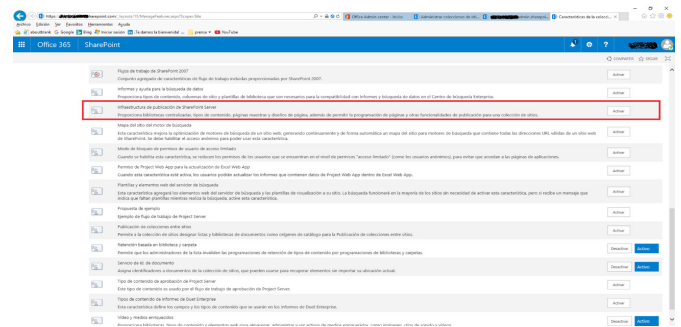


Imagen 3.- Característica de Infraestructura de publicación de SharePoint Server.

- A continuación, ir a Configuración de sitio. En la sección Acciones de sitio ir a Administrar las características del sitio.

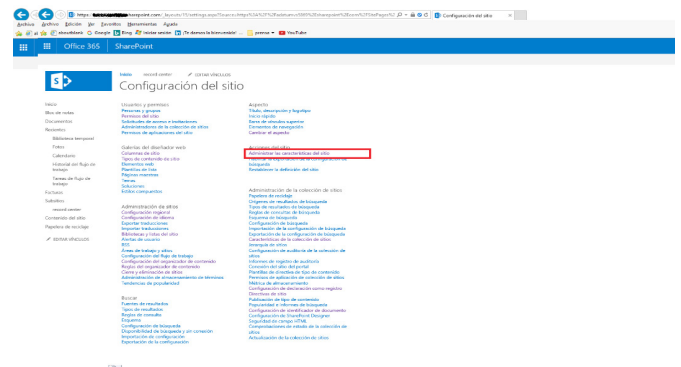


Imagen 4.- Acceso a Administrar las características del sitio.

- Desactivar si están activadas las características:
 - Característica de sitio de la comunidad.
 - Publicación de SharePoint Server.



Imagen 5.- Características a desactivar.

- De nuevo, si nuestro sitio necesita estas características no podremos guardarlo como plantilla.
- A continuación, abrir la web de administración central de SharePoint Online (<https://midominio-admin.sharepoint.com>), ir a Configuración.

“Para conseguir que la opción de guardar como plantilla aparezca en SharePoint 365 hay que realizar varios pasos”



Imagen 6.- Acceso a la administración central de SharePoint Online.

- Dentro de todas las configuraciones globales a nuestro tenant, buscar la sección Script personalizado que contiene dos opciones deshabilitadas por defecto:
 - Permitir que los usuarios ejecuten scripts personalizados en sitios personales.
 - Permitir que los usuarios ejecuten scripts personalizados en sitios creados por los mismos usuarios.
- Activamos las dos opciones. **IMPORTANTE:** Esta configuración tarda 24 horas en ser efectiva.

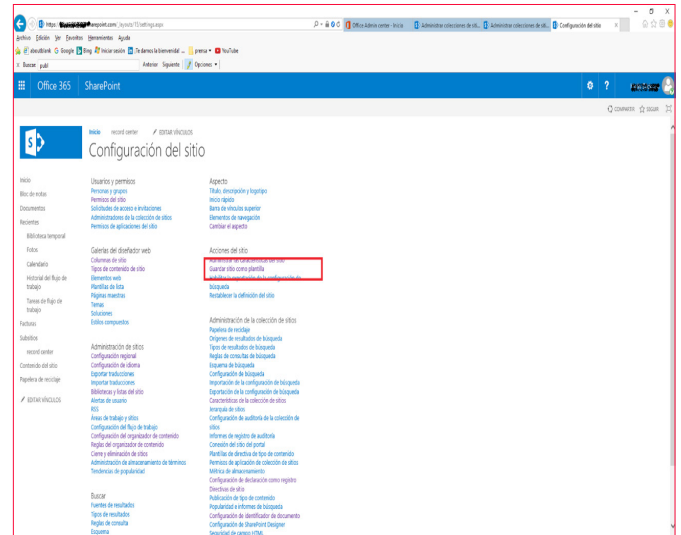


Imagen 8.- Opción de "Guardar sitio como plantilla" disponible.

- Y en la configuración de una lista o biblioteca:

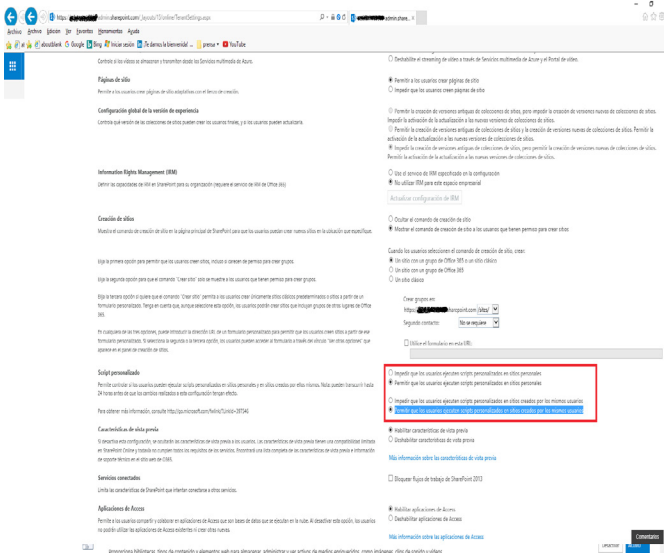


Imagen 7.- Opciones a activar en la administración de SharePoint Online.

- Después de seguir estos pasos y esperar las 24 horas, tendremos disponibles las opciones en la configuración de sitio.

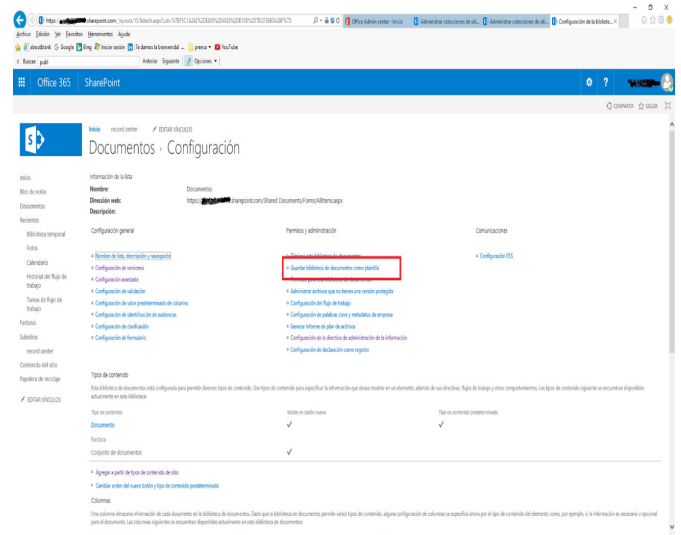


Imagen 9.- Opción de guardar biblioteca como plantilla disponible.

ALBERTO CARLOS ESCOLA FIZ
 Consultor IT freelance
 aescola@alescon.es

Enterprise Mobility Suite Parte 2 – Azure AD Premium

Continuando con el artículo anterior (EMS Parte 1) en esta nueva entrega hablaremos sobre el servicio AAD Premium que 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.

Como se comentó el post anterior, 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 para nuestra integridad de la visión. Microsoft ha trabajado con Gartner para hacer copias gratuitas disponibles que se pueden acceder aquí .

“...AAD Premium que 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...”



Imagen 1.- Cuadrante mágico de Gartner de Gestión de Identidad y Acceso como Servicio.

En el siguiente cuadro vemos las diferencias de las versiones de Azure AD y las características que diferencian de cada plan, incluyendo el servicio de Office 365.

Azure Active Directory editions GA feature comparison - Office 365 IAM features		Azure Active Directory Free	Azure Active Directory Basic	Azure Active Directory Premium	Office 365 apps only
Common features	Directory as a service	500,000 object limit	No object limit	No object limit	No object limit for Office 365 user accounts
	User/group management (add/update/delete)/user-based provisioning, device registration, User-based access management/provisioning, Basic Security/usage reports	Yes	Yes	Yes	Yes
	Single Sign-On	10 apps per user (pre-integrated SaaS and developer-integrated apps)	10 apps per user (pre-integrated SaaS and developer-integrated apps)	No link SSO, Basic SSO, Self-Service App Integration (templates 1)	10 apps per user (pre-integrated SaaS and developer-integrated apps)
	Self-service password change for cloud users	Yes	Yes	Yes	Yes
	Connect (sync engine that extends on-premises directory to Azure Active Directory)	Yes	Yes	Yes	Yes
Premium & basic features	Group-based access management/provisioning - Provisioning customization	Yes	Yes	Yes	Yes
	Self-service password reset for cloud users	Yes	Yes	Yes	Yes
	Company branding (login page/access panel customization)	Yes	Yes	Yes	Yes
	Application Proxy	Yes	Yes	Yes	Yes
	SLA	Yes	Yes	Yes	Yes
	Self-Service Group and app Management/Self-Service application addition/ Dynamic Groups			P1/P2	
	Self-service password reset/change/account unlock with on-premises write-back			P1/P2	
	Advanced usage reporting			P1/P2	
	Multi factor authentication (cloud and on-premises (MFA server))			P1/P2	Limited cloud only for Office 365 apps
	MM (CAL + MIM) owner			P1/P2	
Premium features	Cloud app discovery			P1/P2	
	Automated password rollover			P1/P2	
	Connect Health			P1/P2	
	Conditional Access (User, Application, Location, Device rules)			P1/P2	
	Identity Protection			P2	
	Privileged Identity Management			P2	
Azure Active Directory Premium - Windows 10 only related features					
Join a device to Azure AD Desktop SSO, Microsoft Passport for Azure AD, Administrator (BitLocker) recovery		Yes	Yes	Yes	Yes
ADFS synchronization, Self-Service Password recovery, Additional local administrators to Windows 10 devices via Azure AD Join, Enterprise Data Reporting				Yes	

Imagen 2.- Diferencias entre las distintas versiones de Azure AD.

Este servicio incluye entre otras características:

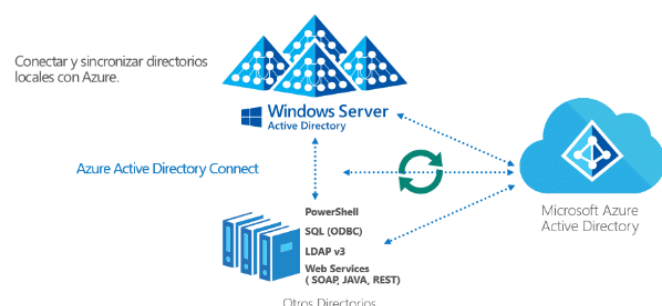


Imagen 3.- Arquitectura de SSO con Azure AD Connect (Más información en el siguiente Post: Realizar la configuración del AD Connet en su organización).

- Servicios Sincronización Cloud (App's):

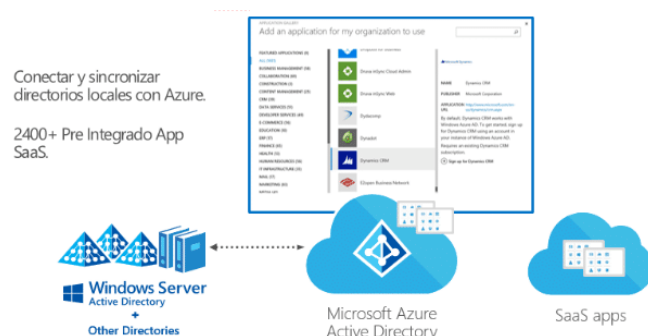


Imagen 4.- Servicios de sincronización de Apps.

- AAD Protección de la identidad: Es una característica de la edición Azure AD Premium P2 que proporciona una vista consolidada en los eventos de riesgo y vulne-

habilidades potenciales que afectan a las identidades de su organización. Protección de la identidad aprovecha las capacidades de detección de anomalías de AD Azure (disponible a través de informes de actividad anómala de AD Azure) existente, e introduce nuevos tipos de eventos de riesgo que pueden detectar anomalías en tiempo real.

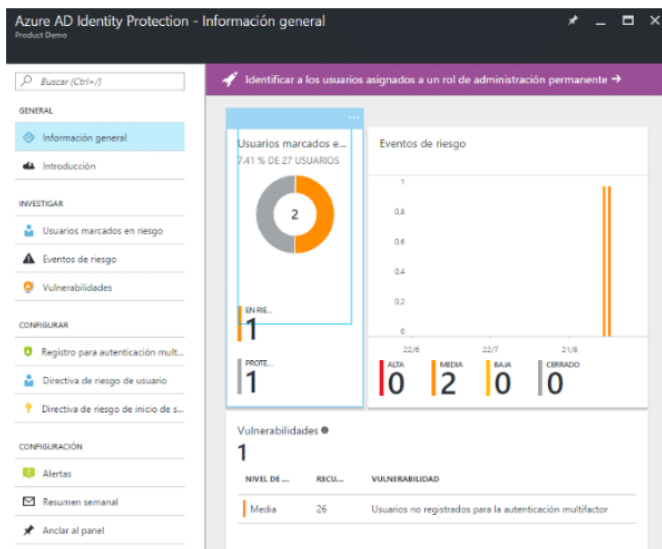


Imagen 5.- Azure AD Identity Protection.

- **Autoservicio:** Ofrece un portal de autoservicio a través del cual los propios usuarios que olviden una contraseña o bloqueen una cuenta, pueden restablecer sus contraseñas a partir de una pregunta de seguridad o un código enviado a su dispositivo móvil.
- **AAD Gestión de Identidad privilegiado:** Puede gestionar, controlar y supervisar el acceso dentro de su organización. Esto incluye el acceso a los recursos de Azure y otros servicios en línea como Office 365 o Microsoft Intune. Minimizar el número de personas que tienen acceso a información segura o recursos, debido a que reduce la posibilidad de que un usuario malintencionado conseguir que el acceso. Este componente puede ayudar a:
 - Ver qué usuarios son administradores de AD Azure.
 - Obtener informes sobre el historial de acceso de administrador y los cambios en las asignaciones de administrador.
 - Recibir alertas sobre el acceso a un papel privilegiado

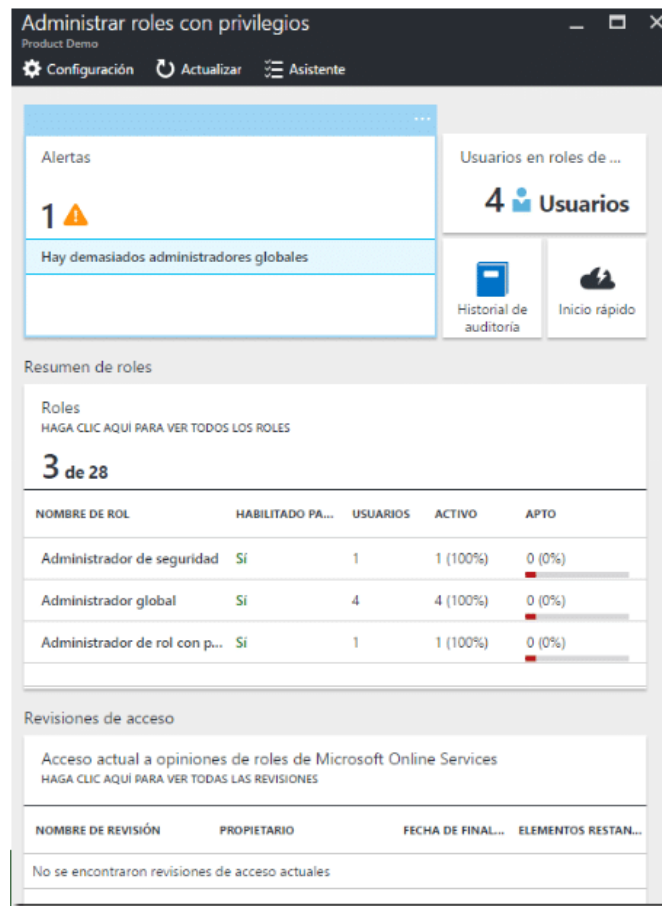


Imagen 6.- Administración de roles con privilegios.

- **Autenticación Multifactor (MFA):** Integra una funcionalidad que te ayuda a prevenir el acceso no autorizado a tus datos en la nube a través de una segunda autenticación mediante una App móvil, una llamada telefónica automatizada o un mensaje de texto.

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 Server & Services | Enterprise Mobility

jorge.castaneda@outlook.com

@xorxe

<http://jorgecastaneda.com>



Alberto Díaz

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

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

Email: adiazcan@hotmail.com

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

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



Fabián Imaz

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

Sitio Web: <http://www.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 la arquitectura, diseño e implementación de sistemas de IT basados en tecnologías de Microsoft, especialmente SharePoint, Office 365 y Azure. Propietario del sitio especializado en información sobre SharePoint en español <http://www.gavd.net>, autor de ocho libros sobre SharePoint y sus tecnologías y numerosos artículos y conferencias sobre el tema.

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/>

Coordinadores de sección

DAVID RIVERA FERNÁNDEZ

Coordinador de System Center
driverafer@hotmail.com

GEOVANY ACEVEDO

Coordinador de Exchange Server
geovany.acevedo@hotmail.com

GASTÓN CRUZ

Coordinador de PowerBi
gastoncruz@gmail.com

XAVIER MORERA

Coordinador de .Net
xavier@familiaromera.com

PABLO PERALTA

Coordinador de Dynamics CRM
pablop2006@gmail.com

JOHN BARRETO

Coordinador de System Center
john.barreto22@hotmail.com

RODOLFO CASTRO AGUILAR

Coordinador de Skype for Business
rodolfo-castro@live.com

ESTEBAN SOLANO GRANADOS

Coordinador de Xamarin
stfansolano@outlook.com

MAXI ACCOTTO

Coordinador de SQL Server
maxi.accotto@gmail.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

