

# COMPARTIMOSS

Revista especializada en Tecnologías SharePoint

INTRODUCCIÓN A MACHINE LEARNING, OFFICE GRAPH, DELVE Y CLUTTER

REST, WEBAPI 2 Y SHAREPOINT 2013 - INTRODUCCIÓN

CÓMO UTILIZAR LAS API'S DE OFFICE365 EN APLICACIONES MÓVILES MULTIPLATAFORMA CON XAMARIN

INTRODUCCIÓN A LOS GRUPOS DE OFFICE365 - TRABAJA COMO UNA RED (Parte I)



02

# COMPARTIMOSS

Revista especializada en Tecnologías SharePoint

## Staff

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

### DIRECCIÓN GENEAL

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

## Contacte con nosotros

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

### BLOGS

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

### REDES SOCIALES

#### Facebook:

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

#### LinkedIn:

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

#### Twitter:

@CompartiMOSScom

## Contenidos

03

### Editorial

Parece que tenemos SharePoint para rato, y hablamos tanto de la versión On-Premises como de la versión Online.

04. Introducción a Machine Learning, Office Graph, Delve y Clutter

07. Esquema de Sitios y Permisos

10. Entrevista David Sánchez Aguilar

12

### Aplicaciones High Trust (Parte I)

15. REST, WebAPI 2 y SharePoint 2013 - Introducción

20. Movilidad sobre SharePoint - Parte I

24. Como utilizar las API's de Office 365 en aplicaciones móviles multiplataformas desarrolladas con Xamarin

28

### Forefront Identity Manager y la sincronización de usuarios en SharePoint 2010

31. Replicación de granjas de SharePoint vía LogShipping

34- Herramientas de Branding para desarrollar aplicaciones de SharePoint elegantes y potentes

37. Introducción a los Grupos de Office 365 – Trabaja como una red (Parte I)

40. La nube de SharePoint On-Premises Nintex



Con 2014 a punto de concluir, desde CompartiMOSS nos complace presentarles una nueva edición (y ya son 22) de la revista de referencia en castellano sobre SharePoint y Office 365. En este nuevo número, contamos de nuevo con excelentes artículos en los que no sólo se abordan los componentes más técnicos de SharePoint y Office 365, sino también las enormes posibilidades que ambas plataformas brindan a organizaciones de todo tipo como demuestra la adición continua de nuevos servicios, capacidades y APIs a Office 365, tales como Delve, Grupos de Office 365 o el reciente Office 365 Video. Junto con estos nuevos servicios, cada vez está más claro que los escenarios y arquitecturas híbridas OnPremises-Online son una alternativa muy atractiva para organizaciones en las que una transición rápida a la nube es posible, aunque necesiten cierto tiempo de adaptación.

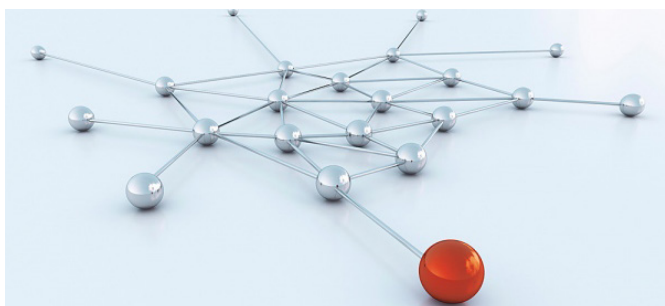
De nuevo, nuestro agradecimiento tanto a los autores que habitualmente colaboran con nosotros como a toda la base de lectores que impulsan CompartiMOSS y que hacen posible que número a número la revista siga creciendo y reinventándose. Y no queríamos despedirnos sin desear a todos nuestros autores y lectores una Feliz Navidad y un Feliz Año nuevo. Nos vemos en 2015 con nuevos números llenos de contenido y de calidad.

**EL EQUIPO EDITORIAL DE COMPARTIMOSS**

# Introducción a Machine Learning, Office Graph, Delve y Clutter

Como muchos de vosotros seguramente ya habréis visto u oído, Microsoft ha lanzado hace un par de meses algo llamado Office Delve, disponible inicialmente en Office 365 (planes “E”), “cosa” muy relacionada con algo llamado Office Graph y Machine Learning. En la misma línea unos días antes de liberar este artículo se ha anunciado Office Clutter como herramienta que también utiliza la misma tecnología base. Este artículo va dirigido a aquellos que todavía no habéis tomado contacto con estas herramientas o queréis saber de qué va esto de Graph, Delve y Clutter.

## Machine Learning

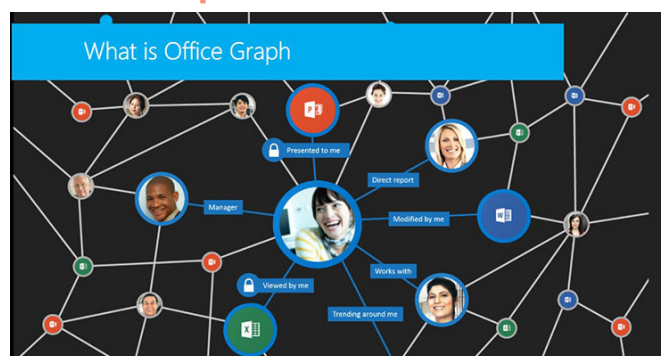


Antes de hablar de Delve debemos hablar de Graph, y antes de hablar de Graph debemos hablar de Machine Learning, pero sobre este último podríamos escribir páginas y páginas sin acercarnos todavía a describir su capacidad y potencial. Entrando en materia podemos definir Machine Learning como una rama de la inteligencia artificial donde el objetivo principal es aprender y analizar señales pasadas con el objetivo de predecir las futuras. Esto que ha quedado tan bonito escrito y tan poco claro a nivel conceptual no es más que (y nada menos) un conjunto de algoritmos destinados a buscar los patrones por los cuales una información origen se transforma en una información destino, y quiero recalcar que lo importante no es la transformación en sí misma sino la búsqueda del patrón que lleva a que un origen se transforme o catalogue en un destino de forma predecible.

Aunque a simple vista esta tarea harto complicada podría tener un beneficio poco tangible, en realidad abre las puertas a un beneficio mucho mayor llamado “Generalización”, que no es otra cosa que aplicar los patrones obtenidos en un “pequeño” conjunto de información e intentar obtener esos patrones de cambio en la información, con el objetivo de predecir cómo se transformará o agrupará en los futuros conjuntos. Existen muchos ejemplos en los que podríamos aplicar esta lógica de autoaprendizaje del

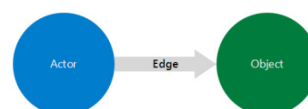
software, pero imaginaros el potencial que puede tener en cosas tan variopintas como predecir resultados de las elecciones basándonos en encuestas pasadas y los resultados que realmente se obtuvieron (de ahí obtendríamos nuestro patrón de información origen e información destino) o los patrones por los cuales un conjunto de células se transforman de “buenas” en “malas”. Si os interesa este tema en particular tenéis más detalle y ejemplo adicional en una publicación reciente en mi Blog: <http://blogs.encamina.com/everyware/2014/11/07/un-poco-de-machine-learning-o-la-imitacion-binaria-de-el-agua-moja/>

## Office Graph



Siguiendo con lo nuestro y ahora que tenemos una noción superficial de lo que es Machine Learning, es más fácil entender Office Graph, ya que Graph es la implementación de Microsoft de Machine Learning. Esta implementación, disponible por ahora sólo en Office 365, utiliza los algoritmos base definidos en Machine Learning con el objetivo de buscar la relación entre 3 grandes entidades como son Personas, Actividades y Contenidos. La forma en que esta información se “une”, aunque ya sabemos que lo correcto sería hablar sobre el patrón que vincula unas entidades con otras, es mediante la búsqueda y aplicación de señales entre Outlook, OneDrive, SharePoint y Yammer. Por ahora solo actúa sobre estos cuatro orígenes, pero cabe esperar la integración con Lync, Skype for Business, OneNote, los adjuntos del correo y el contenido de los enlaces publicados en todas las herramientas mencionadas anteriormente.

## En las tripas de Graph





Seguramente una de las características más importantes de Graph es que permite realizar consultas mediante GQL (Graph Query Language) que está aún en fase beta. Está diseñado para realizar las consultas al modelo de Graph vía el API REST de las búsquedas de SharePoint Online y funciona devolviendo los elementos de una entidad que cumplen con una condición en particular.

Graph descubre el vínculo entre las entidades mediante la conjunción de 3 elementos que son el Actor (quién o “Actor”), la Acción (qué o “Edge”) y el Objeto (dónde u “Object”). De esta forma podríamos obtener relaciones simples del tipo “Javier ha modificado Carta.docx” (actor, acción y objeto), u otras más complejas como “Alberto está trabajando con Artículo\_CompartiMOSS.pptx” (en este caso la acción no es simple sino que intrínsecamente implica varias iteraciones).

La sintaxis de estas consultas se construye como:

```
https://<tenant_address>/_api/search/query?Querytext='*'
&Properties='GraphQuery:actor(21894957\,action\.:1033)'
&SelectProperties='DocId,Title'
```

Donde QueryText es la entidad donde quieres realizar la búsqueda y es obligatoria, GraphQuery es el filtro que quieres aplicar (la acción que quieres obtener) y es opcional, y la última parte corresponde a los datos que quieres que te retorne la consulta.

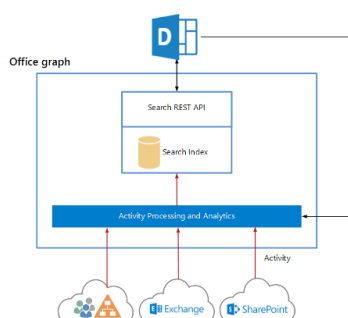
Por ejemplo, si quiero obtener todos los elementos (\*) que han sido vistos o modificados por mí esta sería la llamada (Modificado = acción 1003 y Visto = acción 1001):

```
https://<direccion_tenant>/_api/search/query?Querytext='*'
&Properties='GraphQuery:ACTOR(ME\,OR(action\.:1001\,action\.:1003))'
```

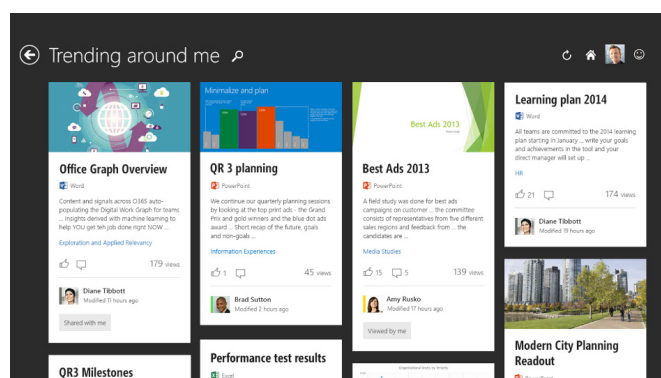
No quiero extender estas líneas con cientos de ejemplos de código pero sería pecado no dejaros este enlace donde tenéis toda la sintaxis y multitud de ejemplos para empezar a jugar.

## Office Delve

Si has llegado hasta aquí, imagino que ya sospechas lo que es Office Delve: es la primera herramienta que brinda al usuario información vinculada o relacionada con sus intereses sin necesidad de estar buceando en los repositorios donde yace esa información dentro de 365 (Exchange Online, OneDrive, SharePoint y Yammer). Para ello realiza consultas a Office Graph y otorga un interfaz más amigable y el posible “rankeado” de los resultados obtenidos.



Esto quiere decir que por ejemplo, para preparar una reunión sobre Delve que tenemos la semana próxima con los colegas y también autores de esta revista, Alberto Diaz y Adrián Diaz, Delve nos ahorra tiempo en lo que se refiere a la búsqueda de material que necesitamos ya que devolverá contenidos referentes a algunas presentaciones que ya comparto internamente, mis post en Yammer sobre esta materia y algunos documentos que ya he compartido internamente. Y lo mejor de todo esto es que estos colegas no han tenido que hacer nada para que Delve les recomiende esta información, sino que ahí está la “inteligencia” de Graph buscando vínculos entre compañeros de trabajo y vínculos entre información que trata del mismo asunto.

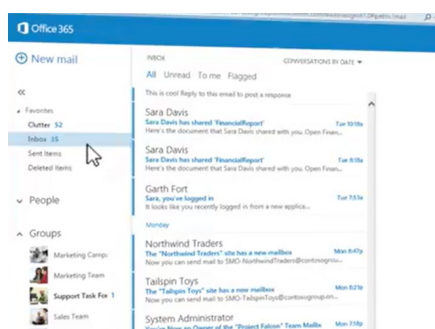


Otra cosa importante que provee Delve es que nos permite utilizarlo como un motor de búsqueda estándar pero agregando una capa donde el usuario es el centro y la frontera de la información al mismo tiempo. Con esto quiero decir que podría realizar una búsqueda sobre un tema en particular y agregar refinamientos (al modo SharePoint) del tipo “que me haya gustado” (Likes en Yammer) o que “son tendencia” (en base a vistas y votaciones en SharePoint y Yammer).

## Office Clutter

Anunciada su liberación por Microsoft este mismo mes, Office Clutter es la implementación de Graph para la Bandeja de Entrada (léase Outlook Online). Su objetivo es “limpiar” o mejor dicho, centrar la atención del usuario en lo realmente importante que llega por correo. Lo más significativo que debemos tener en cuenta es que el mismo tipo de algoritmos de Machine Learning que hemos comentado hasta ahora pueden ser también utilizados para centrar la atención en la bandeja de entrada. Básicamente Clutter mueve elementos no importantes de nuestra Bandeja de Entrada hacia un cajón desastre (Clutter Folder) con el objetivo de que nos centremos en lo realmente importante. Todos tranquilos! Clutter viene desactivado por defecto. Su funcionamiento se basa en valorar nuestro comportamiento con determinado correo y catalogarlo en importante o no según el asunto, cómo vienen los mensajes dirigidos hacia nosotros (To, CC, etc.), la temática, etc. De más está decir que Clutter funcionará mejor cuanto más lo utilicemos (esto es un corolario en Machine Learning) y Microsoft nos garantiza que esta información no será

jamás compartida ya que es privada y personalizada para cada individuo. Una vez que el correo ha pasado al Clutter Folder, sea porque nosotros mismos lo hemos movido ahí o porque Clutter lo ha catalogado así, seguimos teniendo control total sobre ese correo por lo que podemos marcarlo como importante, moverlo, en fin, realizar cualquier acción como con cualquier otro mensaje. Por último mencionar que Clutter también funciona con los clientes de escritorio de Outlook pero su activación/desactivación se realiza por OWA (por ahora) y para los escépticos deciros que esto se puede realizar en cualquier momento. Para aquellos que no pueden esperar y quieren ver esto en funcionamiento ya mismo aquí tenéis un vídeo que lo resume: <https://www.youtube.com/watch?v=uML5km-H5DYE>



## El futuro de Delve y Clutter

Aunque parece que para el caso de Graph el futuro se vislumbra positivo y el espectro de aplicación es inmenso, podríamos decir que no está tan claro para el caso de Delve y Clutter. A mi entender existen dos grandes barreras que ambos deben tumbar y que podemos identificar claramente, aunque muy a mi pesar son contrapuestas:

- La primera es la necesidad de ampliar las fronteras de Delve hacia todo el resto de 365 e incluso hacia algunos servicios de Azure y el resto del mundo. Cuanto mayor sea el rango de acción Delve (y por ende también aplica para Graph) mejor será la capacidad y acierto que tendrá en sus resultados. Parece muy obvia la necesidad de que su alcance incluya contenidos de Lync, CRM Online, PowerBI, o los resultados provenientes de otros servicios afines como Azure SQL Databases, WebSites o Media Services por citar algunos. Lo que no parece tan obvio, y afortunadamente es público el cambio de visión de Microsoft en este aspecto, es incluir resultados externos a la nube de Microsoft, como podrían ser los foros de opinión que cada uno preferimos, las webs de noticias preferidas, los resultados de búsquedas en otros motores, etc. Desde mi modesto punto de vista creo que la solución está en seguir invirtiendo y ampliando el rango de acción por parte de Microsoft por un lado, y la apertura de mente hacia el resto de plataformas y tecnologías por otro, cosas en las que no cabe duda Microsoft está dando pasos muy específicos y positivos en los últimos tiempos.
- La segunda barrera y que probablemente es más

psicológica que otra cosa, está del lado del usuario, ya que en algún momento podría interpretar que Delve y/o Clutter están atropellando los límites de su privacidad en la búsqueda de las relaciones entre la información, o en la catalogación de la misma. Estoy totalmente de acuerdo en que nos metemos en un terreno pantanoso donde la lógica y el sentido común se enfrentan al modo en que cada uno comparte información y la visibilidad u objetivo con el que se ha compartido esa información. Sin ánimo de buscar el debate en este aspecto (seguramente terminaríamos en el punto de partida) debemos ser conscientes y saber que aquí nada sucede de forma mágica, por lo que la información sobre la que se apoya Graph es información a la que de un modo u otro ha podido acceder y que cualquier persona con el tiempo y la paciencia necesarias hubiera sido capaz de vincular e hilar. Por este motivo creo que no debería ser este tema una barrera para el desarrollo y aceptación de Delve y Clutter, pero es verdad que la línea que separa lo que consideramos privado y público es muy fina y cambia de forma muy sutil en cada individuo, patrones que serán todo un reto de descubrir por parte de Graph.

## Para terminar...

Todos hemos visto inventos que vienen y van, algunos caen por el camino y otros llegan para quedarse. El Machine Learning lleva más de 30 años entre nosotros y sigue insistiendo y tomando distintas formas y nombres pero su objetivo es el mismo: hacer que las máquinas “piensen” en pro de hacernos la vida más fácil. Más que pensar que esto es una demostración fehaciente de vagancia humana deberíamos verlo como parte de la evolución, y si alguien no lo ve así está en todo su derecho, pero no olvides a aquellos que dijeron que “esto de los cajeros automáticos va a fracasar porque todo el mundo quiere sacar su dinero del banco interactuando con un cajero físico”, o “es imposible controlar a distancia un robot en la superficie de Marte”, o aquellos otros que dijeron que “una máquina nunca será capaz de construir un coche”.

Otro tema es que tardemos más o menos en utilizar popularmente sus ventajas, lo que está directamente relacionado con lo inteligente y útil que sea su aplicación, pero todo parece indicar que iremos a más en este aspecto. Desde mi humilde opinión no veo tan lejano recibir información de mis colegas y comparativas en tiempo real sobre el phablet de última generación que estuve viendo esta mañana en un escaparate a través de mis Google Glasses, y si esa información perdura y tiene importancia solo y hasta que haga la compra o la desecha, estará todo dicho.

**JAVIER MENÉNDEZ PALLO**

Director Oficina Centro – España y Portugal

ENCAMINA

[jmenendez@encamina.com](mailto:jmenendez@encamina.com)

# Esquema de Sitios y Permisos

Cuando ingresa a SharePoint (intranet, portal, extranet, etc...) a su sitio de trabajo habitual, está ingresando a la colección de sitios que lo contiene, muchas veces navega por el sitio principal de la colección, todas las colecciones de sitios deben tener un sitio principal, en la mayoría de ocasiones es llamado Home, Casa, Inicio, el nombre de la compañía, etc... Este sitio de nivel superior es la raíz de la colección de sitios.

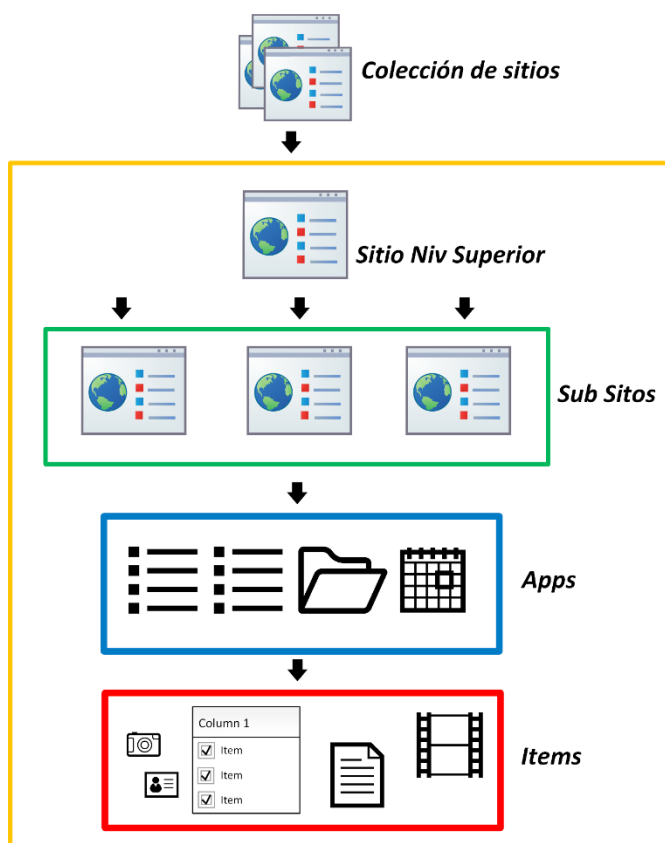


Imagen 1.- Esquema de sitios.

## Importante que se tenga en cuenta

- Herencia de permisos: cuando cualquier elemento de la imagen 1 (Sitio1, subsitio, lista, ítem...) está Heredando permisos, esto significa que obtiene los permisos del elemento inmediatamente arriba de él.
- Permisos Exclusivos: cuando el elemento tiene permisos exclusivos, esto significa que el elemento maneja de forma independiente su esquema de seguridad.

¿Qué sucede cuando de heredar permisos?

Cambia el esquema de permisos del elemento a permisos exclusivos, haciendo una réplica inicial de los permisos el elemento al que estaba heredando.

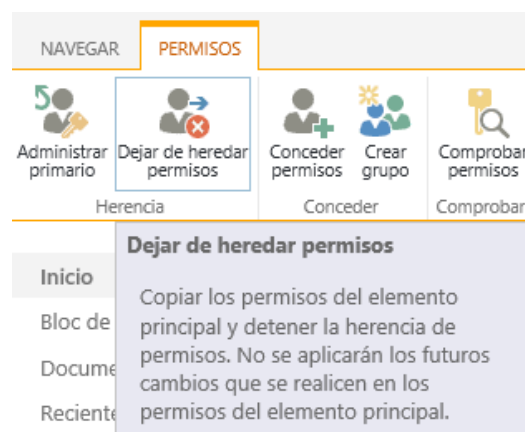


Imagen 2.- Dejar de heredar permisos.

¿Qué sucede cuando tengo un elemento con permisos exclusivos y vuelve a heredar permisos? Este elemento pierde todo el esquema de permisos que tenía y adopta el esquema de permisos del elemento inmediatamente superior a él.

El estándar de SharePoint dicta que al crear un nuevo sitio, este tiene dos opciones, la primera es heredar los permisos, de lo contrario, crear permisos exclusivos para este sitio (inicialmente nos propone que creamos 3 grupos de seguridad para manejar los permisos "Visitantes, integrante y dueños"). A la hora de crear una nueva lista, biblioteca o un nuevo ítem, documento, evento..., estos heredan los permisos predeterminadamente.

Haciendo referencia la "imagen 1" en la parte superior tenemos el sitio de primer nivel, por ser el sitio de nivel superior en el encontrara una sección en la configuración del sitio llamada "Administración de la colección de Sitios" y en la sección "Usuarios y permisos" se encuentra la opción de agregar los administradores de la colección de sitios (únicamente se encuentra en el sitio de primer nivel).

Usuarios y permisos

Personas y grupos

Permisos del sitio

Administradores de la colección de sitios

Permisos de aplicaciones del sitio

Imagen 3.- Agregar Administradores de la colección de sitios.



## Administración de permisos en sitios y apps

Los permisos de los demás elementos se manejan de la misma forma, ingresando a la administración de permisos del elemento, verifique si está heredando permisos y según su requerimiento maneje los permisos heredados o desherede el elemento y maneje permisos exclusivos.

- Para administrar los permisos del sitio, valla a Configuración del sitio -> Usuarios y permisos - Permisos del sitio (puede navegar hasta allí si es propietario del sitio).

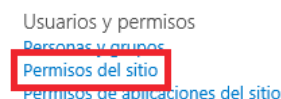


Imagen 4.- Administrar permisos de Sitio.

- Para administrar los permisos de la lista, valla a Configuración de la lista -> Usuarios y permisos - Permisos para esta lista (puede navegar hasta allí si es propietario de la lista)

Permisos y administración

- Eliminar esta lista
- Guardar lista como plantilla
- **Permisos para esta lista**
- Configuración del flujo de trabajo
- Generar informe de plan de archivos
- Configuración de palabras clave y metadatos de empresa
- Configuración de la directiva de administración de la información

Imagen 5.- Administrar permisos de Lista.

**Cuando la creación de sitios es constante y los sitios tienen un alto nivel colaborativo, el ingreso o asignación de permisos puede ser un cuello de botella, para evitar esto se convierte en un problema, SharePoint nos provee tres opciones de solicitud de acceso**

En la interfaz de administración de permisos del sitio se puede evidenciar si algún elemento tiene permisos exclusivos, y acceder a la administración de los permisos de ese elemento desde el vínculo "administrar Permisos" que están frente al nombre de cada elemento.

⚠ Parte del contenido de este sitio tiene permisos diferentes de lo que ve aquí. [Mostrar estos elementos.](#)

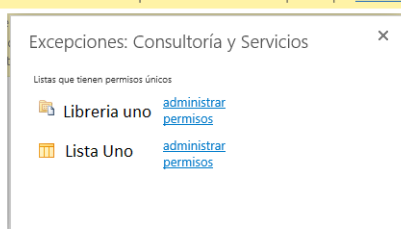


Imagen 6.- Administrar permisos diferentes.

## Configuración solicitud de acceso

layouts/15/AccessDenied.aspx?

Necesita permiso para tener acceso a este sitio.

Enviar solicitud

Imagen 7.- Solicitar permiso.

Opss... ¡estamos trabajando en ello!

Cuando la creación de sitios es constante y los sitios tienen un alto nivel colaborativo, el ingreso o asignación de permisos puede ser un cuello de botella, para evitar esto se convierte en un problema, SharePoint nos provee tres opciones de solicitud de acceso. (Puede seleccionar la opción que se ajuste mejor a su ambiente)

- La primera permite que los usuarios que tengan un nivel de permiso colaborativo puedan compartir archivos y carpetas, esto hace posible que usuarios que NO tengan permisos en el sitio accedan y colaboren en documentos o carpetas específicas.
- La segunda permite que los usuarios inviten a otros a pertenecer al grupo de integrantes del sitio y así obtener acceso colaborativo en el sitio.
- La tercera permite que los usuarios que ingresen al sitio y no tengan permisos, soliciten acceso exponiendo la razón por la cual es necesario su ingreso, todas las solicitudes serán enviadas al usuario que se defina por sitio, el cual podrá decidir si da o no acceso al sitio.

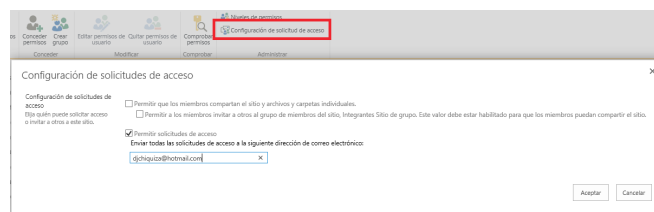


Imagen 8.- Configuración solicitud de acceso.

*Nota: Esta configuración será realizada únicamente por un usuario que sea propietario del sitio o tenga un nivel de permiso mayor a este, los sitios que están heredando permisos también heredan esta configuración.*

## Niveles de permisos

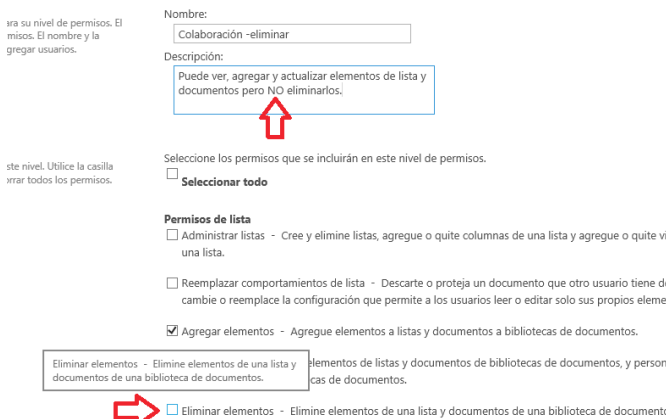
Los permisos indispensables que debe conocer son:

Nombre del permiso	Descripción
Control total	Tiene control total.
Diseño	Puede ver, agregar, actualizar, eliminar, aprobar y personalizar.

Editar	Puede agregar, editar y eliminar listas; puede ver, agregar, actualizar y eliminar elementos de lista y documentos.
Colaborar	Puede ver, agregar, actualizar y eliminar elementos de lista y documentos.
Leer	Puede ver páginas y elementos de lista, y descargar documentos.

Son los permisos más comunes, existen muchos más, en este sitio puede verlos todos. [Ver sitio](#)

No obstante los permisos existentes en ocasiones no cubren nuestros requerimientos, pero eso no es impedimento para nosotros, ¿Por qué?, porque podemos crear nuestros propios niveles de permisos, un ejemplo común es el siguiente: se requiere que en una biblioteca de documentos los usuarios puedan cargar documentos, editarlos, compartílos pero NO eliminarlos, para este requerimiento debemos crear un nuevo nivel de permiso.



ira su nivel de permisos. El misos. El nombre y la gregar usuarios.

Nombre: Colaboración -eliminar

Descripción: Puede ver, agregar y actualizar elementos de lista y documentos pero NO eliminarlos.

Seleccione los permisos que se incluirán en este nivel de permisos.

☐ Seleccionar todo

Permisos de lista

☐ Administrar listas - Cree y elimine listas, agregue o quite columnas de una lista y agregue o quite una lista.

☐ Reemplazar comportamientos de lista - Descarte o proteja un documento que otro usuario tiene o cambie o reemplace la configuración que permite a los usuarios leer o editar solo sus propios elementos.

☒ Agregar elementos - Agregue elementos a listas y documentos a bibliotecas de documentos.

Eliminar elementos - Elimine elementos de una lista y documentos de una biblioteca de documentos.

☐ Eliminar elementos - Elimine elementos de una lista y documentos de una biblioteca de documentos.

Imagen 9.- Configuración solicitud de acceso.

En este caso el nivel de permiso “Colaboración –eliminar” permite hacer lo mismo que el permiso “Colaborar” menos eliminar documentos, siendo así, podemos utilizar la opción copiar nivel de permisos y quitar el la opción de Eliminar elementos en la lista de acciones.

*Nota: para acceder la opción copiar nivel de permisos ingrese a Configuración del sitio -> Usuarios y permisos – Permisos del sitio -> Permisos – Niveles de permisos -> [seleccione el permiso a copiar].*

## Grupos de SharePoint

Los grupos nos permiten dejar de manejar los permisos individualmente (a nivel de usuario) y comenzar a manejar conjuntos de usuarios, lo que hace mucho más fácil la administración, cuando creamos grupos, lo más relevante es seleccionar que usuarios deben ir en cada grupo y que nivel de permisos tendrá este grupo en el sitio.

SharePoint nos provee unos grupos predeterminados con su nivel de permiso, grupos como: Visitantes, Integrantes y Propietarios (grupos nombrados anteriormente).

### Visitantes a este sitio

Los visitantes pueden leer el contenido del sitio web. Crear un grupo de visitantes o volver a utilizar un grupo de SharePoint existente.

Visitantes Sitio de grupo

### Integrantes de este sitio

Los miembros pueden aportar contenido al sitio web. Crear un grupo de miembros del sitio o volver a usar un grupo de SharePoint creado.

Integrantes Sitio de grupo

### Propietarios de este sitio

Los propietarios tienen control total del sitio web. Crear un grupo de propietarios o volver a utilizar un grupo de SharePoint existente.

Propietarios Sitio de grupo

Imagen 10.- Grupos de SharePoint.

Como pasa en los “niveles de permisos”, existen muchos grupos más, y tampoco es impedimento para nosotros si estos grupos no cumplen nuestros requerimientos, porque también podemos crear grupos si lo necesitamos, es más, si los grupos están creados en el AD de la organización podemos utilizarlos dentro de SharePoint y asignarles permisos cuando los asignamos a un sitio o App.

### FERNANDO CHQUIZA

Consultor SharePoint

[djchiquiza@hotmail.com](mailto:djchiquiza@hotmail.com)

[@fchiquiza](mailto:@fchiquiza)

<http://fernando-chiquiza.blogspot.com/>



10

## Entrevista David Sánchez Aguilar

Mi nombre es David Sánchez Aguilar, vivo en Tibás, San José de Costa Rica.

Tengo más de 7 años trabajando con tecnología Microsoft. Una de las razones por las cuales comencé a especializarme en las plataformas de esta gran empresa fue por ver la enorme cantidad de material disponible gratuitamente para todo el público, tanto por parte de Microsoft como de otros especialistas que comparten su conocimiento, fue algo que siempre me llamo la atención y en ese momento decidí que por pasión a la tecnología cuando tuviera alguna especialidad iba a compartir el conocimiento adquirido, colaborando, compartiendo y apoyando a otros profesionales en su aprendizaje.

Tuve la oportunidad de trabajar por primera vez con SharePoint desde que estaba en la universidad, en un proyecto sobre propuestas para Intranet. En aquel entonces me llamaba mucho la atención el desarrollo en .NET, las bases de datos, el manejo de servidores y realmente no sabía en qué área especializarme, sin embargo tenía claro que debía decidir un camino, con SharePoint encontré una platafor-



ma que abarcaba todas las áreas de mi interés y adicionalmente para implementar muchos tipos de soluciones, desde entonces cada día trato de aprender algo nuevo de SharePoint y compartirlo con las comunidades técnicas. Obtuve el reconocimiento de Microsoft Most Valuable Professional (MVP) en el año 2013 y tuve la oportunidad de recibir la renovación el pasado mes de Julio del 2014.

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

Desde que era un niño tuve la oportunidad de trabajar con una computadora y desde ese momento sentí una gran curiosidad por saber cómo funcionaban “esas máquinas”, cuando estaba en la primaria, ayudaba en un negocio familiar de venta de equipo informático a realizar las instalaciones de sistema operativo, mantenimiento y configuraciones básicas de los equipos que se vendían. Más adelante, en la secundaria ya tenía decidido que definitivamente la tecnología era lo que quería para mi carrera profesional y empecé a investigar más sobre la ingeniería en sistemas computacionales, tomaba cursos de programación en mis tiempos libres y siempre me ponía a investigar sobre cómo era el funcionamiento de los servidores. Creo que muchos de mis logros en mi carrera profesional se deben a las “ansias” de aprender, de mantenerme al tanto de las novedades e informado sobre las tecnologías con las que trabajo. Por eso me gustaría motivar a todas las personas que tengan ganas de aprender sobre algún tema a nunca

abandonar este ideal y siempre buscar la manera de ampliar nuestro conocimiento.

### ¿Cuáles son tus principales actividades tecnológicas hoy en día?

Me gusta mantenerme actualizado, cada mañana me mantengo al tanto sobre las novedades de los productos con los que trabajo, comparto los artículos de interés con las comunidades técnicas a las que pertenezco y también me gusta informarme y colaborar sobre los casos que se exponen ante los equipos de desarrollo de producto de SharePoint. En el área profesional me dedico a ser consultor sobre tecnologías Microsoft y entre mis actividades tecnológicas se encuentra analizar requerimientos, diseñar soluciones, programar, realizar pruebas de concepto (PoCs), llevar a cabo implementaciones, migraciones, entre otras actividades, generalmente un 90% sobre la plataforma de SharePoint el otro 10% sobre desarrollo de aplicaciones .NET



*Una de las razones por las cuales comencé a especializarme en las plataformas de esta gran empresa fue por ver la enorme cantidad de material disponible gratuitamente para todo el público, tanto por parte de Microsoft como de otros especialistas que comparten su conocimiento*

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

Pasar tiempo en familia, siempre separo de mi agenda un tiempo para compartir con mis seres queridos, ya sea en el hogar o planear una salida a cenar o al centro comercial. También en mis tiempos libres me gusta leer, conocer nuevos lugares y salir con mis amigos.

## ¿Cuáles son tus hobbies?

Me considero un gamer, me llaman mucho la atención los videojuegos. Me gusta ver series de televisión, como por ejemplo Gotham, Games of Thrones, etc. También ir al cine, leer sobre temas tanto de tecnología como de otros temas por ejemplo psicología, mi superhéroe favorito es Batman, así que me gusta leer comics o ver películas de este personaje. En cuanto a música, me gusta escuchar

rock, mi grupo favorito es Metallica.

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

Tanto Microsoft como otras compañías están apostando muy fuerte a temas de cloud, cada vez más se ven grandes avances en las plataformas en la nube y cada vez más podemos encontrar dispositivos innovadores que se conectan a internet, estamos en una gran etapa para la tecnología con el llamado “Internet de las Cosas” (IoT).

A un corto tiempo no considero que las plataformas “On-premise” vayan a desaparecer, lo que podremos ir viendo son implementaciones híbridas, pero a muchas corporaciones aún les falta alinear muchos detalles para poder llevar sus soluciones a la nube, esto por falta de actualización o restricciones a nivel de los datos, pero sin duda alguna es algo que se va a ir dando con el paso del tiempo.

A nivel de hardware es sorprendente los avances que se han llevado a cabo en los últimos años, cada vez más el procesamiento tiene mayor potencial, el almacenamiento es sumamente barato, a tal punto que en los últimos días Microsoft ofreció almacenamiento ilimitado para las personas con cuentas de Office 365.

---

### DAVID SÁNCHEZ AGUILAR

Microsoft MVP SharePoint Server

<http://davidsanchezaguilar.wordpress.com>

[davidsanchezaguilar@outlook.com](mailto:davidsanchezaguilar@outlook.com) @davidsancheza

# Aplicaciones High Trust (Parte I)

A estas alturas todos sabemos más o menos que es el modelo de aplicaciones en la nube (cloud app model) que trajo SharePoint 2013. Se trata de que el código de las aplicaciones de SharePoint ya no se ejecuta en el proceso de SharePoint (uno de los procesos w3wp.exe de IIS) sino en otra ubicación, fuera de SharePoint. De esta manera se asegura la estabilidad de la granja de SharePoint, ya que el código a medida está físicamente separado de SharePoint.

También sabemos que existen dos principales tipos de aplicaciones de SharePoint 2013: **SharePoint-hosted** y **provider-hosted**. Hasta hace poco existía una tercera solución, auto-hosted, pero nunca salió de la versión “Preview” y fue abandonada en junio de 2014. Las aplicaciones SharePoint-hosted solamente pueden contener código JavaScript, ya que se ejecutan en el navegador, mientras las aplicaciones provider-hosted son las verdaderas cloud apps de SharePoint 2013.

Sin embargo, es poco conocido que las aplicaciones de SharePoint 2013, y aquí me refiero a las provider-hosted, pueden tener **dos modalidades en cuanto a nivel de seguridad**. Una modalidad, la más conocida actualmente es aquella en la que la aplicación y SharePoint utilizan el servicio Access Control Service (ACS) en la nube. La gran mayoría de ejemplos de apps de SharePoint 2013 son ejemplos de esta modalidad, llamada low-trust o de baja confianza. Pero, en los entornos on-premise existe otra modalidad llamada high-trust o de alta confianza.

En un proyecto reciente me he visto metido con mucho detalle en la arquitectura de aplicaciones high-trust. Debido a que hay poco conocimiento de este tipo de aplicaciones, he decidido compartir con vosotros lo que he aprendido construyéndolas, ya que son muy adecuadas para hacer aplicaciones on-premise en entornos corporativos.

## Aplicaciones Low-Trust

En el modelo de aplicaciones low-trust de SharePoint, entre el servidor de SharePoint y el servidor donde está la aplicación existe un componente de Azure, llamado ACS, que hace de mediador para que SharePoint “se fíe” de las llamadas que hace nuestra aplicación hacia él. El flujo de la autorización queda explicado en el diagrama siguiente.

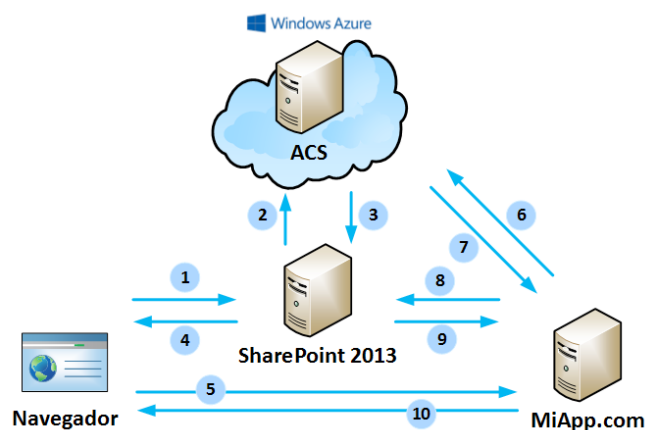


Imagen 1.- El proceso de autorización en una aplicación low-trust.

1. El usuario abre una página de SharePoint en la que está incrustado el WebPart de la aplicación MiApp.com (un IFRAME).
2. SharePoint detecta que hay que autorizar la aplicación y para ello necesita un token de contexto, firmado por ACS. Pide un token a ACS.
3. ACS firma el token con un secreto que solo conocen SharePoint y la MiApp.com y lo devuelve a SharePoint.
4. SharePoint usa el token para incluirlo en la petición que va a realizar dentro del IFRAME (la WebPart). Devuelve el HTML de la página al usuario.
5. Al tener un IFRAME, el navegador realiza la petición a la URL de MiApp.com. La URL incluye el **token de contexto**.
6. La aplicación recibe la petición HTTP. Extrae el token de la petición y comprueba la firma del token con el **secreto** que tiene la aplicación. De esta manera la aplicación sabe que la petición viene de SharePoint y es válida. Para acceder a SharePoint mediante CSOM o REST, la aplicación necesita un **token de acceso**. Dentro del token de contexto recibido, se encuentra un **token de refresco**. MiApp.com hace una petición a ACS pasándole el token de refresco.
7. ACS verifica que el token de refresco es válido para el origen de la petición. Si es así, le devuelve

un token de acceso a SharePoint, válido durante unas horas.

8. MiApp.com realiza una petición a SharePoint para obtener los datos a mostrar. Incluye el token de acceso en la petición.
9. SharePoint comprueba la validez del token de acceso y devuelve los datos que la aplicación le ha pedido.
10. La aplicación renderiza el HTML que se va a visualizar en el IFRAME y lo devuelve al navegador.

Lo más extraordinario de esta danza de peticiones entre el navegador, SharePoint, ACS y la aplicación es que es totalmente transparente para el usuario y que no requiere de ninguna configuración previa para que ACS funcione. Lo único que se necesita es que haya conectividad entre la aplicación, SharePoint y ACS que está en Internet y que haya confianza entre SharePoint y ACS.

Sin embargo, en entornos on-premise es frecuente que las máquinas dentro del datacenter no tengan conectividad hacia Internet, por lo que no podríamos utilizar ACS para la autorización de acceso a SharePoint. Además, lo que es peor, una instalación de SharePoint on-premise utiliza certificados autofirmados (self-signed) para firmar tokens, que no se aceptan en ACS. Configurar ACS para granjas de SharePoint on-premise es posible, pero implica demasiado trabajo (sustituir el certificado autofirmado por un certificado SSL, registrar el certificado en Azure Active Directory que tiene relación de confianza con ACS).

Sin embargo, disponemos de otra modalidad: high-trust apps.

## Aplicaciones High-Trust

Las aplicaciones High-Trust no usan intermediario para hablar con SharePoint, como ACS, sino que se identifican mediante certificados digitales con el sistema de clave pública y privada. SharePoint puede validar que un token proveniente de la aplicación sea válido porque dispone de la clave pública del certificado.

El flujo de la autorización high-trust o también llamada Server-to-Server (S2S) queda explicado en el diagrama siguiente.

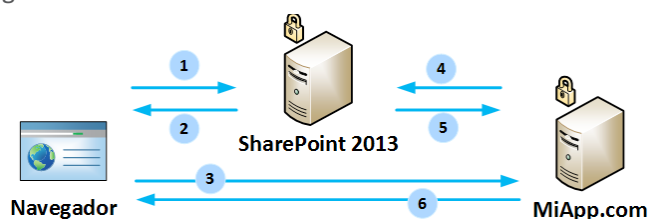


Imagen 2.- El proceso de autorización en una aplicación high-trust.

1. El usuario abre una página de SharePoint en la que está incrustado el WebPart de la aplicación MiApp.com (un IFRAME).

2. SharePoint devuelve el HTML de la página al usuario.
3. Al tener un IFRAME, el navegador realiza la petición a la URL de MiApp.com.
4. La aplicación recibe la petición HTTP. Para acceder a SharePoint mediante CSOM o REST, la aplicación necesita un token de acceso. La aplicación construye el token de acceso y lo firma con la clave privada de su certificado. Incluye este token en la petición a SharePoint.
5. SharePoint comprueba la validez del token de acceso con la clave pública del certificado de la aplicación y devuelve los datos que la aplicación le ha pedido.
6. La aplicación renderiza el HTML que se va a visualizar en el IFRAME y lo devuelve al navegador.

Comparado con el modelo low-trust, la autorización high-trust es más sencilla. La propia aplicación construye su propio token de acceso. Por esto mismo llamamos “high-trust” a este modelo: SharePoint “se fía” de lo que le dice la aplicación, sobre todo en la parte de los permisos del usuario. Mientras en una aplicación low-trust SharePoint prepara un token de contexto, que incluye los datos del usuario actual, una aplicación high-trust rellena la parte de usuario y SharePoint confía en que es correcta. Veremos ejemplos de esta confianza en los artículos siguientes de esta serie.

Sin embargo, alta confianza no quiere decir confianza total. La aplicación puede hacer como mucho lo que quien instala la aplicación le permite, igual que en las aplicaciones low-trust. Lo único que cambia es que la aplicación puede hablar directamente con SharePoint, de servidor a servidor, para acceder a datos y pasar tokens de acceso.

Para más detalles, cabe decir que tanto aplicaciones low-trust como high-trust implementan la autorización siguiendo el estándar OAuth, usando diferentes tipos de flujos de autorización de OAuth recogidos en el estándar.

**existen dos principales tipos de aplicaciones de SharePoint 2013: SharePoint-hosted y provider-hosted. Hasta hace poco existía una tercera solución, auto-hosted, pero nunca salió de la versión “Preview” y fue abandonada en junio de 2014**



## Conclusión

Hemos visto que disponemos de dos modelos de autorización de aplicaciones de SharePoint 2013: low-trust y high-trust. Low-trust utiliza un intermediario en Azure ACS y es muy adecuado para aplicaciones que se van a distribuir en la tienda de SharePoint o se van a instalar en Office 365. High-trust es más adecuado para aplicaciones corporativas ya que simplifica el flujo de autorización y permite mayor flexibilidad en las credenciales del usuario.

En los próximos artículos veremos cómo configurar una

aplicación high-trust en SharePoint 2013 on-premise y cómo podemos mezclar diferentes tipos de autenticación para SharePoint y para la aplicación.

EDIN KAPIC

Arquitecto SharePoint – MVP de SharePoint Server

[edin.kapic@spenta.es](mailto:edin.kapic@spenta.es)

@ekapic

<http://www.spenta.es>

## KWizCom Forms

### Formularios Nativos para SharePoint y Soluciones Móviles

**Cree fácilmente formularios personalizados utilizando los formularios de las listas existentes. Sin necesidad de despliegue ni aprender a utilizar nuevas herramientas!**



[www.kwizcom.com/forms](http://www.kwizcom.com/forms)

# REST, WebAPI 2 y SharePoint 2013 - Introducción

SharePoint siempre ha sido un sistema abierto a aceptar modificaciones por medio de programación y que expone todos sus componentes e información al mundo exterior por medio de un API (Application Programming Interface) muy poderoso. Ese es, probablemente, uno de los factores más importantes para su aceptación empresarial. Aunque SharePoint siempre ha tenido una API remota (basada en Servicios Web), es decir, que permite crear software que interactúa con el servidor sin que sea indispensable instalar los programas localmente, no fue hasta la versión 2010 cuando Microsoft le incluyó un API remoto basado en REST (Representational State Transfer).

REST, como se explica más adelante en detalle, es un protocolo abierto y público que permite intercambiar información entre dos sistemas informáticos remotamente. En los últimos años, REST se ha convertido en el estándar para intercambio de información por excelencia, siendo utilizado por todos los grandes fabricantes de software (Amazon, Google, etc.), incluyendo a Microsoft. REST, fuera de ser un protocolo práctico y fácil de utilizar, soluciona los problemas técnicos de rapidez de ejecución bajo carga y transparencia a través de Internet que otros protocolos de Servicios Web, como SOAP, WSCL y BEEP, presentan.

A su vez, Microsoft ha creado un Framework de programación para crear servicios REST llamado WebAPI, basado en ASP.NET e integrado en Visual Studio. De esta forma, es fácil construir servicios que accedan una gran cantidad de clientes, incluyendo navegadores y dispositivos móviles.

## REST

Representational State Transfer, REST, es el acrónimo de moda cuando se habla de conectar sistemas informáticos diversos de forma remota. REST es un protocolo abierto y público definido por el W3C (World Wide Web Consortium) Technical Architecture Group, la misma organización que define los protocolos de HTTP, HTML y CSS, entre otros.

Aunque en pocas palabras se puede definir a REST como un Framework para enviar comandos a un servidor por medio del URL y recibir respuesta en forma de XML, el protocolo es más complejo que solamente esto:

REST es Cliente-Servidor. Esto significa que el protocolo parte del principio de que la comunicación se realiza

entre dos sistemas, uno que solicita operaciones y otro que las provee, y que existe un contrato entre los dos que especifica en que formato tanto los comandos como los resultados serán enviados para que los dos sistemas se comprendan entre ellos.

REST no mantiene el estado (Stateless), es decir, cuando el servidor realiza la operación solicitada, “olvida” toda la información relacionada con el cliente. Esto implica que tanto la consulta como la respuesta tienen que contener toda la información relevante: no se permite, por ejemplo, realizar consultas escalonadas como una unidad, en las que una consulta se basa parcialmente en los resultados de la operación. En este caso, cada consulta se maneja independientemente, y las subsecuentes operaciones tienen que ser iniciadas por el cliente. También como una consecuencia de esta regla, REST no es transaccional de ninguna manera.

REST permite cachear la información enviada por el servidor. Todo servidor REST debe estar en capacidad de enviarles a los clientes información sobre si es posible cachear la respuesta, y por cuanto tiempo. Esto le permite a las aplicaciones cliente mantener una copia de la respuesta localmente evitando otras consultas, hasta que el tiempo de expiración (dado por el servidor) termina.

REST utiliza una interface de comunicación. Por definición, el protocolo que utiliza REST es HTTP(S) por medio de un URI (Uniform Resource Identifier). Popularmente se puede decir que REST utiliza un URL (Uniform Resource Locator) para enviar las consultas, pero eso es técnicamente incorrecto. Un URI es un subconjunto de un URL, pues solamente contiene información sobre el recurso a utilizar (“../contratos/1”) y no en donde encontrarlos (“http://servidor/contratos/1”). También, y como consecuencia de utilizar HTTP, es necesario utilizar solamente los verbos definidos por ese protocolo (GET, PUT, etc.) y no es posible definir verbos personalizados. Otra consecuencia de utilizar HTTP es que tanto las consultas como las respuestas tienen que ser creadas en XML o en alguno de sus dialectos (Json, por ejemplo) para que puede atravesar proxies, gateways, etc.

OData (Open Data Protocol) está íntimamente relacionada con REST. Este protocolo, que también es abierto y público, permite la creación y consumo de consultas REST. Que es, como se puede utilizar y programar OData se trata en la segunda parte de este artículo.

Finalmente, REST utiliza básicamente XML como lenguaje de comunicación, aunque es posible configurar el protocolo para que utilice JSON. JSON (JavaScript Object Notation) es un formato de intercambio de información, basado en XML, fácil de leer y entender por humanos, y fácil de generar por computadores pues utiliza convenciones que son familiares a lenguajes de programación de la familia de “C” (CSharp, C++, Java, Perl, Python, etc.). Además, JSON utiliza menos ancho de banda que XML por el hecho de ser menos “locuaz”.

## WebAPI

ASP.NET WebAPI es el Framework de Microsoft para crear servicios HTTP y REST. Es bastante similar a ASP.NET MVC (Model View Controller) pues contiene enrutadores, controladores, acciones, filtros e inyección de dependencias, aunque no forma parte del Framework MVC. WebAPI es una parte básica de la plataforma ASP.NET y puede ser utilizada con cualquier tipo de aplicaciones, incluyendo las creadas utilizando MVC, aplicaciones Web y aplicaciones Windows. También puede ser utilizada para la creación de aplicaciones de Servicios Web.

WebAPI es una excelente Framework para exponer data y servicios a diferente tipos de dispositivos, además de ser open source y una plataforma ideal para construir servicios REST sobre el Framework .NET. A diferencia de servicios REST creados con WCF (Windows Communication Foundation), utiliza y expone todas las características de HTTP y el desarrollador no necesita definir ningún tipo de configuraciones especiales para que el software funcione con diferentes dispositivos.

Aunque WebAPI se puede utilizar con Visual Studio 2012 instalando el Framework adicionalmente a la instalación básica, Visual Studio 2013 lo instala por defecto. La versión actual de WebAPI, 2.0, forma parte de todas las versiones de Visual Studio 2013, hasta el update 4. Microsoft ha anunciado que la versión 2.2 del WebAPI, que soporta las características de la última versión de OData (4.0), formará parte de Visual Studio en la próxima versión.

Aunque servicios REST se pueden programar también con WCF, WebAPI proporciona todos los elementos básicos por medio de las plantillas de Visual Studio, aligerando y facilitando considerablemente la creación de servicios. Aunque no existen cifras oficiales de Microsoft, se pueden encontrar pruebas de carga en Internet (<http://blogs.msdn.com/b/fkaduk/archive/2014/01/07/rest-wcf-vs-webapi-throughput.aspx> por ejemplo) que muestran que la capacidad de respuesta de servicios creados con WebAPI, aunque estadísticamente ligeramente inferior a los creados con WCF, no difieren significativamente, pero las ventajas que ofrece en rapidez de programación la señalan como la herramienta indicada para la creación de servicios REST.

Básicamente un servicio REST creado con WebAPI consiste de dos componentes principales: Modelos y Controladores.

Un servicio Web puede tener uno o más modelos y uno o más controladores. A diferencia de un programa MVC, un servicio creado con WebAPI no necesita Viewers (un servicio Web no tiene interface propia):

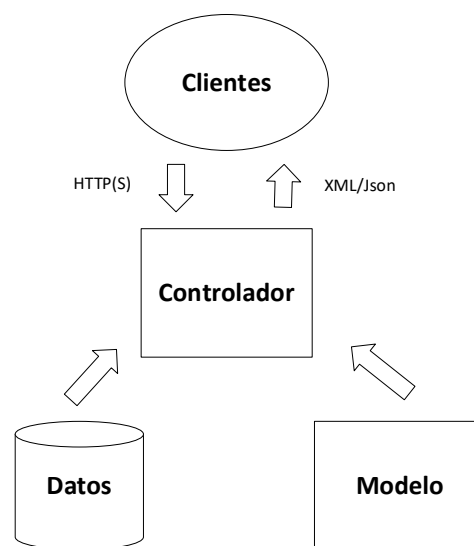


Imagen 1.- Funcionamiento de un servicio REST creado con WebAPI

Los Clientes envían consultas al Controlador utilizando HTTP(S) y reciben respuestas en forma de XML o JSON. El Modelo es una clase de .NET que ofrece una representación de las entidades que el servicio maneja. Los Datos puede ser cualquier tipo de repositorio: bases de datos, otros servidores como SharePoint o CRM, archivos, etc. El Controlador es el corazón del sistema: accede a los Datos y los procesa utilizando su lógica interna transformándolos en las entidades definidas por el Modelo. El Framework de WebAPI se encarga de manejar toda la comunicación con los Clientes, el ruteo de las consultas al Controlador apropiado, la validación de las consultas, etc.

## Creación de un Servicio REST con WebAPI 2

El siguiente ejemplo ha sido pensado para que sea lo más sencillo posible, de tal forma que se pueda demostrar lo fácil que es crear un servicio REST con el WebAPI. Por simplicidad, inicialmente el ejemplo solamente utiliza como fuente de datos un array creado en el mismo servicio, pero en los siguientes artículos se irá ampliando para acceder datos externos provenientes de SharePoint. El ejemplo y las imágenes han sido creados utilizando Visual Studio 2013 Update 4 y C#, pero cualquier versión de Visual Studio 2013 (y Visual Basic) producirá los mismos resultados.

- Inicie Visual Studio. Seleccione “Nuevo proyecto” y luego seleccione la plantilla “Web” – “Aplicación web ASP.NET”. Asígnele un nombre al proyecto.
- En la ventana de Nuevo proyecto, seleccione “Empty” como plantilla y “Web API” en la sección de “Agregar



carpetas y referencias principales para:”. Si la casilla de “Host en la nube” esta seleccionada, deselectionela pues se va a crear el proyecto localmente. Igualmente, seleccione “Sin autenticación” desde el botón “Cambiar autenticación”, si es que el botón está activado.

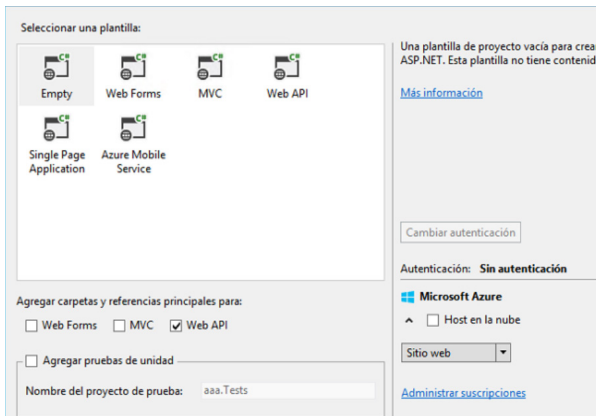


Imagen 2.- Proyecto de Visual Studio para el servicio REST con WebAPI.

- La plantilla de Visual Studio crea toda la estructura del proyecto, incluyendo carpetas para los controladores y modelos, y todas las referencias iniciales necesarias. Note que si en el punto anterior se selecciona la plantilla “Web API”, Visual Studio crea un sitio completo y totalmente funcional de MVC que funciona con el Web API. Este proyecto puede ser interesante como punto de partida para un sitio funcional y como método de aprendizaje, pero no es práctico para crear un nuevo servicio REST desde el principio.
- Seleccione la carpeta “Models” en el Explorador de soluciones y utilizando el menú contextual (botón derecho del ratón) seleccione “Agregar” – “Clase”. Asígnele un nombre a la nueva clase (en el ejemplo se llama “Elemento”).
- En la ventana de la nueva clase, bajo la definición de la clase, agregue el código que se muestra a continuación:

```
public class Elemento
{
    public Elemento()
    {
    }

    public Elemento(int Id, string Nombre, string Categoria, decimal Precio)
    {
        this.Id = Id;
        this.Nombre = Nombre;
        this.Categoria = Categoria;
        this.Precio = Precio;
    }

    public int Id { get; set; }
    public string Nombre { get; set; }
    public string Categoria { get; set; }
    public decimal Precio { get; set; }
}
```

El Modelo “Elemento” define las entidades que el servicio va a utilizar, en este caso, una llamada “Elemento” que

tiene como propiedades un identificador, nombre, categoría y precio. La clase además tiene un constructor que inicializa los valores de la entidad.

- Seleccione la carpeta “Controllers” en el Explorador de soluciones y utilizando el menú contextual (botón derecho del ratón) seleccione “Agregar” – “Controlador”.
- En la ventana de Scaffold seleccione “Controlador de Web API 2 – en blanco”. Asígnele un nombre al nuevo archivo respetando el final del nombre que debe ser como esta predefinido (“Controller”). En el ejemplo se llama “ElementosController”.

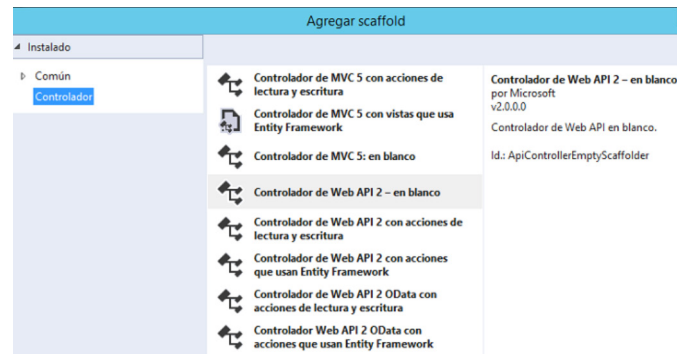


Imagen 3.- Selección del Controlador para el servicio REST con WebAPI.

**Con la estrategia definida por Microsoft indicando que todos sus servidores deberán ser utilizados principalmente desde la nube (y preferiblemente la nube de Microsoft: Azure y Office 365), es técnicamente indispensable que exista una forma de acceder remotamente a la información contenida en ellos.**

- En la nueva ventana, bajo la definición de la clase, agregue el siguiente código:

```
public class ElementosController : ApiController
{
    private static List<Elemento> myElementos = Inicializador();

    private static List<Elemento> Inicializador()
    {
        List<Elemento> ElementosIniciales = new List<Elemento>();
        ElementosIniciales.Add(new Elemento(1, "aaa", "III", 1.1M));
        ElementosIniciales.Add(new Elemento(2, "bbb", "222", 2.2M));
        ElementosIniciales.Add(new Elemento(3, "ccc", "333", 3.3M));
        return ElementosIniciales;
    }

    public IEnumerable<Elemento> GetAllElementos()
    {
        return myElementos;
    }
}
```

```
public IHttpActionResult GetElemento(int id)
{
    Elemento unElemento = myElementos.Where(e => e.Id == id).
    First();
    if (unElemento == null)
    {
        return NotFound();
    }
    return Ok(unElemento);
}
```

Note que la clase hereda de “ApiController” en lugar de “Controller” como lo hace un programa basado en MVC. Inicialmente, como se indicó anteriormente, se define un array con valores fijos para el ejemplo. Dos métodos definen las acciones que el servicio Web va a realizar: retornar todos los Elementos de la Lista genérica (GetAllElementos) y retornar un Elemento específico indicando su identificador como parámetro de entrada (GetElemento). El primer método devuelve el array completo y el segundo utiliza una función lambda para encontrar el elemento solicitado. Note que se utilizan dos métodos especiales (“NotFound” y “Ok”) que el WebAPI se encargara de renderizar apropiadamente como códigos de HTML (404 y 200) en la respuesta al cliente.

Abra el archivo “App\_Start” – “WebApiConfig.cs” y observe en la línea 19 como se ha definido la ruta de acceso a los métodos (endpoints): “api/{controller}/{id}”. Esto significa que el framework del WebAPI reemplazara “{controller}” con el nombre del archivo que contiene el Controlador, y “{id}” con el identificador que se encuentre en el URL

## Testeo del Servicio Web

En principio ya se tiene un servicio REST totalmente funcional. Inicie el proyecto desde Visual Studio (F5 o “Depurar” – “Iniciar depuración”). Visual Studio compila el proyecto y lo hostea utilizando IIS Express utilizando un puerto que se puede definir en las Propiedades del proyecto, sección “Web” – “Url del proyecto”, o utilizando el que Visual Studio mismo le asigne de forma aleatoria. Note que el navegador mostrará un mensaje indicando “Error HTTP 403.14 – Forbidden”, que se puede negar pues un servidor REST no tiene interface de usuario, solamente es necesario que este configurado correctamente en IIS (o en IIS Express, como lo hace Visual Studio)

En el navegador utilice el URL “http://localhost:[NumeroPuerto]/api/elementos”. Internet Explorer no muestra los resultados sino que permite descargarlos en un archivo separado o abrir el archivo (“elementos.json”) directamente. Firefox si muestra el resultado directamente en la pantalla. Si se utiliza el botón de “Abrir” en Internet Explorer, el resultado aparecerá en Visual Studio mismo (si no es que otro programa ha sido configurado para abrir archivos con extensión .json):

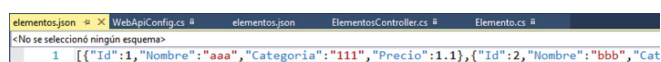


Imagen 4.- Respuesta del servicio REST mostrando todos los Elementos.

- De igual forma, utilizando el URL “http://localhost:[NumeroPuerto]/api/elementos/1”, el servicio retornara los valores del Elemento solicitado:



Imagen 5.- Respuesta del servicio REST mostrando un Elemento.

Aunque Internet Explorer no es la mejor herramienta para testear Servicios REST, otros navegadores, como Firefox, disponen de mejores posibilidades para ver e interactuar con los resultados. Por ejemplo, Firefox muestra los resultados directamente en la pantalla sin necesidad de descargar y abrir un archivo como en IE.

Probablemente la mejor herramienta actualmente para depurar y trabajar con servicios REST es Fiddler (<http://www.telerik.com/fiddler>). Esta herramienta permite no solo interceptar las consultas y respuestas y mostrarlas en diferentes formatos, sino que también permite crear y enviar consultas directamente al servicio sin necesidad de crear una aplicación cliente especializada, entre otras muchas cosas.

La siguiente figura muestra Fiddler enviando una consulta (pestaña “Composer”) al servicio acabado de crear. Note que se está usando el verbo “GET” y el URL creado por IIS Express y Visual Studio (“http://localhost:33497”) junto con el URI que llama al servicio (“/api/elementos”):

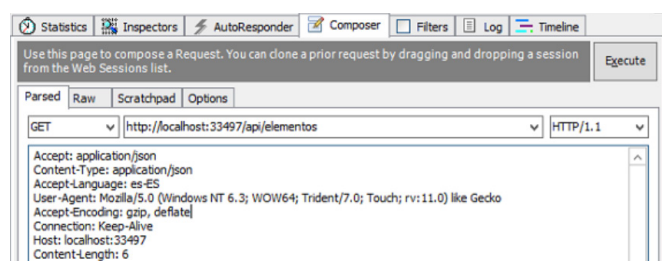


Imagen 6.- Consulta al servicio REST utilizando Fiddler.

Para ver la respuesta del servicio, utilice la pestaña “Inspectors” de Fiddler. Es posible renderizar la respuesta en XML, en Json o verla tal como la envía el servicio (“Raw”). Adicionalmente, Fiddler ofrece mucha más información al respecto, como por ejemplo los cookies utilizados.

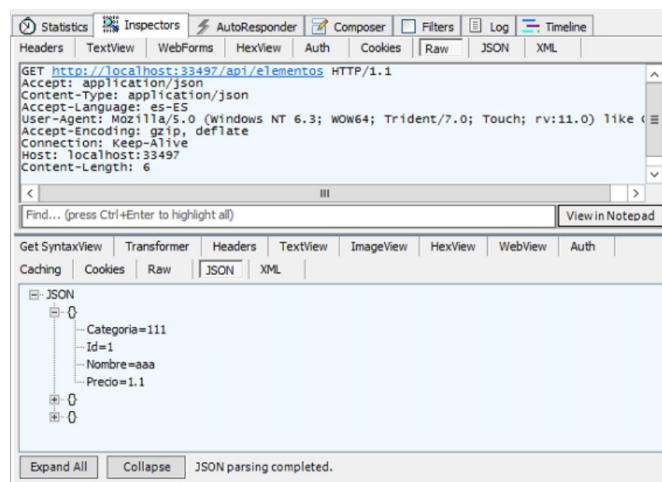


Imagen 7.- Respuesta del servicio REST utilizando Fiddler.

## Operaciones CRUD con el Servicio REST

Hasta ahora, el ejemplo solamente permite hacer consultas para ver todos los elementos, o solo uno de ellos. Pero REST permite realizar todas las operaciones CRUD (Create, Read, Update, Delete) con la información.

Amplíe la clase “ElementosController” con los siguientes tres métodos:

```
public HttpResponseMessage PostElemento([FromBody] int ElementoId)
{
    Elemento nuevoElemento = new Elemento();
    nuevoElemento.Id = ElementoId;
    myElementos.Add(nuevoElemento);
    HttpResponseMessage myRespuesta =
    Request.CreateResponse(HttpStatusCode.Created);
    myRespuesta.Headers.Location = new Uri(Request.
    RequestUri + "/" +
    (nuevoElemento.Id).ToString());
    return myRespuesta;
}

public void PutElemento(int id, [FromBody] string Nombre)
{
    Elemento myElemento = myElementos.Where(e => e.Id == id).
    First();
    myElemento.Nombre = Nombre;
}

public void DeleteElemento(int id)
{
    myElementos.Remove(myElementos.Where(e => e.Id == id).
    First());
}
```

El primer método crea un nuevo elemento en el array y devuelve una respuesta del tipo “HttpResponseMessage” indicando el resultado final de la operación. Note en este método una de las restricciones de Json: solamente se puede enviar un parámetro decorado como “FromBody”, lo que obliga a crear primero el elemento solamente con su identificador como en el ejemplo, y luego, en operaciones subsiguientes, agregarle la información adicional (Nombre, Categoría y Precio). Hay principalmente dos formas de solucionar la limitación:

- Enviar un objeto serializado que contenga toda la información, en este caso un objeto de tipo “Elemento”.
- Enviar los parámetros separados como QueryString.

El segundo método modifica un Elemento existente, utilizando como parámetros su identificador y el valor del campo a modificar. Aquí vale la misma restricción descrita anteriormente: solamente se puede enviar un valor utilizando la decoración de “FromBody”.

Finalmente el tercer método elimina un Elemento utilizando como parámetro de entrada el identificador del mismo. En estos dos últimos métodos se utiliza una consulta Lambda para encontrar rápidamente el elemento a modificar o eliminar. Los tres métodos utilizan los verbos estándar de HTML para las operaciones a realizar: POST para crear un elemento, PUT para modificarlo y DELETE

para eliminarlo. El WebAPI se encarga de enrutar las consultas a los métodos respectivos, por ejemplo la consulta POST al método “PostElemento”.

La Figura 8 muestra la consulta enviada al Servicio por medio de Fiddler.

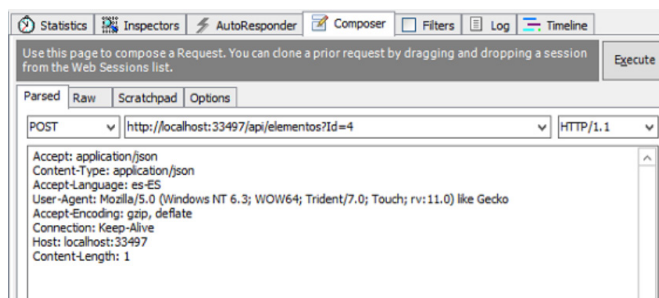


Imagen 8.- Consulta POST al servicio REST utilizando Fiddler.

En la Figura se puede ver que el valor esperado por el método (el valor del Identificador) se envía tanto en la consulta como en el Cuerpo de la consulta (“Body”) y que se está utilizando un verbo “POST”. También es indispensable indicar que el “Content-Type” es del tipo “application/json”, como se ve en la figura.

Para modificar (y eventualmente eliminar) un elemento, la consulta es muy similar, necesitando solamente modificar el verbo y el body. La Figura 9 muestra una consulta para modificar el Nombre de un elemento:

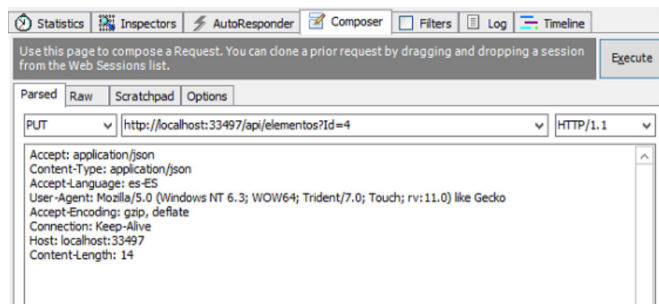


Imagen 9.- Consulta para modificar un elemento utilizando Fiddler.

## Conclusión

SharePoint 2013 y SharePoint Office 365 utilizan extensivamente REST como medio de comunicación para manipular información. La creación de Servicios REST se ha simplificado considerablemente desde la aparición del Framework WebAPI de Microsoft, que se integra perfectamente en Visual Studio 2013 y permite la creación de servicios REST rápida e intuitivamente, haciendo que los desarrolladores se dediquen a escribir solamente el código que importa, es decir, la lógica del servicio mismo, y no pierdan tiempo creando y modificando configuraciones que se pueden realizar automáticamente.

**GUSTAVO VELEZ**

MVP de SharePoint Server

[gustavo@gavd.net](mailto:gustavo@gavd.net)

<http://www.gavd.net>

# Movilidad sobre SharePoint - Parte I

Algunas de las preguntas típicas que nos solemos hacer los desarrolladores con frecuencia a la hora de abordar una solución de movilidad sobre SharePoint son: ¿se verá nuestra Intranet con el iPad?, ¿tiene un diseño adaptable?, ¿será compatible con todos los dispositivos?

Estas y otras preguntas, forman cada vez más, parte de los requisitos esenciales en todos los proyectos informáticos. A continuación describiremos algunos métodos para llevarlo a cabo tanto en SharePoint 2010 como en SharePoint 2013.

Vamos a diferenciar estas alternativas en 2 grupos o familias que serían:

- Aplicaciones Personalizadas
- Personalización de la interfaz

Las acciones que llevaremos a cabo serán modificar la representación de la información para poder adaptarla según el dispositivo. Estamos limitados en cuanto a que la interfaz no puede ser totalmente personalizada, pero conseguimos nuestro objetivo de una manera rápida y sencilla. Las herramientas que usaremos serán:

- Device Channels (no disponible en SharePoint 2010)
- CSS Media Queries

## Device Channels

Esta nueva característica solo disponible en SharePoint 2013, nos permite representar nuestra Intranet de diferentes maneras dependiendo del dispositivo, forma parte de las características de publicación y nos permite modificar los estilos, imágenes y otros aspectos sin variar la Url, y a través del User Agent, se hace la detección del dispositivo.

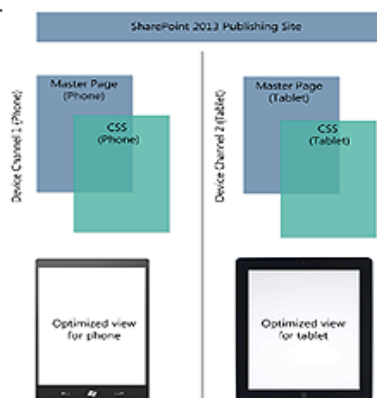


Imagen 1.- Ejemplo de estructura con Device Channels.

## CSS Media Queries

Este módulo de CSS3 permite crear estilos específicos para cada tamaño, de manera que en una sola hoja de estilos podemos agregar todas las casuísticas dependiendo del tipo de resolución, sin tener en cuenta el dispositivo.

```
/*Hide items by default (not mobile device view)*/
#mobileHeader,
#mobileMenu,
#logoMobile,
#quickButtons{
    display:none;
}

/*---Mobile Device view ----- */
@media screen and (max-device-width: 640px) {
    #mobileHeader,
    #mobileMenu,
    #logoMobile,
    #quickButtons{
        display:block;
    }
    body{
        font-size:.8em;
    }
    #contentwrapper{
        background-image:none !important;
    }
}
```

Imagen 2.- Ejemplo de CSS aplicando media queries.

## Aplicaciones Personalizadas

En este apartado, las acciones serán crear una aplicación totalmente personalizada, pero que consuma los datos de nuestra Intranet, de manera que podamos crear interfaces nativas que nos permite mejorar la experiencia de usuario, aunque por otro lado necesitaremos más tiempo para conseguir nuestro objetivo. Las herramientas, tecnologías de desarrollo y formas de consumo de la información que usaremos serán:

### Tipos de Aplicaciones

- App Híbrida - Apache Cordova /PhoneGap.
- App Universal Windows - W8 & WP8.
- App Nativa Multiplataforma – Xamarin.

### Tecnología de Desarrollo

- JavaScript
- C#

### Consumo de la información

- SharePoint 2013: Servicio API REST, JSOM (Modelo Objeto JavaScript) y CSOM (Modelo Objeto Cliente)
- SharePoint 2010: Servicio SOAP, API REST (Limitado), JSOM (Modelo Objeto JavaScript) y CSOM (Modelo Objeto Cliente)



## Publicación en los Market Place

- Es necesario tener cuenta y licencia para publicar en el market de aplicaciones de las distintas plataformas

## Como crear una App Híbrida Apache Cordova /PhoneGap

Cuando hablamos de App Híbrida nos referimos a una aplicación Web que la encapsulamos en una aplicación móvil. Esto nos permite emular la capa visual de cada dispositivo a través de un WebView o capa que los Frameworks nombrados anteriormente nos permiten acceder a la mayoría de funcionalidades del dispositivo en el que estamos corriendo la aplicación.

A través de estas funcionalidades podemos construir nuestra capa de presentación con la funcionalidad, y acceder a la mayoría de eventos de nuestro dispositivo pudiendo a su vez, consumir la información de nuestro SharePoint:

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1">
  <script src="components/loader.js"></script>
  <link rel="stylesheet" href="components/loader.css">
  <link rel="stylesheet" href="css/style.css">
  <script>
    // PhoneGap event handler
    document.addEventListener("deviceready", onDeviceReady, false);
    function onDeviceReady() {
      console.log("PhoneGap is ready.");
      navigator.notification.alert('Aviso', null, 'CompartiMoss', 'Cerrar');
    }
  </script>
  <script type="text/javascript" charset="utf-8" src="phonegap-demo/main.js"></script>
</head>
<body onload="init();">
  <h1>Welcome to Monaca!</h1>
  <a class="button--large" href="phonegap-demo.html">Start Demo</a>
</body>
</html>
```

Imagen 3.- Ejemplo Inicialización y Mensaje Alerta.

## Como Empezar

- Herramientas:
  - Visual Studio 2013 Update 3 + Aplicaciones Híbridas Multi Dispositivo para Visual Studio CTP2.0.
- Tener a mano las referencias de la API:
  - [http://cordova.apache.org/docs/en/4.0.0/guide\\_support\\_index.md.html#Platform%20Support](http://cordova.apache.org/docs/en/4.0.0/guide_support_index.md.html#Platform%20Support)
- Seguir la documentación publicada en MSDN:
  - <http://msdn.microsoft.com/en-us/library/dn771545.aspx>

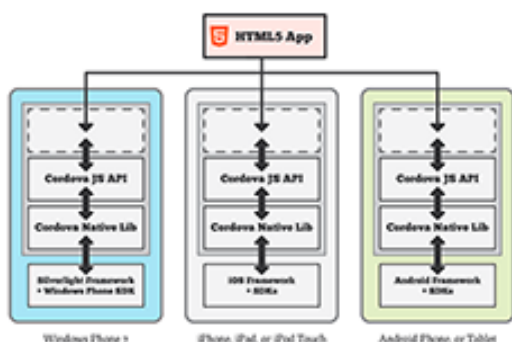


Imagen 4.- Muestra la estructura de un proyecto Cordova.

## Como crear una App Universal Windows (WP8 & W8)

Cuando hablamos de App Universal hablamos de aplicaciones creadas para Windows y Windows Phone que comparten código, controles, estilos y otros componentes entre dos proyectos de Visual Studio. Esto permite desarrollar más rápidamente la aplicación y que sea más fácil de mantener. Una vez tenemos una aplicación de una de las 2 plataformas, es sencillo poder crear una aplicación para la otra, y aunque compartimos código, siempre nos permite poder diferenciarlas en el caso de controles no soportados, por ejemplo.

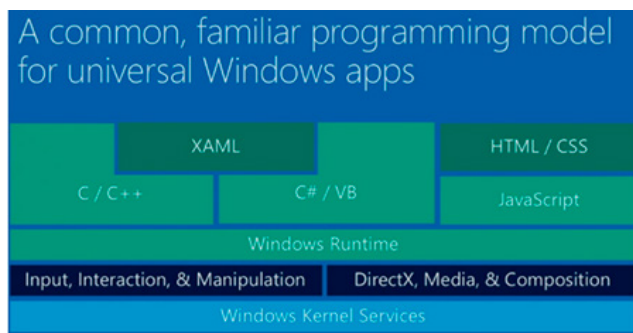
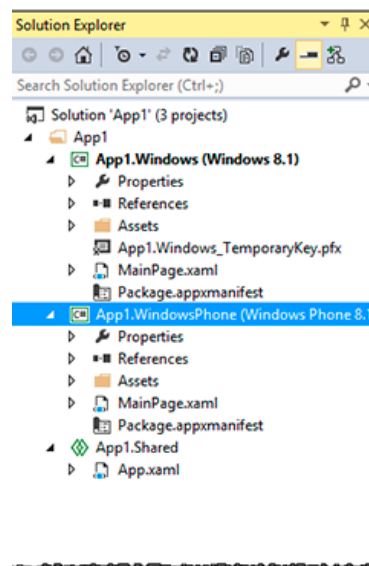


Imagen 5.- Muestra el ejemplo de un proyecto de Universal APP y el modelo de programación.

## Ejemplo

```
protected override async void OnLaunched(LaunchActivatedEventArgs e)
{
    Frame rootFrame = Window.Current.Content as Frame;

    // Do not repeat app initialization when the Window already has content,
    // just ensure that the window is active

    if (rootFrame == null)
    {
        // Create a Frame to act as the navigation context and navigate to the first page
        rootFrame = new Frame();

        // Set the default language
        rootFrame.Language = Windows.Globalization.ApplicationLanguages.Languages[0];

        if (e.PreviousExecutionState == ApplicationExecutionState.Terminated)
        {
            // Restore the saved session state only when appropriate
            try
            {
                await SuspensionManager.RestoreAsync();
            }
            catch (SuspensionManagerException)
            {
                //Something went wrong restoring state.
                //Assume there is no state and continue
            }
        }
    }
}
```

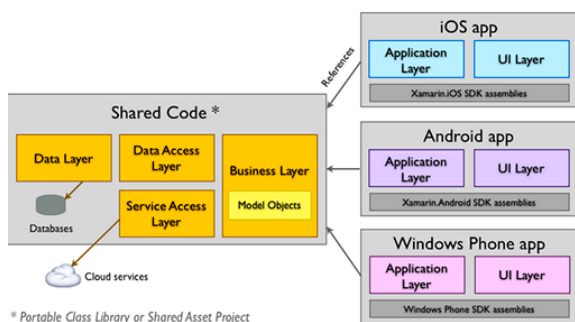
## Como Empezar:

- Herramientas:
  - Visual Studio 2013 Update 2 + Window App Certification Kit + Windows Phone 8.1 SDK
- Tener a mano las referencias de la API:
  - <http://msdn.microsoft.com/en-us/library/windows/apps/br211377.aspx>
- Seguir la documentación publicada en el sitio de desarrollo para Windows:
  - <http://dev.windows.com/en-us/develop/building-universal-windows-apps>

## Como crear una App Nativa Multi-plataforma – Xamarin

Cuando hablamos de App Multiplataforma, nos referimos a una aplicación que es capaz de mostrar controles nativos a partir del código que escribamos. En el caso de Android sería Java, en el caso de iOS sería Objective C o Swift, y en el caso de Windows Phone sería C# y Xaml. Xamarin por su parte, es una plataforma de desarrollo que nos permite desarrollar nuestras aplicaciones desde código C# y poder compilarlo y paquetizarlo al resto de plataformas móviles como Android, iOS y Windows Phone.

Podemos compartir la capa de datos, negocio o servicio, y sólo nos quedaría personalizar la capa de presentación para cada plataforma como vemos en la siguiente imagen. Viendo esta problemática es cómo surgió la nueva característica: Xamarin-Forms



\* Portable Class Library or Shared Asset Project

Imagen 6.- Muestra la estructura de un proyecto Xamarin.

## Xamarin Forms

Esta nueva funcionalidad surgió hace poco y lo que nos

permite es desarrollar interfaces de diseño nativas con código C# compartido entre todas las plataformas. Con un mismo código, tenemos 3 capas de diseño iguales, una para cada plataforma.

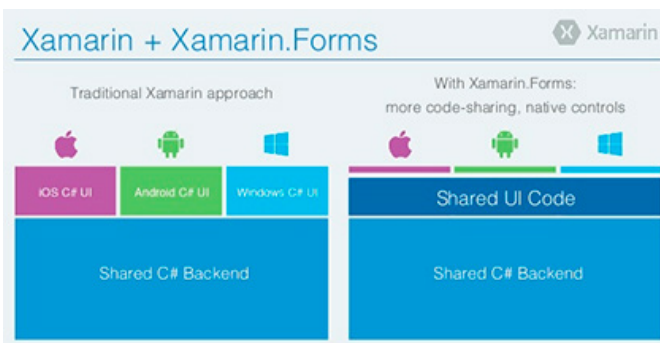


Imagen 7.- Comparativa entre Xamarin y Xamarin Forms.

```
using Xamarin.Forms;

var profilePage = new ContentPage {
    Title = "Profile",
    Icon = "Profile.png",
    Content = new StackLayout {
        Spacing = 20, Padding = 50,
        VerticalOptions = LayoutOptions.Center,
        Children = {
            new Entry { Placeholder = "Username" },
            new Entry { Placeholder = "Password", IsPassword = true },
            new Button {
                Text = "Login",
                TextColor = Color.White,
                BackgroundColor = Color.FromHex("770066")
            }
        }
    }
};

var settingsPage = new ContentPage {
    Title = "Settings",
    Icon = "Settings.png",
    (...)
};

var mainPage = new TabbedPage { Children = { profilePage, settingsPage } };
```

Imagen 8.- Ejemplo creando una interfaz UI con Xamarin Forms.

## Como Empezar

- Herramientas:
  - Visual Studio 2013 Update 2 + Xamarin for Visual Studio + Xamarin Business Licence
- Tener a mano las referencias de la API:
  - <http://developer.xamarin.com/guides/>
- Seguir la documentación publicada en el sitio de desarrollo para Xamarin:
  - <http://developer.xamarin.com/>

**Algunas de las preguntas típicas que nos solemos hacer los desarrolladores con frecuencia a la hora de abordar una solución de movilidad sobre SharePoint son: ¿se verá nuestra Intranet con el iPad?, ¿tiene un diseño adaptable?, ¿será compatible con todos los dispositivos?**

## En resumen...

El resumen de las opciones que tenemos para movilizar SharePoint son las siguientes:

Solución	Tecnología	Pros	Contras
Device Channels	HTML5 + CSS3	<ul style="list-style-type: none"> <li>✓ Master Page y CSS específico por dispositivo</li> <li>✓ Estructura única aislada , por dispositivo</li> </ul>	<ul style="list-style-type: none"> <li>✓ Tener que crear toda la estructura y tener que mantenerla para cada dispositivo.</li> <li>✓ No disponible en SharePoint 2010</li> </ul>
CSS Media Queries	HTML5 + CSS3	<ul style="list-style-type: none"> <li>✓ Estándar web , <b>fácil de mantener</b> y modificar</li> <li>✓ Centralizado en una hoja de estilos.</li> </ul>	<ul style="list-style-type: none"> <li>✓ No permite distinguir por tipo de dispositivo</li> <li>✓ Todo debe ser personalizado y adaptado a SharePoint.</li> </ul>
App Híbrida - Apache Cordova /PhoneGap	HTML5 + CSS3	<ul style="list-style-type: none"> <li>✓ Estándar web , fácil de mantener y modificar</li> <li>✓ Capa de Acceso a funciones del dispositivo</li> <li>✓ Multiplataforma</li> </ul>	<ul style="list-style-type: none"> <li>✓ Capa Visual no nativa</li> <li>✓ Peor rendimiento que una APP Nativa</li> <li>✓ No accesibles todas las funciones del dispositivo</li> </ul>
App Universal Windows W8 & WP8	XAML + C# HTML5 + Javascript	<ul style="list-style-type: none"> <li>✓ Compartición de Capa Visual y capa de datos</li> <li>✓ Proyectos aislados pero compartiendo código</li> </ul>	<ul style="list-style-type: none"> <li>✓ No es multiplataforma</li> <li>✓ Solo funciona en dispositivos W8 y WP8</li> </ul>
App Nativa Multiplataforma - Xamarin	C#	<ul style="list-style-type: none"> <li>✓ Multiplataforma</li> <li>✓ APP Nativa</li> <li>✓ Sin Necesidad de aprender otros lenguajes</li> <li>✓ Compartición de código</li> </ul>	<ul style="list-style-type: none"> <li>✓ Limitado a los controles por defecto o crear controles personalizados</li> <li>✓ Lentitud al depurar</li> </ul>

## Conclusiones

A través de los ejemplos mostrados, podemos concluir que tenemos muchas maneras y diferentes tecnologías para mostrar la información de nuestro SharePoint. Todas ellas son válidas, solamente hay que buscar la más adecuada dependiendo de nuestros requisitos. En el próximo artículo hablaremos sobre cómo crear una App y como consumir información de nuestro SharePoint.

**RUBÉN TORIBIO**

SharePoint Architect en Necsia

@rtoribio

**JORGE SERRANO**

Microsoft MVP en Necsia

@JorgeSerran0





# Como utilizar las API's de Office 365 en aplicaciones móviles multiplataformas desarrolladas con Xamarin

Si echamos la vista al evento Connect() de Microsoft (evento en el que se presentó la nueva versión de Visual Studio) hay dos grandes novedades en cuanto a desarrollo. La primera las mejoras incluidas en Visual Studio para realizar aplicaciones multiplataforma ya sea a través de Apache Cordova (HTML5, JavaScript y CSS3) o bien a través de Xamarin (realizando aplicaciones nativas haciendo uso de C#/F#). La segunda de las novedades es tener preparados nuestros desarrollos para que sean fácilmente escalables.

Dada estas dos premisas dentro de nuestro entorno diario está claro donde más nos repercute: Office365. Cada vez salen muchas más novedades para Office365 que no disponemos directamente en nuestra versión OnPremise. Novedades como el App Launcher, Office Delve, Office Video, Office365 API's etc... Por lo que más pronto que tarde tendremos que pensar en la forma en la que podemos utilizar todas estas novedades en nuestros desarrollos.

Partiendo de estas cuestiones, en este artículo vamos a ver como poder realizar aplicaciones multiplataforma haciendo uso de Xamarin y que consuman los servicios que nos facilita Office 365.

## ¿Qué es Xamarin?

Xamarin es una herramienta para los desarrolladores de aplicaciones móviles, y la novedad de esta herramienta es la capacidad que tiene para que el desarrollador escriba su App en lenguaje C#/F# y el mismo código sea traducido para ejecutarse en iOS, Android y Windows Phone. Xamarin se puede integrar bien con Visual Studio o desarrollar las aplicaciones con su propio IDE Xamarin Studio.

La primera propuesta de desarrollo era la siguiente: desarrollar una interfaz de usuario nativa con los controles de cada plataforma, consiguiendo de esta forma la gran experiencia nativa que estábamos buscando, y a nivel de lógica de la aplicación en vez de desarrollar cada App con lenguajes diferentes como C++, Java u Objective C, desarrollar usando C# para todas, compartiendo por tanto la lógica de negocio de las aplicaciones para las tres plataformas móviles. Esto se consigue gracias a que Xamarin es compatible con los PCL's (Portable Class Library). Propicia que podamos compilar una librería solo una vez con unos mismos parámetros y dicha librería pueda ser utilizada desde la aplicación de Windows Phone, iOS e Android de Xamarin.



Imagen 1.- Esquema de desarrollo tradicional con Xamarin.

Esta forma de desarrollo tiene como principal ventaja, el uso de un único lenguaje para todas las aplicaciones pero tiene como gran inconveniente que tenemos que desarrollar la interfaz gráfica en cada una de las plataformas. Esto es un problema porque aunque la mayor parte del desarrollo se realice en C#, necesitaremos conocer las particularidades de cada plataforma.

En Xamarin, conocedores de esta problemática, han evolucionado su producto sacando Xamarin Forms. Este tipo de proyecto lo que hace es que desarrollemos una única interfaz, siendo esta UI la misma en cada uno de los dispositivos en los que se visualiza, es decir, el desarrollo ahora quedaría de la siguiente forma:

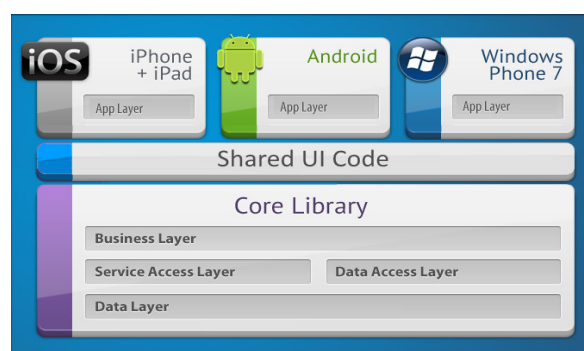


Imagen 2.- Arquitectura de Xamarin Forms

## ¿Cómo integrarlo con Office 365?

En los artículos anteriores hemos visto como incluir estas API's dentro de desarrollos ASP.NET o Windows 8 es muy sencillo, prácticamente es transparente para el desarrollador. Pero como he comentado en la introducción el objetivo actual es que cualquier desarrollo se pueda utilizar en cualquier dispositivo y cualquier plataforma. Para poder utilizar las API's de Office 365 en cualquier tipo de proyecto se necesita autenticar la aplicación con Office



365 con el fin de obtener acceso a los datos de sus usuarios. La autenticación de la aplicación incluye tres pasos: crear el contexto de Office 365, obtener el token de acceso, y a continuación ya podemos realizar las peticiones a la API de Office365 que deseemos acceder (Service Calendar, Contacts, Mail, y MyFiles son los servicios por el momento).

## Proceso de Autenticación

Nuestro cliente que haga uso del servicio de Discovery de Office 365 debe de realizar el siguiente flujo para poder validar la App que estamos desarrollando:

1. Registrar su aplicación en el Portal de la Administración de Azure y configurar el código de la aplicación con el ID de cliente y redirigir URI. Luego, en el Portal de la Administración de Azure, configurar los permisos para la aplicación. Para más detalles de cómo realizarlo <http://blogs.encamina.com/desarrollando-sobresharepoint/2014/11/11/registrar-una-aplicacion-en-el-portal-de-azure/>
2. Su aplicación obtiene el email del usuario. Entra en contacto con Servicio de Discovery con la dirección de correo electrónico y a los recursos que la aplicación quiere acceder.

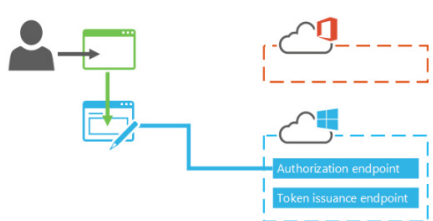


Imagen 3.- Paso 1 de autenticación

3. La aplicación se comunica con el servicio de autorización AD de Azure y el usuario se autentica y da el consentimiento sobre los recursos a los que tiene acceso (en el supuesto de que no se haya concedido el consentimiento antes). AD Azure emite un código de autorización.

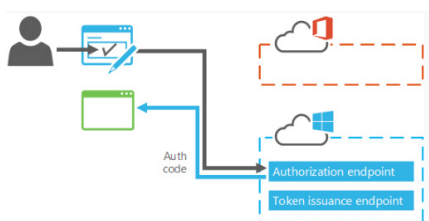


Imagen 4.- Paso 3 de autenticación.

4. Su aplicación obtiene el código de autorización. Azure devuelve un token de acceso.

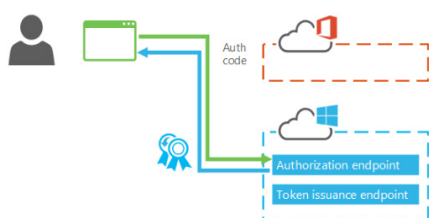


Imagen 5.- Paso 4 de autenticación

5. Su aplicación se comunica con el Service Discovery usando el token de acceso. Service Discovery regresa respuesta HTTP con identificadores de recursos y URI de acceso para las API's de Office 365.

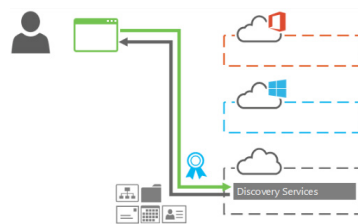


Imagen 6.- Paso 5 de autenticación

6. Su aplicación envía el token de actualización Azure AD, en este token se solicita el acceso a la API de Office 365 junto con sus recursos. Finalmente el AD Azure devuelve un token de acceso para el recurso especificado, junto con un token de actualización.

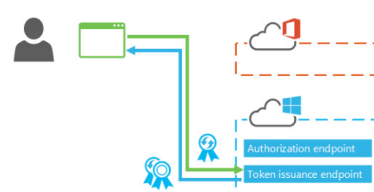


Imagen 7.- Paso 6 de autenticación

7. La aplicación ahora puede llamar a la Oficina 365 APIs utilizando el URI del Discovery Service y el token de acceso.

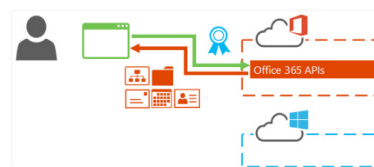


Imagen 8.- Peticiones a la APIs de Office365

## ¿Cómo lo implementamos?

### Requisitos Previos:

- Visual Studio 2013 Service Pack 2 (mínimo).
- Instalado Xamarin para Visual Studio 2013.
- Cuenta de Office365.
- Suscripción de Azure.

Para tener un entorno preparado para esta aplicación seguir los pasos del siguiente link: <https://github.com/OfficeDev/DevCampTraining/blob/master/02.Setting%20up%20the%20Environments/hands-on-lab.md>

### Objetivo:

Vamos a desarrollar una aplicación Android en la cual se nos pedirá por pantalla el Alias de la cuenta de correo que deseamos encontrar dentro de nuestro Exchange Online. Para llevar a cabo este desarrollo implementamos dos proyectos: 1 Portable Class Library y una aplicación Xamarin.Android

## Manos a la obra:

1. Iniciamos VS 2013 y creamos un proyecto de PCL (Portable Class Library): “Aplicaciones de escritorio -> Biblioteca de clases”.

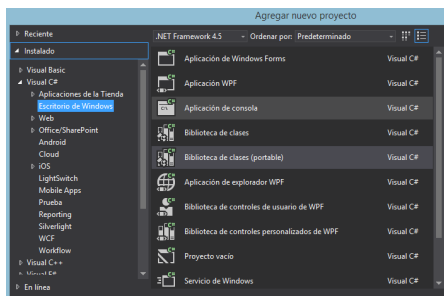


Imagen 9.- Tipo de Proyecto a elegir

2. Seleccionamos los proyectos en los que se puede utilizar esta librería. En nuestro caso además de seleccionar .NET Framework 4.5, Windows 8 y Windows Phone 8.1, seleccionaremos también Xamarin.iOS y Xamarin.Android.

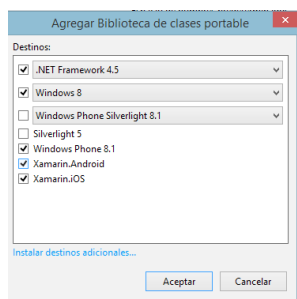


Imagen 10.- Selección plataformas donde se puede utilizar la biblioteca

3. Para facilitarnos la autenticación existe un paquete en NuGet que nos abstrae de implementar de forma manual todo el proceso anterior. Para ello en nuestra solución instalaremos el paquete NuGet Active Directory Authentication Library (ADAL). Esta librería va por la versión 3.0 (en fase previa).

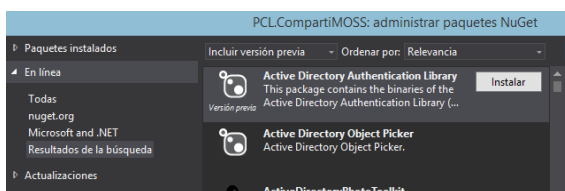


Imagen 11.- Instalación Paquete ADAL en NuGet.

NOTA: La versión que soporta acceso a PCL es la que está en fase previa, se ha anunciado que en futuras versión incluirá soporte para Xamarin.Forms. Se recomienda solamente poner en producción paquetes NuGet que están en estado Estable.

4. Instalaremos el paquete de NuGet JSON.Net para transformar las peticiones realizadas a la API de Office 365 en clases de C#.
5. A continuación dentro del proyecto añadiremos una clase CompartiMOSS donde incluiremos la autenticación contra Office 365 y la llamada a la API de Graph para obtener los emails que respon-

den a este alias. El primer paso que tenemos que hacer es llamar al ADAL para solicitar las credenciales al usuario por pantalla.

```
string graphRequest = String.Format(CultureInfo.InvariantCulture, "{0}/(1)/users?api-version={2}&$filter=mailNickname eq '{3}'", graphResourceUri, authResult.TenantId, graphApiVersion, alias);
HttpClient client = new HttpClient();
HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Get, graphRequest);
request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", authResult.AccessToken);
HttpResponseMessage response = await client.SendAsync(request);
string content = await response.Content.ReadAsStringAsync();
JResult = JObject.Parse(content);
```

Obtenido el token de acceso ya podemos realizar llamadas a las APIs de Office 365 de la siguiente forma:

```
AuthenticationContext authContext = new AuthenticationContext(commonAuthority);
if (authContext.TokenCache.ReadItems().Count() > 0)
    authContext = new AuthenticationContext(authContext.TokenCache.ReadItems().First().Authority);
authResult = await authContext.AcquireTokenAsync(graphResourceUri, clientId, returnUri, parent);
```

6. Nos crearemos otra clase, User.cs que es la clase donde almacenaremos la información obtenida del servicio REST. Esta clase tendrá información relativa del usuario.
7. Una vez ya tenemos nuestra librería, el siguiente paso es crearnos la aplicación en el dispositivo que deseamos utilizar. Para este ejemplo vamos a crear una App de Android. Agregamos nuevo proyecto Android-> Blank App (Android).

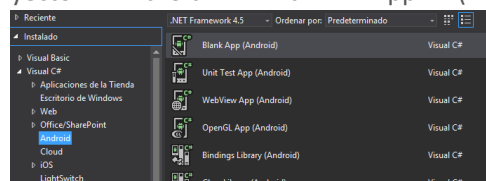


Imagen 12.- Agregamos proyecto Android a la solución.

En este nuevo proyecto, agregamos el paquete NuGet de ADAL y añadimos la referencia al PCL creado anteriormente quedando una estructura del proyecto como la siguiente:

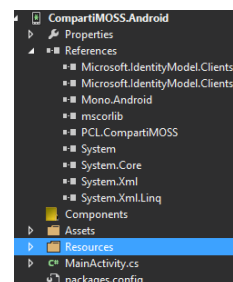


Imagen 13.- Estructura del Proyecto Android.

Ahora en el proyecto nos queda añadirle la Interfaz gráfica y añadirle la funcionalidad que hemos

implementado el PCL. Para agregar la UI a nuestra aplicación vamos a la carpeta Resources/layouts/Main.axml y le añadimos los componentes que nos hacen falta para nuestro desarrollo. En esta caso añadiremos diversos un EditText y varios TextView a dicho fichero. Para finalizar con el desarrollo dentro del fichero MainActivity.cs agregamos la llamada a la librería implementada anteriormente:

```
protected override void OnCreate(Bundle bundle)
{
    SetContentView(Resource.Layout.Main);
    Button button = FindViewById<Button>(Resource.Id.searchBut-
ton);
    button.Click += async delegate
    {
        EditText searchTermText = FindViewById<EditText>(Resource.
        Id.searchTermText);
        TextView statusResult = FindViewById<TextView>(Resource.
        Id.statusResult);
        TextView givenResult = FindViewById<TextView>(Resource.
        Id.givenResult);
        TextView surnameResult = FindViewById<TextView>(Re-
        source.Id.surnameResult);
        TextView upnResult = FindViewById<TextView>(Resource.
        Id.upnResult);
        TextView phoneResult = FindViewById<TextView>(Resource.
        Id.phoneResult);
        List<User> results = await PCL.CompartiMOSS.CompartiMOSS.
        SearchByAlias(searchTermText.Text, new AuthorizationParam-
        eters(this));
        statusResult.SetTextColor(Color.Green);
        givenResult.SetText(results[0].givenName, TextView.Buffer-
        Type.Normal);
        surnameResult.SetText(results[0].surname, TextView.Buf-
        ferType.Normal);
        upnResult.SetText(results[0].userPrincipalName, TextView.
        BufferType.Normal);
        phoneResult.SetText(results[0].telephoneNumber, Text-
        View.BufferType.Normal);
    };
}
```

Si ejecutamos nuestra aplicación observamos que nos solicita la autenticación contra Office 365 de la siguiente forma:

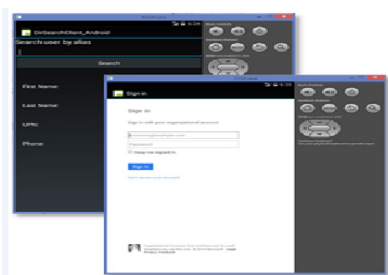


Imagen 14.- Aplicación Android accediendo a las APIs de Office 365.

## Conclusión

En este artículo hemos realizado una introducción al desarrollo de aplicaciones Nativa haciendo uso de Xamarin.

Hemos comprobado una de sus grandes ventajas: el compartir la lógica de negocio entre las diversas plataformas. También hemos visto.

*Si echamos la vista al evento **Connect()** de Microsoft (evento en el que se presentó la nueva versión de Visual Studio) hay dos grandes novedades en cuanto a desarrollo. La primera las mejoras incluidas en Visual Studio para realizar aplicaciones multiplataforma ya sea a través de Apache Cordova (HTML5, JavaScript y CSS3) o bien a través de Xamarin (realizando aplicaciones nativas haciendo uso de C#/F#).*

## Referencias:

- <http://msdn.microsoft.com/en-us/office/office365/howto/discover-service-endpoints>
- [http://www.cloudidentity.com/blog/2014/10/30/ADAL-NET-V3-PREVIEW-PCL-XAMARIN-SUPPORT/?utm\\_content=bufferb39be&utm\\_medium=social&utm\\_source=linkedin.com&utm\\_campaign=buffer](http://www.cloudidentity.com/blog/2014/10/30/ADAL-NET-V3-PREVIEW-PCL-XAMARIN-SUPPORT/?utm_content=bufferb39be&utm_medium=social&utm_source=linkedin.com&utm_campaign=buffer)
- <http://msdn.microsoft.com/en-us/office/office365/howto/authenticate-and-use-services>
- <http://blogs.encamina.com/desarrollandosobresharepoint/2014/11/11/registrar-una-aplicacion-en-el-portal-de-azure/>

**ADRIÁN DÍAZ CERVERA**

SharePoint Architect at Encamina

MVP SharePoint Server

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

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

[adiaz@encamina.com](mailto:adiaz@encamina.com) @AdrianDiaz81

# Forefront Identity Manager y la sincronización de usuarios en SharePoint 2010

## Qué es Forefront Identity Manager

En su versión básica (como la que tenemos con el miisclient.exe en SharePoint 2010) es usado para importar y exportar datos entre sistema de identidades.

## Para qué nos sirve en SharePoint

Nos sirve para ver (en línea) la ejecución de todas las operaciones que ocurren durante el proceso de sincronización cuando ejecutamos el servicio de User Profile (en la administración central de SharePoint).

Cuáles son los pasos a continuación

Los pasos que tenemos que seguir para usar FIM en SharePoint 2010 son los siguientes:

- Abrimos la administración central de SharePoint (en este caso 2010 y ejecutamos el servicio de User Profile (ya previamente configurado).



- Iniciamos la ejecución del servicio (en este caso sincronización completa)



- Para esto (dentro del servidor donde tenemos la administración central de SharePoint instalada) llegamos al path:

C:\Program Files\Microsoft Office Servers\14.0\Synchronization Service\UIShell\

- Y ejecutamos miisclient.exe (para el caso de SharePoint 2010, en SharePoint 2013 estará en la carpeta 15).

Name	Profile Name	Status	Start Time	End Time
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 07:29:14 p.m.	15/10/2014 07:29:38 p.m.
MOSS-Web318-a24	MOSS_FULLSYNC_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 07:29:32 p.m.	15/10/2014 07:29:13 p.m.
MOSSAD-COL	DS_FULLSYNC	success	15/10/2014 07:29:09 p.m.	15/10/2014 07:29:09 p.m.
MOSS-Web318-a24	MOSS_DELTAMPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSS-Web318-a24	MOSS_DELTAMPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 07:25:20 p.m.	15/10/2014 07:25:20 p.m.
MOSSAD-COL	DS_FULLSYNC	success	15/10/2014 07:25:20 p.m.	15/10/2014 07:25:20 p.m.
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 11:13:15 a.m.	15/10/2014 11:13:15 a.m.
MOSS-Web318-a24	MOSS_DELTAMPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 11:13:14 a.m.	15/10/2014 11:13:14 a.m.
MOSSAD-COL	DS_DELTASync	success	15/10/2014 11:13:12 a.m.	15/10/2014 11:13:12 a.m.
MOSSAD-COL	DS_DELTASync	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSSAD-COL	DS_DELTASync	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSS-Web318-a24	MOSS_DELTAMPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSSAD-COL	DS_DELTASync	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 10:28:48 a.m.	15/10/2014 10:28:48 a.m.
MOSS-Web318-a24	MOSS_DELTASync_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 10:28:47 a.m.	15/10/2014 10:28:47 a.m.

Name	Profile Name	Status	Start Time	End Time
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 07:29:14 p.m.	15/10/2014 07:29:38 p.m.
MOSSAD-COL	DS_FULLSYNC	success	15/10/2014 07:29:09 p.m.	15/10/2014 07:29:09 p.m.
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 07:25:20 p.m.	15/10/2014 07:25:20 p.m.
MOSSAD-COL	DS_FULLSYNC	success	15/10/2014 07:25:20 p.m.	15/10/2014 07:25:20 p.m.
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 11:13:15 a.m.	15/10/2014 11:13:15 a.m.
MOSSAD-COL	DS_DELTASync	success	15/10/2014 11:13:14 a.m.	15/10/2014 11:13:14 a.m.
MOSSAD-COL	DS_DELTASync	success	15/10/2014 11:13:12 a.m.	15/10/2014 11:13:12 a.m.
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSSAD-COL	DS_DELTASync	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSSAD-COL	DS_DELTASync	success	15/10/2014 11:12:51 a.m.	15/10/2014 11:12:51 a.m.
MOSS-Web318-a24	MOSS_EXPORT_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 10:28:48 a.m.	15/10/2014 10:28:48 a.m.
MOSS-Web318-a24	MOSS_DELTASync_46425c9-9425-499b-acd8-508a167ed72a	success	15/10/2014 10:28:47 a.m.	15/10/2014 10:28:47 a.m.

- Si nos damos cuenta nos muestra cuantos usuarios se adicionan y se actualizan.
- Si presionamos sobre update nos aparece el listado de usuarios actualizados.

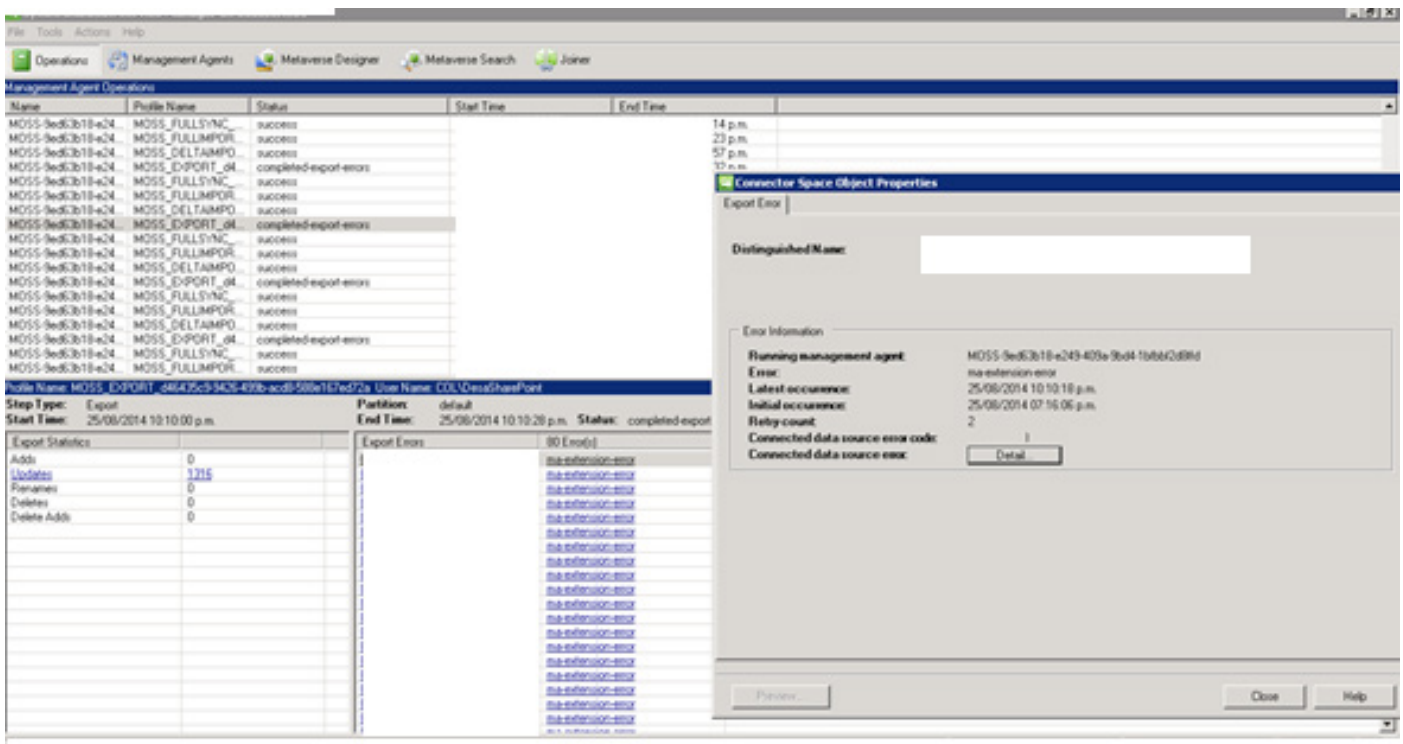


- Y termina la ejecución (tanto del FIM como de la sincronización de perfiles)





- En caso de error sale una pantalla como la siguiente, en la cual nos aparecen los usuarios con error y el motivo del error.



Espero que este artículo sea de mucha utilidad.

**ENRIQUE RHENALS BÁRCENAS**

MCP SharePoint 2010 y 2013

erhenalsb@hotmail.com

@enriquerhenals

<http://enriquerhenalsb.blogspot.com>

*Forefront Identity Manager en su versión básica (como la que tenemos con el miisclient.exe en SharePoint 2010) es usado para importar y exportar datos entre sistema de identidades.*

# 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.



# Replicación de granjas de SharePoint vía LogShipping

## Como establecemos Log Shipping

Para establecer la replicación por Log Shipping, es necesario realizar las siguientes acciones:

1. Eliminar la aplicación web de contenido configurada originalmente.
2. Apagar los servidores de SharePoint de la granja secundaria.
3. Acceder al SQL Manager desde la granja primaria.
4. Conectar el SQL Manager con la granja secundaria.
5. Para cada una de las bases de datos a replicar:
  - a. Pulsar con el botón derecho sobre la BBDD correspondiente, y seleccionar propiedades.
  - b. En el cuadro que se abre a continuación, seleccionar la última opción en la izquierda, denominada Transaction Log Shipping.
  - c. En la ventana, seleccionar "Configure this database as a primary database for a Log Shipping Configuration".

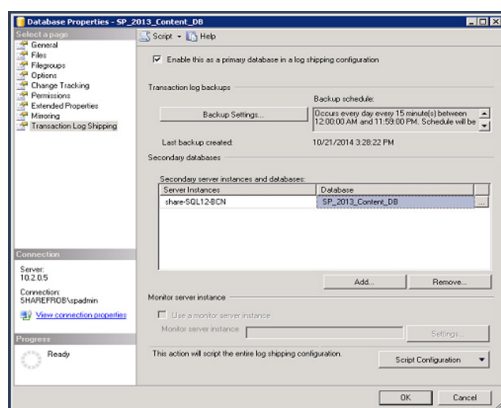


Imagen 1.- Ventana de configuración de Log.

A continuación, configurar las opciones de backup a utilizar. En este paso se recomienda crear, en la unidad de logs, tanto del servidor origen como en el servidor destino, un directorio compartido denominado "LogShipping"

- d. En la parte inferior, denominada "Secondary Databases", pulsar sobre el botón Add. Esto permitirá añadir contra la base de datos que va-

mos a replicar. Se recomienda, para las bases de datos de contenido borrar la base de datos de destino para comprobar que la replicación se ha llevado la última versión del origen. La ventana que aparecerá es la siguiente:

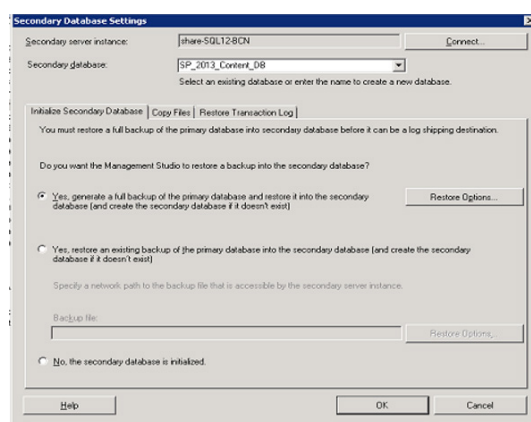


Imagen 2.- Configuración de la base de datos secundaria.

Seleccionar la primera opción. En la segunda pestaña, "Copy Files", seleccionar I:\LogShipping, según el nombre de directorio seleccionado.

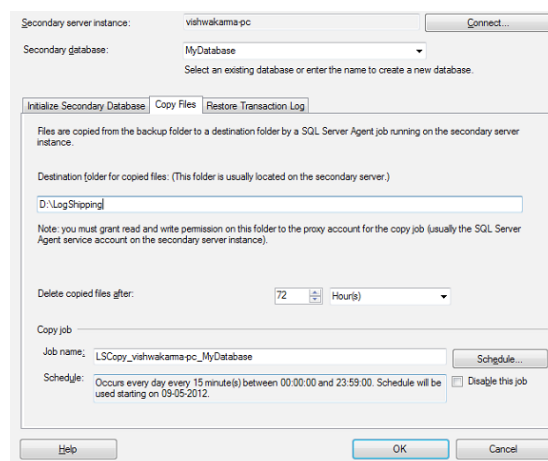


Imagen 3.- Seleccionamos la ruta destino de los ficheros.

En la tercera pestaña, denominada "Restore Transaction Log", seleccionar None. Esto dejará la base de datos destino permanentemente en modo restauración, por lo que recibirá las modificaciones efectuadas cada vez que se realice el backup de origen (de forma estándar, cada 15 minutos).

que el contenido mostrado es el mismo que en el origen.

Con esta última comprobación, queda demostrado que el log shipping está correctamente configurado. A continuación, es necesario para de nuevo los servidores de SharePoint, y volver a habilitar el backup de las base de datos de origen para que el log shipping quede funcionando. Las bases de datos deben aparecer en destino como Restoring.

## Procedimiento de Balanceo por Disaster Recovery

El procedimiento de Balanceo por Disaster Recovery será necesario en caso de fallo catastrófico de la granja principal. Este procedimiento consistirá en arrancar la granja secundaria, restaurando las bases de datos de contenidos y de aplicaciones de servicio. Para ello:

Para cada base de datos en estado Restoring:

1. En el servidor de origen, deshabilitar el job de replicación accediendo desde sql a Database -> Properties -> Log Shipping replication -> Backup Settings -> Disable this job .

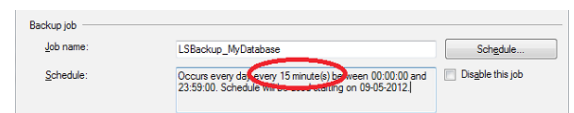


Imagen 5.- Deshabilitamos el job de replicación.

2. Lanzar la orden RESTORE DATABASE database WITH NO RECOVERY.
3. Arrancar los servidores de SharePoint-
4. Una vez estén todos los servidores levantados, acceder a la Administración Central.
5. [Solo para las aplicaciones web] acceder a administración de aplicaciones y bases de datos de servicio.
6. Seleccionar cada aplicación web a comprobar, y verificar que en la base de datos asociada (la cual debe ser la que hemos restaurado) aparecen correctamente al menos una colección de sitios.
7. En caso de que alguna base de datos no se haya montado correctamente, lanzar la siguiente orden en powershell:

```
Mount-SPContentDatabase "MyDatabase" -DatabaseServer "My-Server" -WebApplication http://sitename
```

El orden correcto de aplicación de este procedimiento es levantar primero las bases de datos de las aplicaciones de servicio, y posteriormente levantar las bases de datos de las aplicaciones web.

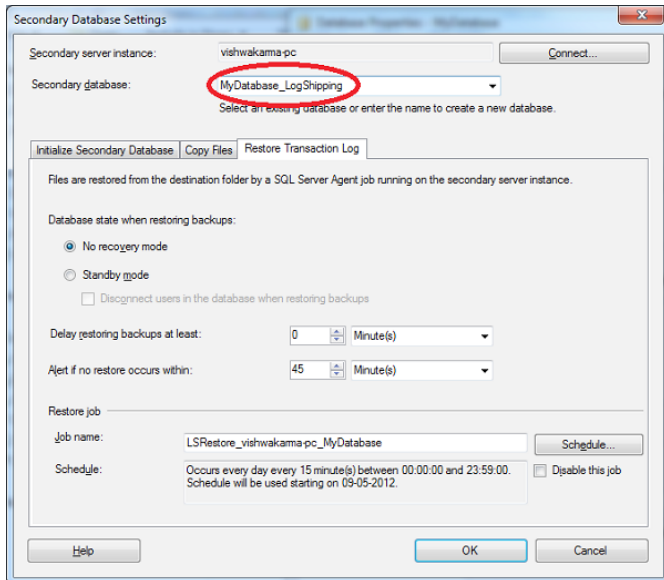


Imagen 4.- Establecemos la opción "No recovery mode".

6. Al pulsar aceptar, el sistema mostrará un cuadro de dialogo donde veremos cómo se hace un backup de origen, se exporta al servidor secundario, y se restaura, mostrando este la base de datos en estado Restoring.



7. Acceder a las bases de datos de origen, y para el job de backups (este paso es para evitar perdida de datos durante la configuración en destino).
8. Para cada una de las bases de datos que estén en estado restoring, lanzar el siguiente comando desde una ventana de comandos de SQL:

```
RESTORE DATABASE database WITH RECOVERY
```

## Pasos necesarios solo para la aplicación WEB (no para las bbdd de aplicaciones de servicio)

9. A continuación, levantar los servidores de SharePoint de la granja destino, y configurar una nueva aplicación web. En la especificación de la base de datos de contenido, introducir el nombre de la base de datos restaurada (en el ejemplo, SP\_2013\_Content\_DB) y no seleccionar nombre de host. En caso contrario, la aplicación web no reconocerá las colecciones de sitios ya configuradas en la base de datos exportada.
10. Tras la creación, en la opción de administración de aplicaciones -> Administrar bases de datos, debe aparecer la base de datos asociada con la nueva aplicación web, y las colecciones de sitios correctamente inventariadas.
11. Acceder a la nueva aplicación web y comprobar



*Aunque el establecimiento de una granja secundaria es un mecanismo de seguridad importante, en modo alguno esto puede sustituir a un adecuado dimensionamiento de la granja principal, conforme a las mejores prácticas, y a un adecuado mantenimiento según un plan de gobernanza adecuado.*

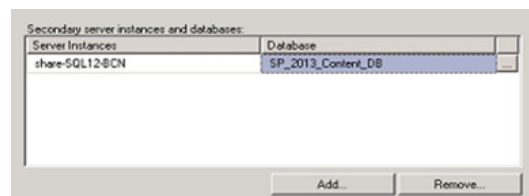


Imagen 6.- Bases de datos secundarias.

## Procedimiento de Actualización de la granja secundaria

Este procedimiento se deberá ejecutar cada vez que se requiera instalar nuevas versiones de las soluciones desplegadas en la granja. Para ello:

1. Aplicar el mecanismo de disaster recovery hasta tener la granja secundaria operativa.
2. Instalar las actualizaciones de las soluciones y comprobar que las aplicaciones web responden correctamente.
3. Una vez comprobada la correcta aplicación de las actualizaciones, apagar los servidores de la granja SP comenzando por los frontales.
4. Para cada base de datos que deba ser sincronizada:
  - a. Volver a activar el job de replicación en el SQL de origen.
  - b. Comprobar que las bases de datos se ponen en modo replicación (puede tardar unos minutos, hasta que se lance la siguiente replicación).
  - c. En caso de no ponerse en modo replicación, borrar la base de datos destino del logshipping y volver a añadirla.

## Procedimiento de Balanceo hacia la granja Principal

Una vez solucionado el motivo que nos obligó a enviar el tráfico de los usuarios hacia la granja secundaria, es necesario volver a balancear este tráfico hacia nuestra granja principal. Para garantizar que no se pierde trabajo de una granja a otra, mi recomendación es establecer de nuevo el log shipping desde la granja secundaria hacia la principal, de forma que todas las bases de datos en la granja principal sean actualizadas con los contenidos generados en el tiempo entre el desastre y su resolución. Una vez replicadas estas bases de datos, será posible volver a pasar el tráfico a la granja principal, y reestablecer el log shipping como estaba configurado originalmente.

### Nota Importante:

Aunque el establecimiento de una granja secundaria es un mecanismo de seguridad importante, en modo alguno esto puede sustituir a un adecuado dimensionamiento de la granja principal, conforme a las mejores prácticas, y a un adecuado mantenimiento según un plan de gobernanza adecuado. Para profundizar en estos contactos recomiendo leer los artículos de mis amigos Alberto Días, Juan Carlos Gonzalez y Miguel Tabera.

---

### FABIÁN CALVO

Team leader

[fcalvo@encamina.com](mailto:fcalvo@encamina.com)

[@fcvspan](#)

<http://www.encamina.com>

# Herramientas de Branding para desarrollar aplicaciones de SharePoint elegantes y potentes

A lo largo de este artículo hablaremos de Bootstrap como Framework para personalizar sitios de SharePoint Online y de SharePoint OnPremises, pero ¿Qué es Bootstrap? Bootstrap es un Framework muy completo que permite crear interfaces web con CSS y JavaScript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive Design.

Aun ofreciendo todas las posibilidades que ofrece Bootstrap a la hora de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos.

El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos.

Entre muchas de las posibilidades que ofrece Bootstrap podemos ver un claro ejemplo en la siguiente dirección: <http://www.ancadia.com/Pages/horizonscodes.aspx>. Citando un ejemplo, con el siguiente código podemos crear una tabla:

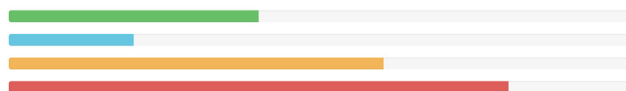
```
<div class="panel panel-default">
  <!-- Default panel contents -->
  <div class="panel-heading">Panel heading</div>
  <div class="panel-body">
    <p>...</p>
  </div>
  <!-- Table -->
  <table class="table">
    ...
  </table>
</div>
```

Panel heading			
Some default panel content here. Nulla vitae elit libero, a pharetra augue. Aenean lacinia bibendum nulla sed consectetur. Aenean eu leo quam. Pellentesque ornare venenatis vestibulum. Nullam id dolor id nibh ultricies vehicula ut id elit.			
#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

O bien con otro simple código podemos crear algo más complejo como las tan buscadas y utilizadas barras de progreso:

```
<div>
  class="progress">
  <div>
    class="progress-bar"
    role="progressbar"
    aria-valuenow="60"
    aria-valuemin="0"
    aria-valuemax="100"
    style="width: 60%;>
    <span class="sr-only">60% Complete</span>
  </div>
</div>
```

Y como resultado obtenemos:



Es muy fácil hacer diseños dinámicos dentro del Bootstrap ya que todo el código está predefinido. Hasta aquí todo bien para los diseñadores Web, **¿Pero qué tiene que ver esto con las Apps de SharePoint?**

***Bootstrap es un Framework muy completo que permite crear interfaces web con CSS y JavaScript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive Design.***

Como ya sabemos las Apps de SharePoint 2013 se basan al igual que las Web en HTML, CSS3 y JavaScript, la idea de integrar un Bootstrap dentro del desarrollo de Apps es genial, ya que nos adelantaría muchísimo el trabajo a nivel de interface y además podemos crear aplicaciones muchos más profesionales y dinámicas.

También es importante conocer que hay muchísimos desarrolladores trabajando para Bootstrap, la buena noticia es que podemos encontrarnos con ciertos desarrollos profesionales que podemos adaptar fácilmente a nuestra

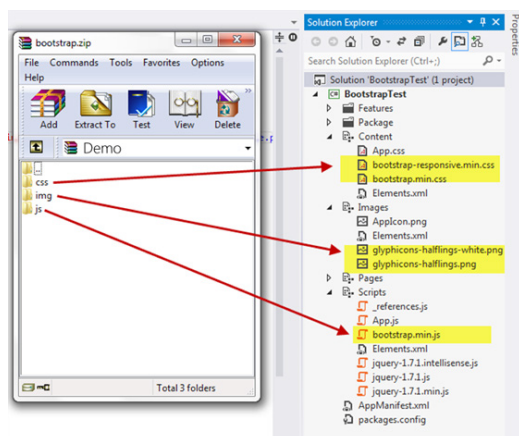
App una vez esté integrada. Es mucho más fácil integrar CSS y JavaScript a un Bootstrap que a una App o Web que se está desarrollando desde cero, esto es debido a que el Bootstrap ya trae una estructura claramente definida y muy simple.

La idea es muy buena pero ¿Cómo lo hacemos?

- El primer paso es crear nuestra App dentro de Visual Studio sin ningún código.
- Luego procederemos a descargar el Bootstrap que más nos convenga:

Versión	Descripción	Precio	Enlace de Descarga
Twitter Bootstrap	Es el <b>Bootstrap</b> convencional y original con las herramientas básicas de diseño.	Gratis	<a href="http://getBootstrap.com/">http://getBootstrap.com/</a>
Ancadia Horizons	Este Bootstrap ya está adaptado completamente a SharePoint y además incluye herramientas de diseños para una apariencia más profesional de nuestra app como: Parallax, Modern UI y Flat Design.	\$75	<a href="http://www.ancadia.com/horizons">http://www.ancadia.com/horizons</a>

- Una vez descargado Bootstrap procedemos a descomprimirlo en nuestro disco.
- Copiar el contenido de la carpeta CSS a la carpeta "Content" de la aplicación que ha creado en Visual Studio. Es importante copiar solo los ficheros "-min."
- Copiamos el contenido de la carpeta \IMG del bootstrap a la carpeta "Images" de la aplicación.
- Para finalizar copiamos la carpeta \JS del Bootstrap a la carpeta "scripts" de la aplicación.



De esta forma, hemos vinculado los ficheros copiados a nuestra aplicación.

- Hasta ahora hemos visto cómo y dónde hay que copiar los ficheros necesarios para ejecutar el Bootstrap dentro de nuestra App, ahora debemos hacer funcionar esos ficheros. Lo primero que de-

bemos hacer es agregar la siguiente referencia del Bootstrap debajo de la siguiente etiqueta: "PlaceHolderAdditionalPageHead" dentro de nuestra App:

```
<link rel="Stylesheet" type="text/css" href="..\Content/Bootstrap-responsive.min.css" />
<link rel="Stylesheet" type="text/css" href="..\Content/Bootstrap.min.css" />
<script type="text/javascript" src="..\Scripts/Bootstrap.min.js"></script>
```

## Retoques finales

Bootstrap utiliza dos archivos de imágenes que definen los iconos. Estos archivos de imagen fueron copiados en el paso anterior a la App justo en la carpeta \img. Como ya sabemos a diferencia del Bootstrap, SharePoint guarda las imágenes de la App en la carpeta "\Images":

- Crearemos una carpeta nueva llamada "\Images" haciendo uso del Explorador de Soluciones en Visual Studio.
- Editamos el archivo "Bootstrap.min.css" que se encuentra dentro de nuestra App (específicamente en la carpeta "Content")
- Luego hacemos una búsqueda con la palabra clave "IMG" dentro del fichero, a continuación encontraremos la referencia que necesitamos modificar y simplemente cambiamos la ruta \img por \images.

```
..background-image:url("../img/glyphicons-halflings.png") ... Cambiar a ->
..background-image:url("../images/imagen.png")...
```

## Conclusión

Existen varios Frameworks costosos y complejos en el mercado como por ejemplo KENDO UI, que nos brindan funciones muy atractivas como el poder hacer gráficas y conexiones a la base datos. Pero con esta solución podemos obtener casi el 90% de esas funciones brindadas por estos Frameworks de una manera gratuita o muy bajo coste sin la necesidad de volver aprender a programar en nuevos entornos. Adicionalmente al ser un sistema abierto nos permitirá integrar códigos de terceros que pueden potenciar nuestra aplicación.

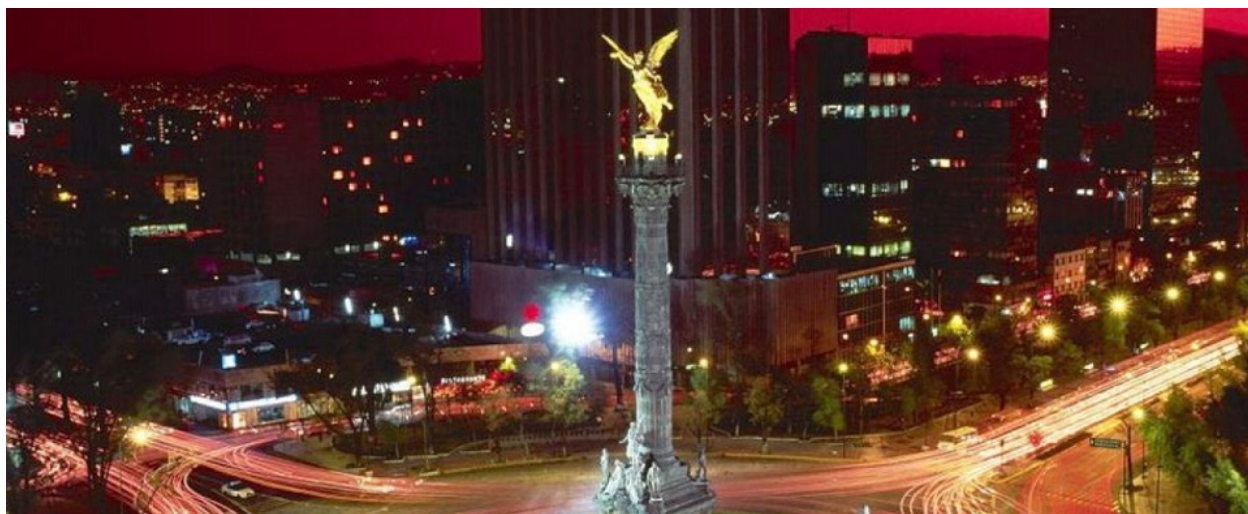
**DARWIN CASTRO MARÍN**

Microsoft Certified Solutions Expert

[darwin.castro@ancadia.com](mailto:darwin.castro@ancadia.com)

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

<http://www.ancadia.com>



# SharePoint Saturday Mexico 2015

**Fecha:** Sábado 31 de Enero de 2015

**Lugar:** TelmexHub – Calle Isabel La Católica #51, Cuauhtemoc, Ciudad de México, México.

**Horario:** 9 am a 7:30 pm

Te invitamos al primer SharePoint Saturday en la Ciudad de México, un evento gratuito en el que podrás disfrutar de sesiones técnicas impartidas por influenciadores, MVPs, líderes técnicos y respetados profesionales en el ámbito; así como tener la oportunidad de hacer networking con gente relacionada en las tecnologías SharePoint. Tendremos sesiones para IT Pros y Developers.

Regístrate en: <http://www.spsevents.org/city/mexicocity/mexicocity2015>

Invitan:

Comunidad SharePoint México





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

Desde un punto de vista puramente conceptual, un Grupo se define como un espacio de trabajo compartido para e-mail, conversaciones, archivos y eventos de calendario en el que los miembros de un grupo pueden colaborar de forma rápida, productiva y sencilla a través de una interfaz de usuario simple y que permite que el usuario esté en todo momento centrado en la actividad que tiene lugar en el Grupo.

Aunque el anuncio de la disponibilidad de los Grupos de Office 365 se realizó el pasado mes de septiembre, la disponibilidad de los mismos no es global a día de hoy en Office 365, sino que se está liberando de forma gradual empezando por aquellos tenants de Office 365 en el que se ha habilitado la característica de “First Release” que permite disponer de las nuevas funcionalidades y novedades liberadas en Office 365 antes de que estén disponibles de forma global para todos los tenants de Office 365.

## Creación de un Grupo de Office 365

La característica de Grupos de Office 365 está disponible, por el momento, a través del OWA de cada usuario de Office 365. Para poder crear un Grupo de Office 365 desde el OWA (**Nota:** Microsoft está liberando de forma incremental una serie de comandos PowerShell para poder trabajar con Grupos de Office 365 por medio de la línea de comandos), basta con realizar los siguientes pasos:

- Accedemos al OWA de Office 365 y localizamos en el menú vertical la sección Grupos desde la que podemos ver un listado de Grupos en los que estamos participando bien porque los hemos creado nosotros o porque nos hemos suscrito a los mismos. Desde aquí podremos crear un nuevo Grupo haciendo clic en el botón “+”.

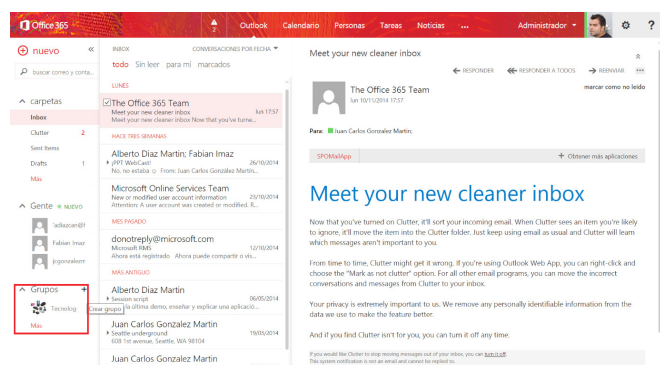


Imagen 1.- Acceso a la funcionalidad de Grupos de Office 365.

- La creación del Grupo se realiza a través de un panel que se habilita en el mismo momento en el que hacemos clic en la opción de Crear grupo. En este panel tendremos que completar la información relativa al nombre del grupo, el identificador del mismo, su descripción y si se trata de un grupo público o privado. Una vez estemos listos, pulsamos “CREAR” para que se proceda a crear el nuevo Grupo de Office 365.

CREAR

DESCARTAR

Crear un grupo

Un grupo proporciona un espacio para un calendario de grupo, conversaciones y documentos compartidos.

Elija un nombre:

Id. de grupo:

Disponible

Agregue una descripción:
 

Grupo colaborativo de CompartiMOSS, la revista líder de habla hispana sobre SharePoint y Office 365

Privacidad:
 

Público: cualquiera puede ver el contenido

☒ Suscriba a miembros para que reciban las conversaciones de grupo en sus bandejas de entrada (Los miembros pueden cancelar su suscripción)

Imagen 2.- Panel de creación de un nuevo Grupo de Office 365.

- Cuando el Grupo de Office 365 está listo, podremos empezar añadir integrantes al mismo. Por el momento, sólo se pueden añadir como miembros a usuarios de nuestro tenant de Office 365, no siendo posible añadir usuarios de otros tenants de Office 365 o usuarios externos. Para añadir los usuarios al Grupo, basta con pulsar el botón “AGREGAR”.

AGREGAR

AHORANO

Agregar miembros

Agregue contactos al grupo. Todas ellas recibirán un correo electrónico en el que se les informará del grupo nuevo.

Agregar miembros:

NuberosDummy Dummy

nuberosnetdm@NuberosNET.onmicrosoft.ca

NuberosNet WebMaster

nuberosnetwm@NuberosNET.onmicrosoft.ca

Imagen 3.- Panel de adición de miembros al Grupo creado.

Como resultado de añadir los integrantes al Grupo, se generan una serie de correos electrónicos de notificación a cada usuario indicando que forma parte del Grupo y además proporcionando una serie de instrucciones para empezar a colaborar en el mismo.

## Bienvenido al grupo "CompartiMOSS"

Juan Carlos Gonzalez Martin le ha agregado al grupo "CompartiMOSS"

Grupo colaborativo de CompartiMOSS, la revista líder de habla hispana sobre SharePoint y Office 365



**Iniciar una conversación**  
Correo electrónico compartimoss@NuberosNET.onmicrosoft.com



**Ver archivos del grupo**  
Organice archivos en una ubicación para compartir y participar fácilmente.

Público: todas las personas de su organización pueden ver todo el contenido del grupo.

**Office 365** Participar con el grupo de Office 365  
[Examinar las conversaciones](#) [Ver calendario](#)

Imagen 4.- Correo tipo de bienvenida enviado a los miembros del Grupo

- Con estas configuraciones iniciales del se muestra la página principal del Grupo desde la que podremos comenzar a crear conversaciones, personalizar el Grupo (añadiendo un logotipo identificativo) o bien acceder a las funciones de administración del Grupo.

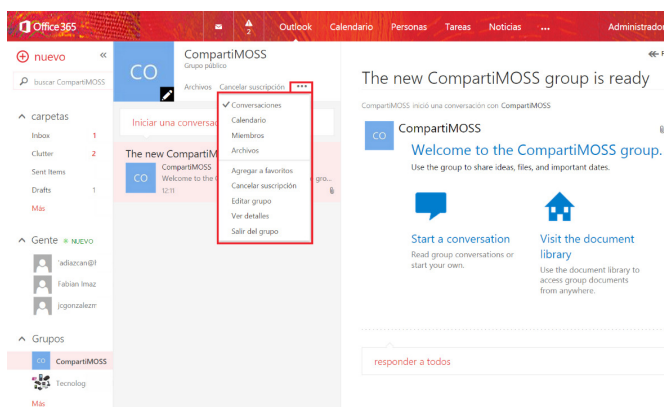


Imagen 5.- Página de bienvenida de un Grupo y opciones de administración.

## Trabajo colaborativo en un Grupo de Office 365

El trabajo colaborativo en un Grupo de Office 365 se basa en los siguientes elementos:

- Creación de conversaciones en el mismo de forma directa o bien a través del correo electrónico asociado al Grupo.
- Creando eventos en el Calendario del Grupo, disponible desde el momento en el que el Grupo es creado.
- O bien compartiendo información en el espacio colaborativo asociado al Grupo que es una Colección de

**Desde un punto de vista puramente conceptual, un Grupo se define como un espacio de trabajo compartido para e-mail, conversaciones, archivos y eventos de calendario en el que los miembros de un grupo pueden colaborar de forma rápida, productiva y sencilla.**

Desde la perspectiva conversacional, la experiencia de creación de un diálogo es la misma que podemos encontrar en cualquier plataforma o herramienta social del mercado como Yammer, LinkedIn, etc. En este sentido, cada Grupo cuenta con un muro donde se pueden ver las conversaciones que están teniendo lugar y desde el que es posible añadir nuevas conversaciones o bien realizar comentarios en conversaciones existentes:

- La respuesta a la conversación se puede realizar directamente desde el propio Grupo o bien a través del correo de notificación que cada integrante del Grupo recibe desde el mismo momento en el que se crea una conversación.



Imagen 6.- Ejemplo de conversación en un Grupo y experiencia de usuario en la respuesta.

- Por supuesto, en estas conversaciones se puede agregar no solo texto, sino que también se puede adjuntar archivos a la misma.

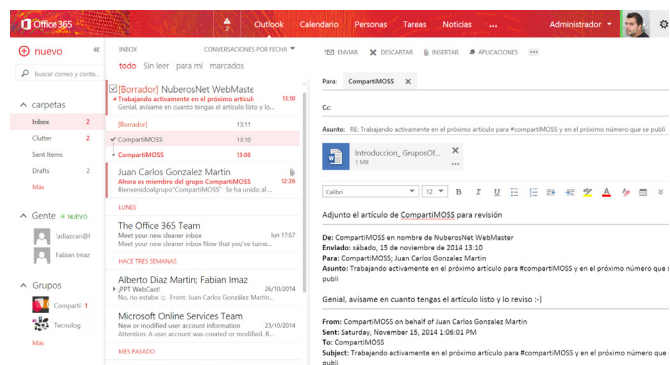


Imagen 6.- Ejemplo de respuesta a una conversación del Grupo vía correo electrónico.

- Y como se puede apreciar en la Imagen 6, en la bandeja de entrada se reciben todos los mensajes relativos a las conversaciones del Grupo. Estos mensajes se pueden eliminar de la bandeja sin que sean bo-

rrados del Grupo como sucede con cualquiera de las plataformas sociales del mercado.

## Calendario de Grupo

Los Grupos de Office 365 incorporan un calendario de Grupo en el que se pueden ir programando los distintos eventos relacionados con la actividad del mismo. Para acceder al calendario:

- Desde las opciones de configuración del Grupo (parte superior del muro), hacemos clic en la opción “Calendario”.

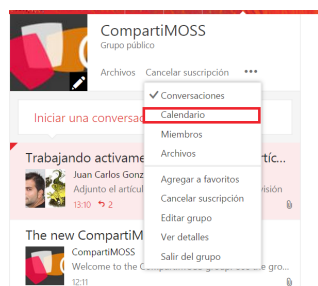


Imagen 7.- Acceso al calendario del Grupo.

- A continuación se muestra el calendario del Grupo de Office 365 en el que cada integrante del Grupo puede añadir nuevas citas, solapar dicho calendario con su calendario personal, etc.

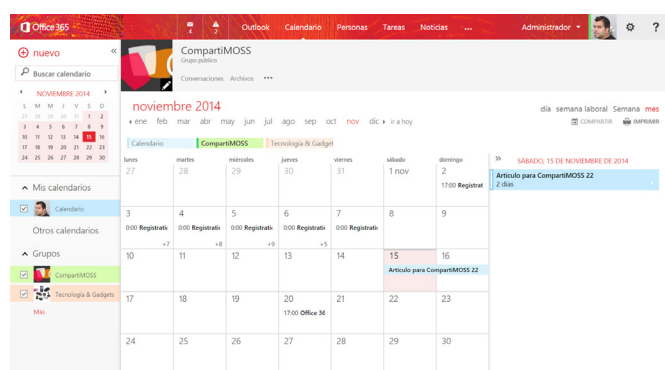


Imagen 8.- Calendario del Grupo.

## Archivos

De la misma forma que un Grupo de Office 365 dispone de un Calendario, cada Grupo cuenta con un espacio de

documentación que es una colección de sitios que cuenta con una biblioteca de documentos en la que se puede ir cargando documentación del grupo. Precisamente es esta característica la que a día de hoy tiene mucho margen de mejora no solo desde el punto de vista de experiencia de usuario, sino también de gobernanza y control. Para acceder al sitio asociado al Grupo, basta con hacer clic en la opción “Archivos” disponible en el menú de opciones del Grupo

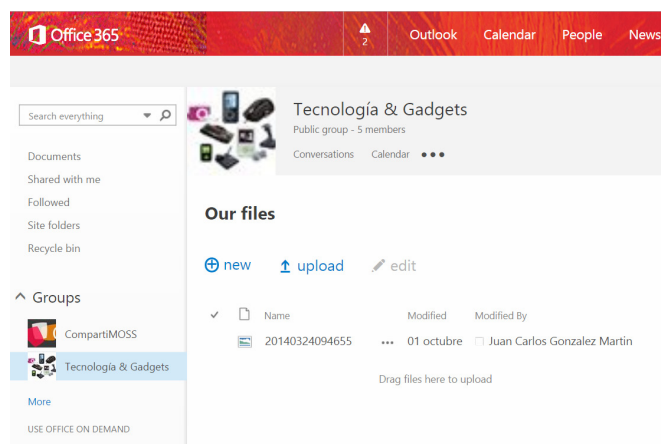


Imagen 9.- Ejemplo de sitio asociado a un Grupo de Office 365.

## Conclusiones

Siguiendo con su estrategia de aportar valor de forma continua a Office 365, Microsoft liberó el pasado mes de septiembre una nueva característica denominada Grupos destinada a facilitar el trabajo colaborativo en torno a una misma temática desde el OWA y proporcionando elementos de valor añadido como el Calendario de Grupo, el Sitio de Grupo y la integración a nivel de mensajería con Exchange Online.

**JUAN CARLOS GONZÁLEZ MARTÍN**

MVP SharePoint de SharePoint Server / Arquitecto de Soluciones en LKS

E-Mail de Contacto: [jc.gonzalez@lks.es](mailto:jc.gonzalez@lks.es)

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

Blog: <http://geeks.ms/blogs/jcgonzalez>

Web: [www.lks.es](http://www.lks.es)

# La nube de SharePoint On-Premises

Actualmente tenemos dos versiones de SharePoint, SharePoint 2013 y SharePoint Online, con algunas funcionalidades de diferencia, bien porque nos encontramos con ciertos límites del entorno de ejecución en la nube, bien porque se han desarrollado nuevas funcionalidades que no han llegado a la versión On-Premises.

## ¿Por qué tener SharePoint 2013 en la nube?

A estas alturas, todos tenemos claros los ahorros de coste que supone usar la nube como servicio o a modo de infraestructura, eso sí, desde el punto de vista de SharePoint, hay que tener claro que el modo servicio, SharePoint Online, limita su extensibilidad y las personalizaciones, extensiones o desarrollos actuales no son compatibles. Si hablamos de control, coste y eficiencia, tenemos:

- SharePoint On-Premises nos obliga a administrar y gestionar el hardware/software
- SharePoint IaaS, nos abstrae del hardware y nos permite centrarnos en el software, manteniendo el acceso a todas las opciones de desarrollo y despliegue
- SharePoint Online (SaaS) nos abstrae tanto del hardware como del software, a cambio de un entorno limitado de extensibilidad.

Nuestra primera opción debería de ser SharePoint Online pero por el camino nos encontraremos opciones en la que nos debemos de plantear un modelo híbrido, bien con SharePoint totalmente On-Premises o a modo de IaaS, por ejemplo, en Azure.

## Plataforma de desarrollo y pruebas

SharePoint tiene fama de monstruo de las galletas y se come todos los recursos de hardware que se ponen a su disposición. Si desarrollamos soluciones para SharePoint necesitas tener un SharePoint instalado por cada desarrollador, lo que nos lleva a tener un mínimo de 16Gb de RAM por máquina virtual.

Un entorno de pruebas o Pre-Producción, debería de ser un entorno similar al de Producción, lo que nos podría llevar a 4, 5 o 6 servidores.

¿Tenemos infraestructura suficiente dar soporte a estas necesidades? ¿Cuántas horas de pruebas necesitamos?

Por qué no llevarnos estos entornos a la nube y aprovecharnos de un coste reducido cuando no estén en uso, facilidad en el aprovisionamiento y mayor seguridad al no tener a nuestros partners desarrollando en nuestro data-center.

## Plataforma de intranet y colaboración

Cuando SharePoint Online no cumple con nuestros requisitos de negocio, no tenemos otra que usar la versión On-Premises. Plantearse tener esta misma versión en modo IaaS, soportado por Microsoft en Azure, es más que viable y supone ahorros de coste y de administración, además de agregar un nivel adecuado de seguridad.

*Actualmente tenemos dos versiones de SharePoint, SharePoint 2013 y SharePoint Online, con algunas funcionalidades de diferencia, bien porque nos encontramos con ciertos límites del entorno de ejecución en la nube, bien porque se han desarrollado nuevas funcionalidades que no han llegado a la versión On-Premises.*

## Recuperación ante desastres de una granja On-Premises en Azure

Azure tiene todas las capacidades para convertirse en nuestro datacenter secundario en una zona geográfica segura. Dependiendo del modelo de recuperación que necesitamos, mantendremos los costes controlados en función de los recursos que estemos consumiendo. Muy a tener en cuenta que, en caso de desastre, la elasticidad de la nube es un punto a favor para garantizar la respuesta ante el aumento de la carga de trabajo.



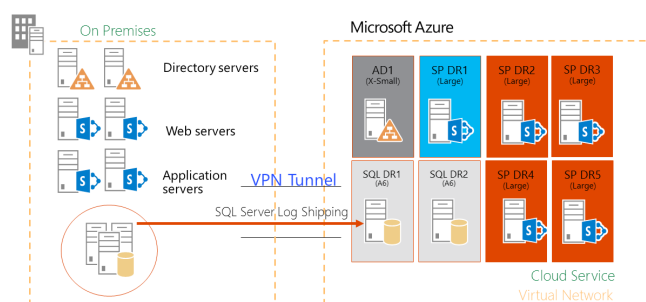


Imagen 1.- Diagrama de recuperación ante desastre en Azure.

Además, uno de los últimos servicios disponibles en Azure permite mantener una infraestructura de recuperación de sitios o máquinas virtuales desde On-Premises a Azure.

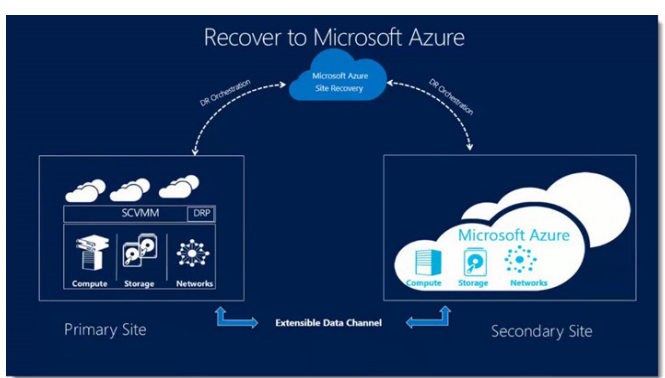


Imagen 2.- Recuperación ante desastre usando Azure.

Desde el punto de vista de SharePoint, es un modelo más sencillo y administrable de mantener las réplicas de nuestro datacenter principal en Azure, o de máquinas o servicios sencillos, se encuentren donde se encuentren.

## Sitios públicos de SharePoint en Azure

SharePoint es una de las principales plataformas de pu-

blicación de contenido web, pero SharePoint Online no cubre todas las capacidades que tenemos en SharePoint On-Premises. Si nuestro sitio web necesita extensibilidad o escalabilidad, SharePoint Online no cubre nuestras necesidades y la opción la encontramos en SharePoint On-Premises.

Utilizar la infraestructura de Azure para tener los servidores públicos de la organización, nos permite aprovecharnos de la seguridad de los datacenters de Microsoft y la elasticidad que nos permite es escalado de la infraestructura. Todo esto para centrarnos en construir un buen sitio y no gastar recursos en construir una buena infraestructura.

## Conclusiones

El principal reto que nos encontraremos para subirnos a la nube es el cálculo del coste, dependiendo de los servicios que usemos, tendremos una serie de variables que compondrán el gasto mensual que tendremos que pagar. Teniendo claro cómo podemos estimar estos costes, por ejemplo, mediante el cálculo de la telemetría de los servicios que estamos usando actualmente en nuestro datacenter, se nos abre un camino directo hacia Azure donde encontraremos infinidad de ventajas a nivel de costes y mantenimiento. Teniendo claro que la opción más clara es caminar hacia SharePoint Online, no siempre es la más adecuada ante las necesidades de los usuarios pero no hay que perder de vista que no siendo en Online, SharePoint 2013 gana ventajas en Azure.

**ALBERTO DÍAZ MARTÍN**

MVP SharePoint

[adiazcan@hotmail.com](mailto:adiazcan@hotmail.com)

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

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



## 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](http://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](http://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](mailto:adiazcan@hotmail.com)

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

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



## Fabián Imaz

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

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

Email: [fabiani@siderys.com.uy](mailto:fabiani@siderys.com.uy)

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

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



## Gustavo Velez

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

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

Email: [gustavo@gavd.net](mailto:gustavo@gavd.net)

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



## Juan Carlos González Martín

Juan Carlos González, es Ingeniero de Telecomunicaciones por la Universidad de Valladolid y Diplomado en Ciencias Empresariales por la Universidad Oberta de Catalunya (UOC). Cuenta con más 10 años de experiencia en tecnologías y plataformas de Microsoft diversas (SQL Server, Visual Studio, .NET Framework, etc.), aunque su trabajo diario gira en torno a las plataformas SharePoint y Office 365. Juan Carlos es MVP de SharePoint Server desde el año 2008, coordinador del grupo de usuarios .NET de Cantabria (Nuberos.Net, [www.nuberos.es](http://www.nuberos.es)) y co-fundador del Grupo de Usuarios de SharePoint de España (SUGES, [www.suges.es](http://www.suges.es)), del Grupo de Usuarios de Cloud Computing de España (CLOUDES) y de la Comunidad de Office 365. Desde el año 2011 participa junto con Gustavo Vélez, Fabián Imaz y Alberto Díaz en la dirección de CompartiMOSS. Hasta la fecha, ha publicado cuatro libros sobre SharePoint y varios artículos en castellano y en inglés sobre la plataforma.

Email: [jcgonzalezmartin1978@hotmail.com](mailto:jcgonzalezmartin1978@hotmail.com)

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

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

# ¿Desea colaborar con CompartiMOSS?



La subsistencia del magazine depende de los aportes en contenido de todos. Por ser una revista dedicada a información sobre SharePoint en español, todo el contenido deberá ser directamente relacionado con Microsoft SharePoint y escrito en castellano. No hay limitaciones sobre el tipo de artículo o contenido, lo mismo que sobre el tipo de versión. Si desea publicar algo, por favor, utilice uno de los siguientes formatos:

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

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

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

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

[revista@compartimoss.com](mailto:revista@compartimoss.com)

[adiazcan@hotmail.com](mailto:adiazcan@hotmail.com)

[fabiani@siderys.com.uy](mailto:fabiani@siderys.com.uy)

[jcgonzalezmartin1978@hotmail.com](mailto:jcgonzalezmartin1978@hotmail.com)

[gustavo@gavd.net](mailto:gustavo@gavd.net)



