



MadPoint  
SharePoint Technical  
Community Madrid

#content #people #SharePoint #apps #develop  
#search #Office 365 #cloud #farm #communities  
#sites #BI #ECM #insights  
#business #Office #workflows #IT  
#Welcome

Colaboran:



Plain Concepts



**spenta**



Microsoft



**Beezy**  
Social at work



#MadPoint

# Sobre MadPoint

 #MadPoint





LinkedIn

<http://www.linkedin.com/groups?gid=4554702>



Twitter

@MadPoint



Web

[www.madpoint.org.es](http://www.madpoint.org.es)

 #MadPoint



Mario Cortés Flores

mariocortesflores@hotmail.com  
@mariocortesf



Miguel Tabera Pacheco

miguel.tabera@outlook.com  
@migueltabera



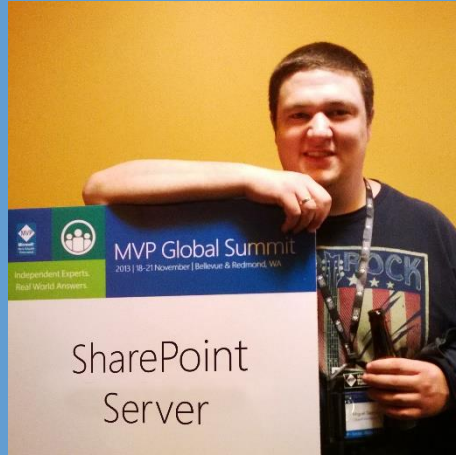
Christian Ruiz López

@\_christian\_ruiz



# Hoy veremos...

- Introducción a las API cliente de SharePoint (C#, JavaScript y REST)
- Introducción a las apps de Office y SharePoint
  - Tipos de app
  - Beneficios por usar apps
- Conceptos básicos al desarrollar apps
- La tienda de apps de SharePoint
- Taller: crea una app de SharePoint y publícala en la tienda
  - Andrés Lorbada
- Taller: creación de una app de Office 365 y Azure
  - Christian Ruiz



# Miguel Tabera Pacheco

## SharePoint Server MVP

### Plain Concepts

 [www.sinsharepointnohayparaíso.com](http://www.sinsharepointnohayparaíso.com)

 @migueltabera

# Prepara tus apps de SharePoint y Office 365 para el futuro



# Las nuevas API cliente de SharePoint 2013

```
8 // This function is executed after the DOM is ready
9 // Place any code you want to run when Default.aspx
10 // The code creates a context object which is needed
11 function SharePointReady() {
12     context = new SP.ClientContext.get_current();
13     web = context.get_web();
14     }
15     getUsername();
16 // This function prepares, loads, and then executes a SharePoint
17 function getUsername() {
18     user = web.get_currentUser();
19     context.load(user);
20     context.executeQueryAsync(onGetUserNameSuccess, onGetUserNameFail);
21 // This function is executed if the above OM call is successful
22 // replaces the content of the 'welcome' element with the user name
23 onGetUserNameSuccess() {
24     message'.text('Hello ' + user.get_title());
25 // This function is executed if the above OM call fails
26 onGetUserNameFail(sender, args) {
27     message'.text('Error: ' + args.get_message());
28     }
29 }
```



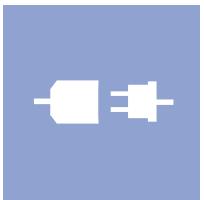
# Las tres API cliente



- Client-side object model (CSOM)
  - Para aplicaciones que usen C#.
  - Para aplicaciones Silverlight y Windows Phone.



- JavaScript object model (JSOM)
  - Muy sencillo para aplicaciones web y JavaScript de Windows 8.
  - Útil para desarrollar apps de SharePoint para otras plataformas (Android, iOS...)

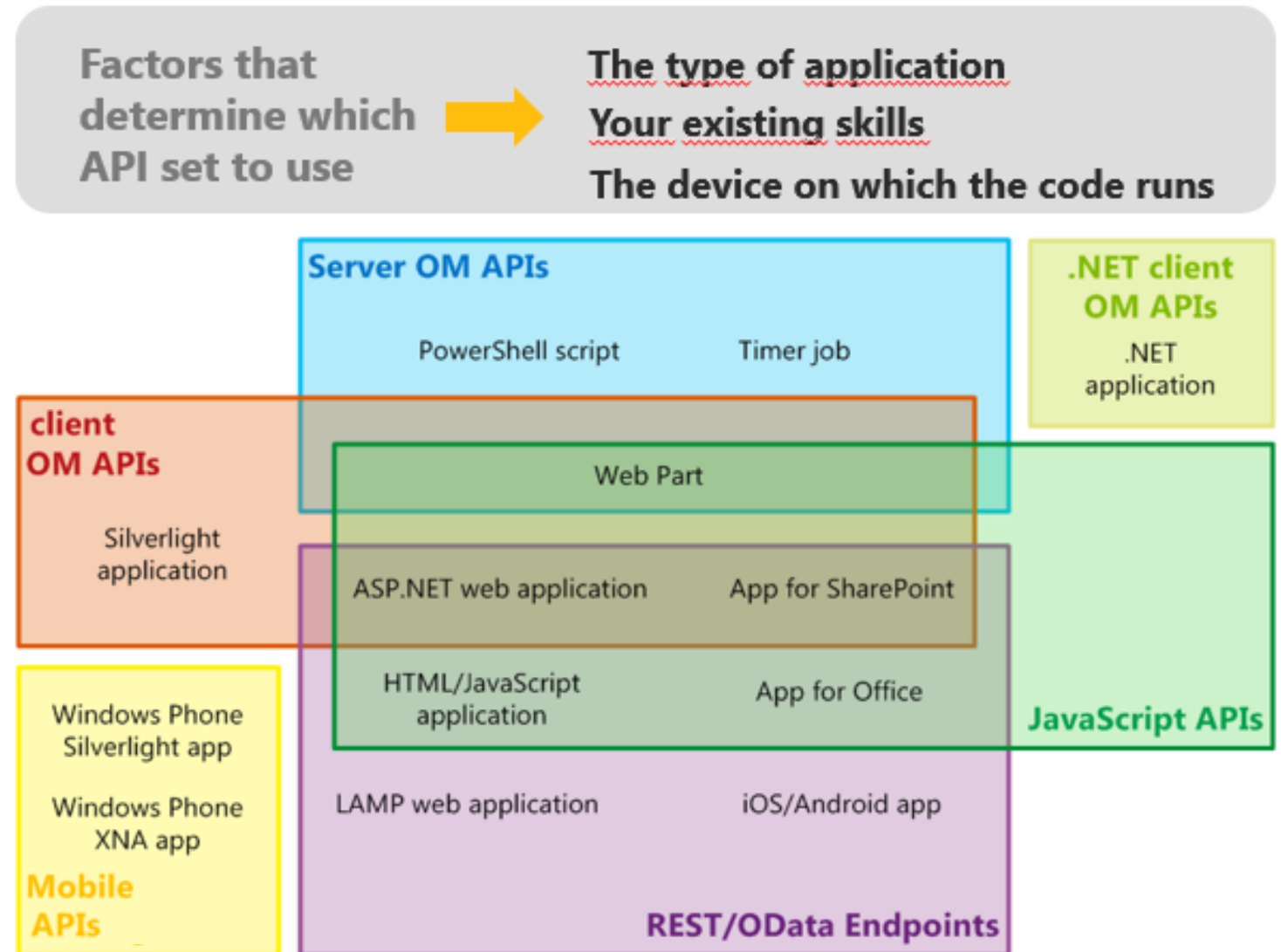


- REST
  - API independiente de la plataforma basada en estándares como OAuth y OData.



# Resto de APIs

- Combinando las tres API cliente con el resto de API se pueden desarrollar las distintas soluciones





# Ayuda y referencias de las API

- Reference for SharePoint 2013 APIs (Server, CSOM, JavaScript)
  - <http://msdn.microsoft.com/en-us/library/jj193038.aspx>
- How to: Complete basic operations using SharePoint 2013 **client library** code
  - <http://msdn.microsoft.com/en-us/library/fp179912.aspx>
- How to: Complete basic operations using **JavaScript library** code in SP 2013
  - <http://msdn.microsoft.com/en-us/library/jj163201.aspx>
- How to: Complete basic operations using SharePoint 2013 **REST endpoints**
  - <http://msdn.microsoft.com/en-us/library/jj164022.aspx>



# Trabajando con C#

```
// Starting with ClientContext, the constructor requires a URL to the
// server running SharePoint.
ClientContext context = new ClientContext("http://SiteUrl");

// Assume the web has a list named "Announcements".
List announcementsList = context.Web.Lists.GetByTitle("Announcements");

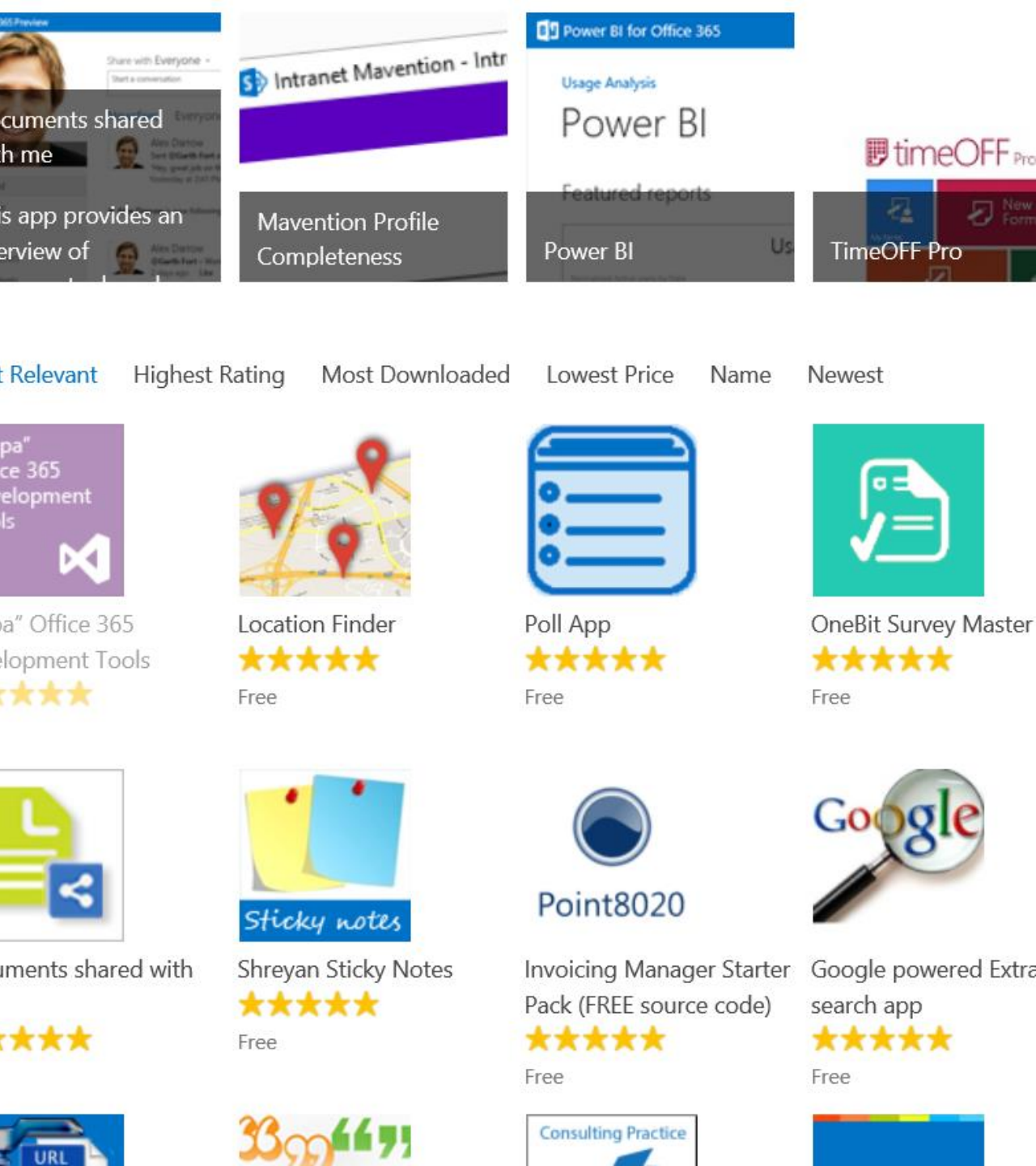
// This creates a CamlQuery that has a RowLimit of 100, and also specifies Scope="RecursiveAll"
// so that it grabs all list items, regardless of the folder they are in.
CamlQuery query = CamlQuery.CreateAllItemsQuery(100);
ListItemCollection items = announcementsList.GetItems(query);

// Retrieve all items in the ListItemCollection from List.GetItems(Query).
context.Load(items);
context.ExecuteQuery();
foreach (ListItem listItem in items)
{
    // We have all the list item data. For example, Title.
    label1.Text = label1.Text + ", " + listItem["Title"];
}
```



# Trabajando con JavaScript

```
function createListItem(siteUrl) {  
    var clientContext = new SP.ClientContext(siteUrl);  
    var oList = clientContext.get_web().get_lists().getByTitle('Announcements');  
  
    var itemCreateInfo = new SP.ListItemCreationInformation();  
    this.oListItem = oList.addItem(itemCreateInfo);  
    oListItem.set_item('Title', 'My New Item!');  
    oListItem.set_item('Body', 'Hello World!');  
    oListItem.update();  
  
    clientContext.load(oListItem);  
    clientContext.executeQueryAsync(  
        Function.createDelegate(this, this.onQuerySucceeded),  
        Function.createDelegate(this, this.onQueryFailed)  
    );  
}  
  
function onQuerySucceeded() {  
    alert('Item created: ' + oListItem.get_id());  
}  
  
function onQueryFailed(sender, args) {  
    alert('Request failed. ' + args.get_message() + '\n' + args.get_stackTrace());  
}
```



# Introducción a las apps de SharePoint





# ¿Qué es una app de SharePoint


- Aplicaciones independientes que se integran con SharePoint mediante APIs cliente (CSOM, JavaScript, REST)
- Diseñadas para no ensuciar ni cargar la granja de SharePoint
- El código no se ejecuta en el servidor de SharePoint
- Desaparición paulatina del modelo tradicional de desarrollo
- Código cliente
- Centradas en escenarios concretos
  - Petición de vacaciones
  - Registro a eventos
  - ...




# Las app de integran en SharePoint


## Site Contents

Lists, Libraries, and other Apps


 add an app




Announcements  
0 items  
Modified 4 hours ago




Documents  
0 items  
Modified 4 hours ago



Location Finder  
**new!**



Site Assets  
1 item  
Modified 4 hours ago



Site Pages  
2 items  
Modified 4 hours ago

Office 365 Outlook Calendar People Newsfeed SkyDrive Sites ... Admin Miguel Tabera Pacheco

BROWSE PAGE

test EDIT LINKS

test

Home

Notebook

Documents

Recent

Location Finder

Announcements

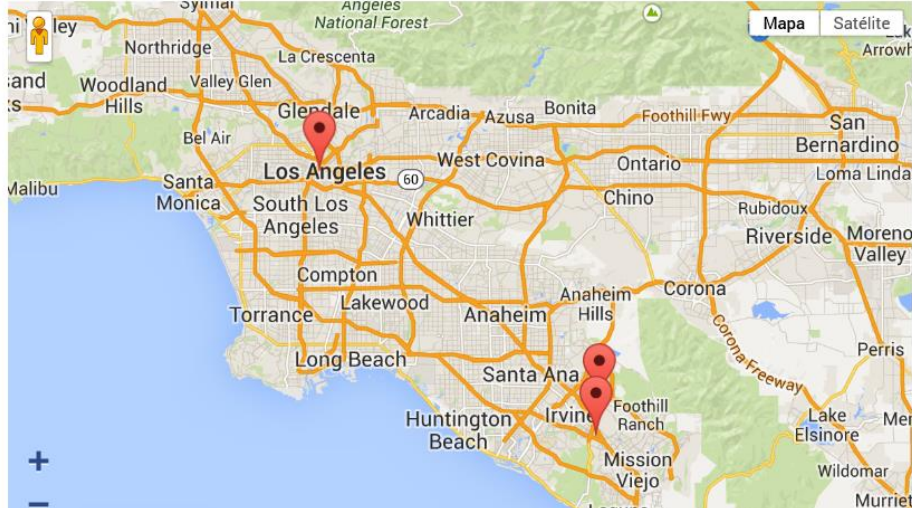
Tasks

Calendar

Site Contents

EDIT LINKS

### Location Finder



Documents

+ new document or drag files here

✓	Name	
✓	Venta	...
✓	Cuentas	...




# Cambio filosófico


- Suponen un cambio filosófico en SharePoint.
- Ahora todo son apps
- Pensar en las app de móvil: Funcionalidades concretas bastante independientes, pero que permiten acceder a características e información del teléfono solicitando autorización al usuario

Lists, Libraries, and other Apps


SITE WORKFLOWS




add an app




Calendar  
0 items  
Modified 1 minute ago




MicroFeed  
2 items  
Modified 1 minute ago




Site Pages  
2 items  
Modified 1 minute ago




Announcements  
0 items  
Modified 1 minute ago



Documents  
0 items  
Modified 1 minute ago



Site Assets  
1 item  
Modified 1 minute ago



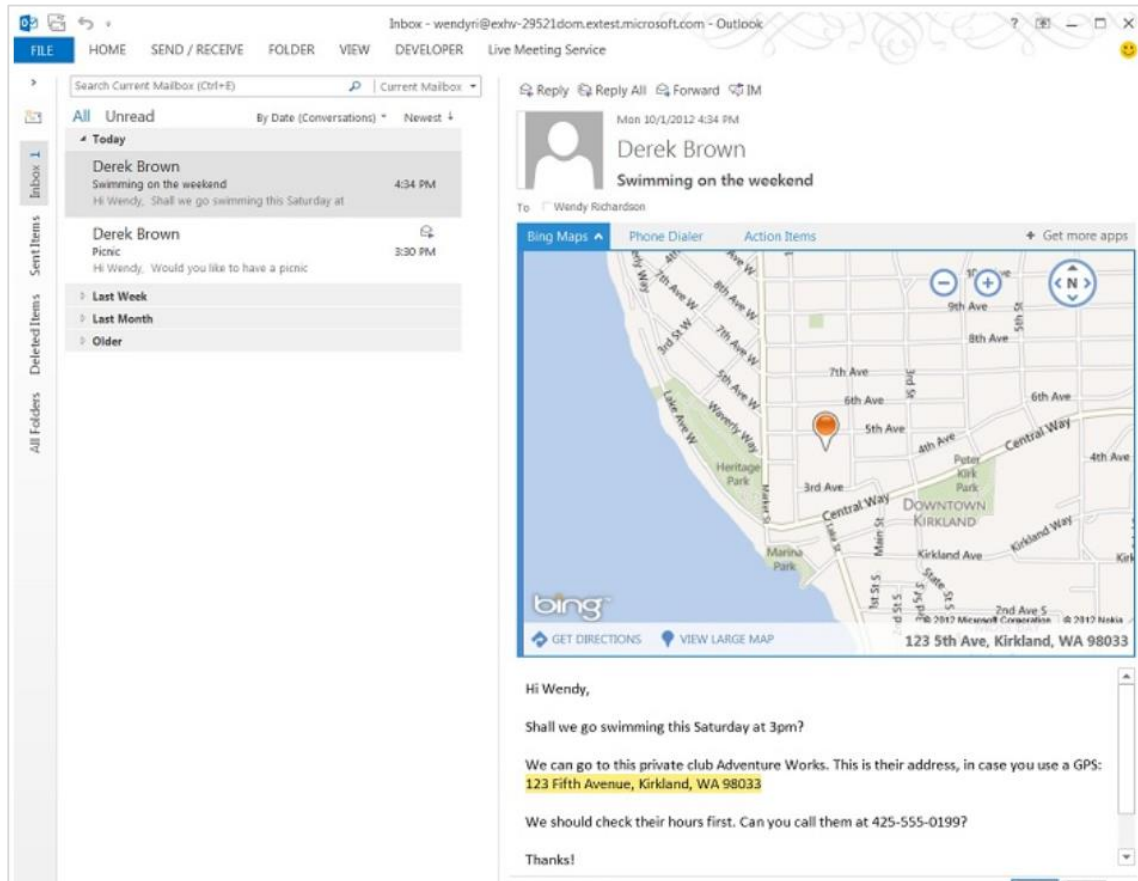
Tasks  
0 items  
Modified 1 minute ago



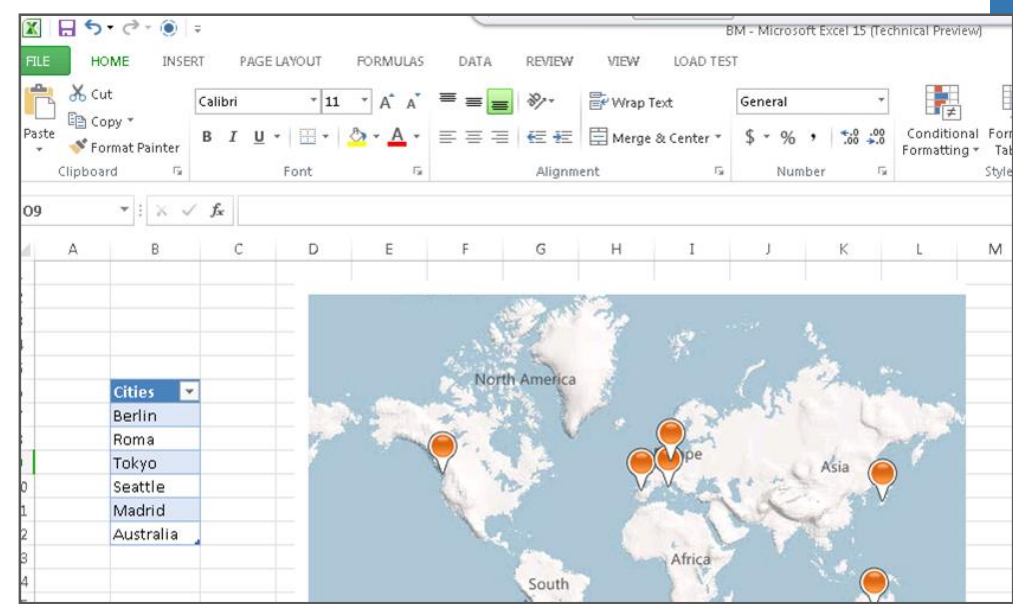
# Beneficios

- No se despliega código personalizado en el servidor SharePoint
  - Migraciones futuras más sencillas
  - Funcionan tanto en SharePoint Online como en SharePoint Server
- Reduce la rampa de aprendizaje
  - No es tan necesario conocer muchos conceptos de SharePoint
  - Se puede usar HTML5, JavaScript y REST si no se conoce .NET
- Las apps pueden residir en servidores externos con diferentes tecnologías
- Tienen más flexibilidad que las tradicionales webparts, controles, etc.

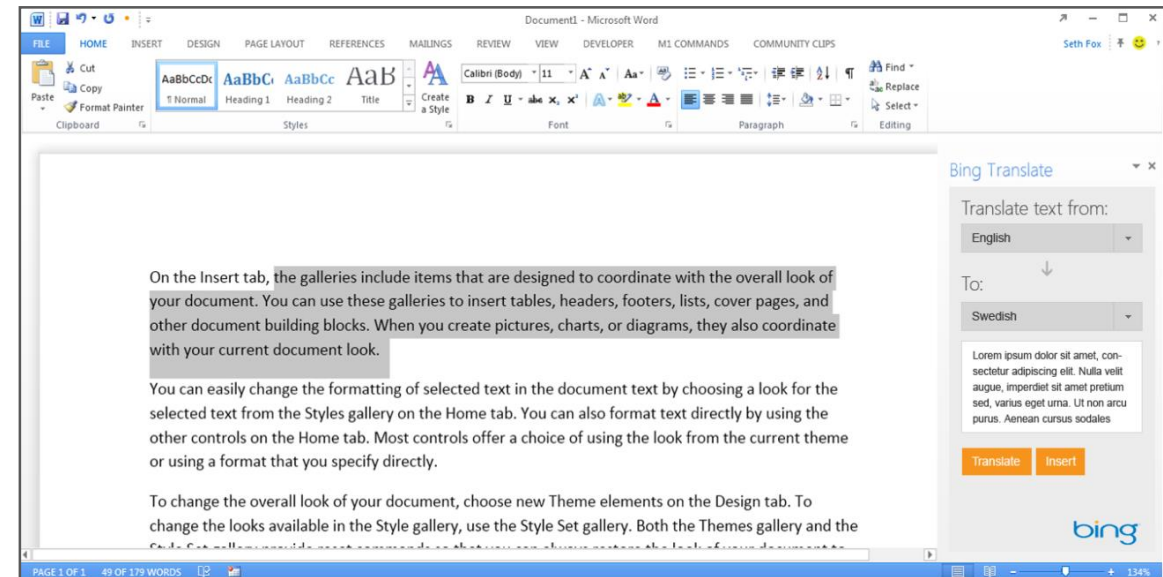
# ¡Office cliente también tiene apps!



Mail app (Outlook, Outlook Online)



Content app (Excel, Excel Online)



Task pane app (Word, Excel, Excel Online, PowerPoint, Project Professional)

# Conceptos básicos para desarrollar apps de SharePoint







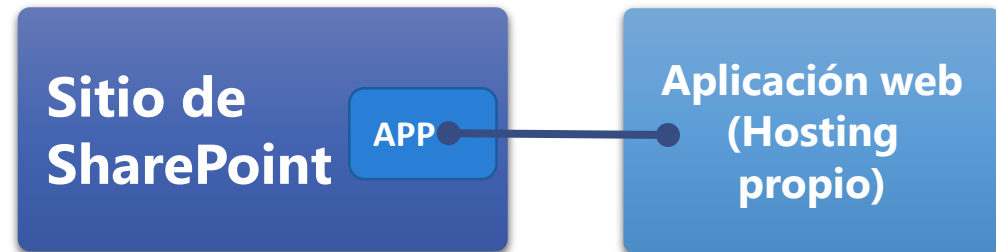
# Los tres tipos de app de SharePoint

## Cloud-based Apps

Llamadas a SharePoint para obtener o escribir información usando las API CSOM/REST + OAuth

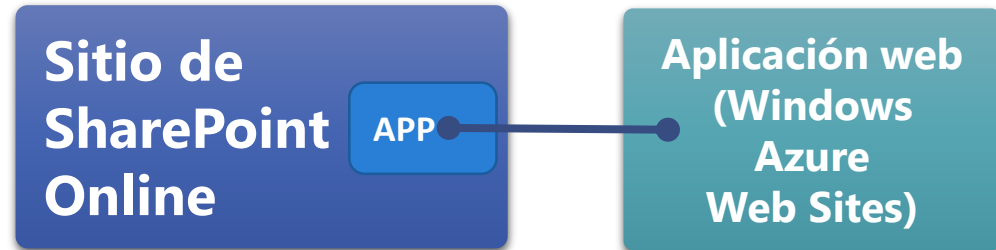
### Provider-Hosted App

La app reside en cualquier infraestructura y y está hecha con cualquier lenguaje. Da igual que el hosting sea Windows Server, Apache... Puede estar en .NET, Java, PHP, HTML5, JavaScript...



### ~~Autohosted App~~

~~Similar a la anterior, salvo que el hosting es Windows Azure + SQL Azure. El aprovisionamiento e instalación se hace de forma invisible. Solo en Office 365.~~



### SharePoint-Hosted App

No necesita hosting. El código de la app se ejecuta en el navegador cliente mediante JavaScript y/o REST.



# Formas para las app

## Descripción

## Ejemplo



App que se abre a pantalla completa para mostrar funcionalidades avanzadas. Permite tener la barra de navegación del sitio para volver

Reserva de recursos, panel de negocio...

Aunque no sean una app, Word Online, PowerPoint Online, Excel Online y OneNote Online funcionan de forma similar.



Se inserta en las páginas de SharePoint como una webpart tradicional

Visor del tiempo, visor de acciones, panel de control, mapa...



Añade nuevas acciones para trabajar con documentos e ítems en la cinta y menús

Abrir un visualizador del documento, imprimir un documento...



# Las app por dentro

- Una app por dentro es un sitio de SharePoint
- La app puede tener sus propias listas y bibliotecas que duran el tiempo que la app esté instalada
- También puede acceder a las listas y bibliotecas del sitio que las contiene





# El archivo AppManifest.xml

- AppManifest.xml define las propiedades de la app
- Visual Studio 2012/2013 ofrece un diseñador visual

The properties of the deployment package for your app are contained in the app manifest file. You can use the Manifest Designer to set or modify one or more of the properties.

General Permissions Prerequisites Supported Locales Remote Endpoints

Use this page to set properties that identify and describe your app when it is deployed.

Title:

Name:

Version:

Icon:

Required size: 96 x 96 pixels

Start page:

Query string:

SharePoint Hosted

```
<AppPrincipal>  
  <Internal />  
</AppPrincipal>
```

Autohosted

```
<AppPrincipal>  
  <AutoDeployedWebApplication />  
</AppPrincipal>
```

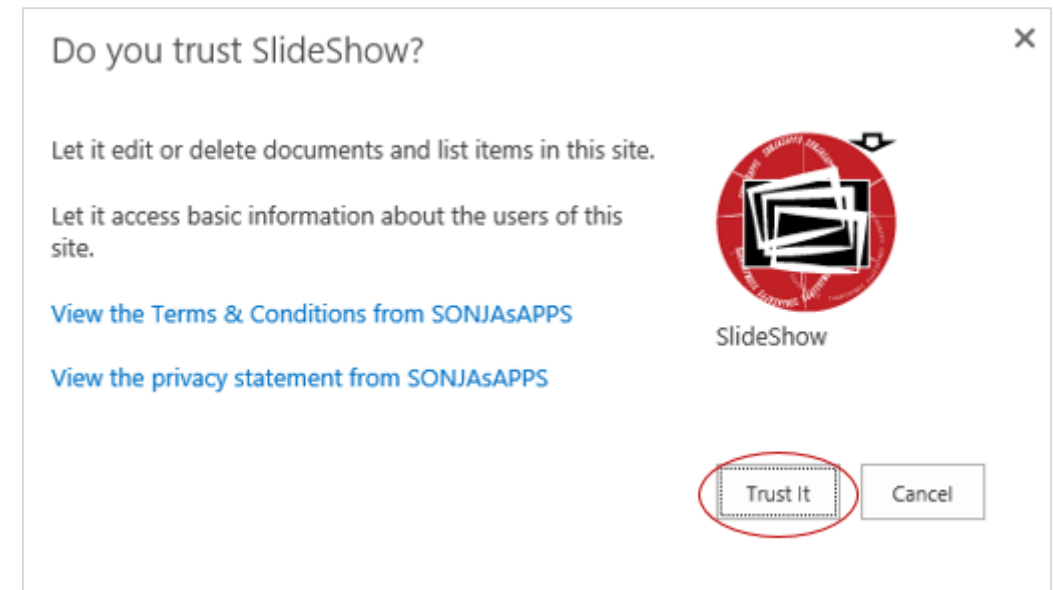
Provider Hosted

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



# Comunicación entre las app y SharePoint

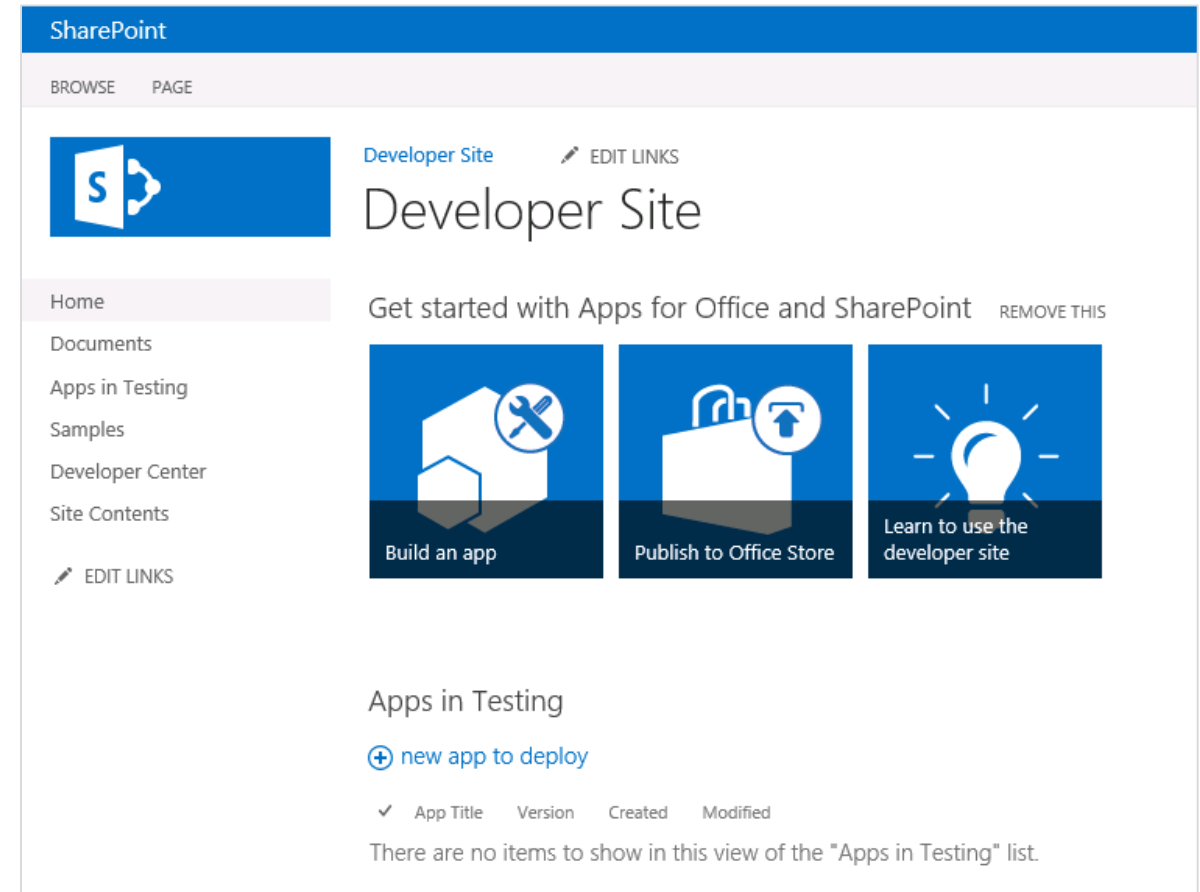
- Las apps provider hosted y autohosted se encuentran fuera de SharePoint y necesitan interactuar con su contenido para:
  - Acceder a datos
  - Efectuar operaciones (crear listas, editar permisos, etc.)
- Las app se comunican con SharePoint vía CSOM y REST
- Las apps necesitan permisos para acceder al contenido de SharePoint
  - Esto se consigue gracias al nuevo soporte a OAuth de SharePoint 2013





# Sitio del desarrollador y catálogo de apps

- El catálogo de apps es una colección de sitios con una biblioteca en la que almacenaremos los paquetes de las app
- El sitio del desarrollador es una colección de sitios con una biblioteca para almacenar apps que permite hacer debug y desplegar la app desde Visual Studio







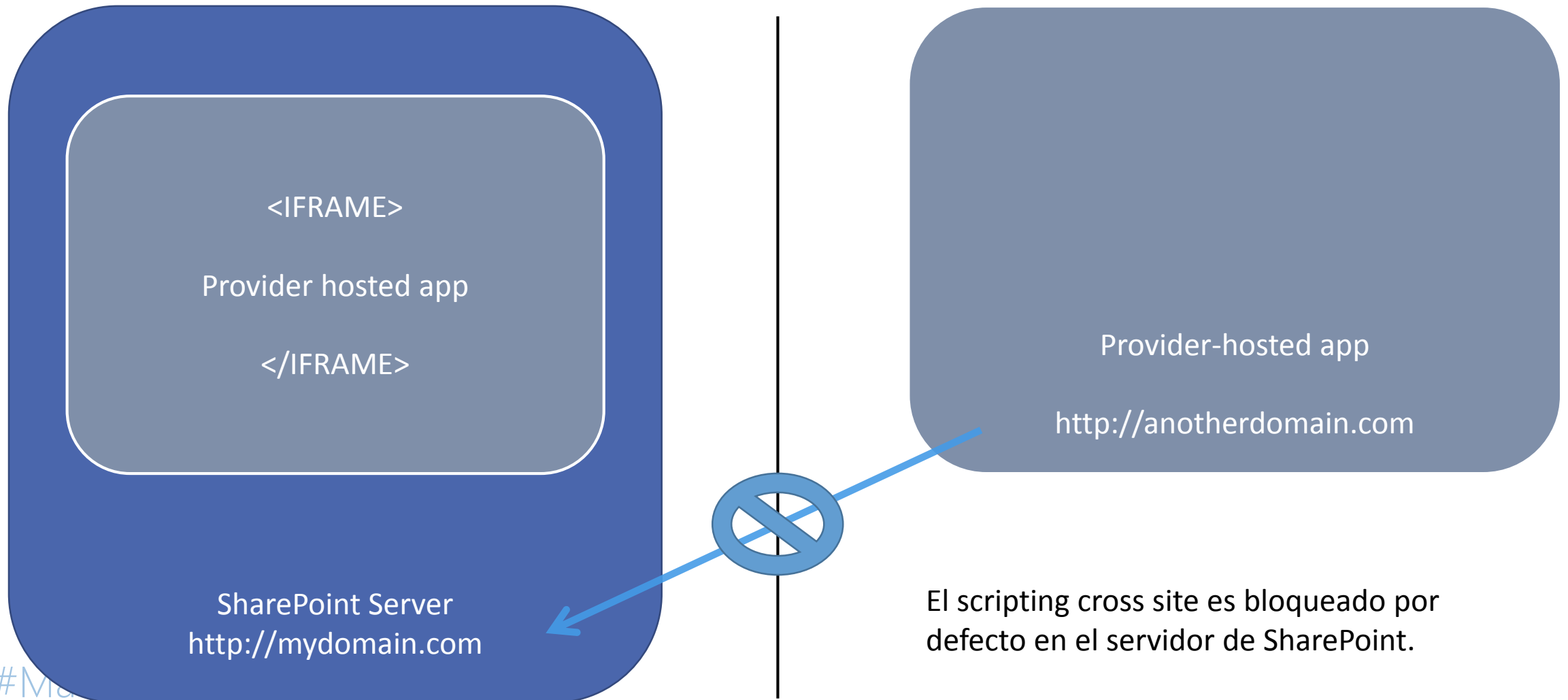
# Llamadas cross-domain



# Llamadas cross-domain

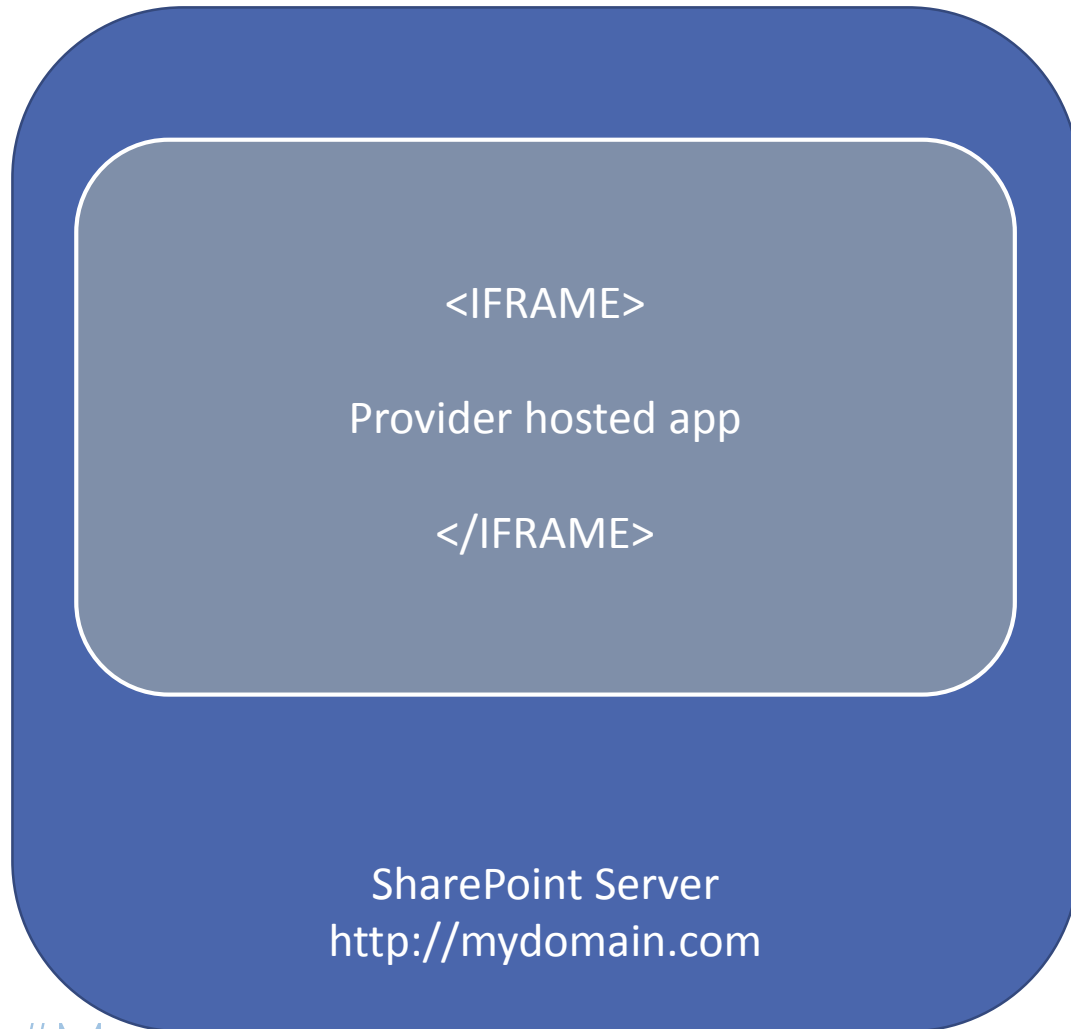
- Una app de SharePoint se muestra en la página dentro de un IFRAME
- La app está en un dominio distinto que el sitio de SharePoint
- Los navegadores no permiten a los elementos de la página interactuar con más de un dominio para evitar ataques
- Librería JavaScript **SP.RequestExecutor.js**
  - Se encuentra en el directorio LAYOUTS
- Con esta librería, las app pueden llamar a SharePoint aunque esté en dominio diferente
- El mecanismo de trabajo no es específico de SharePoint, es un patrón que se usa en la industria

# Las llamadas cross-domain son bloqueadas



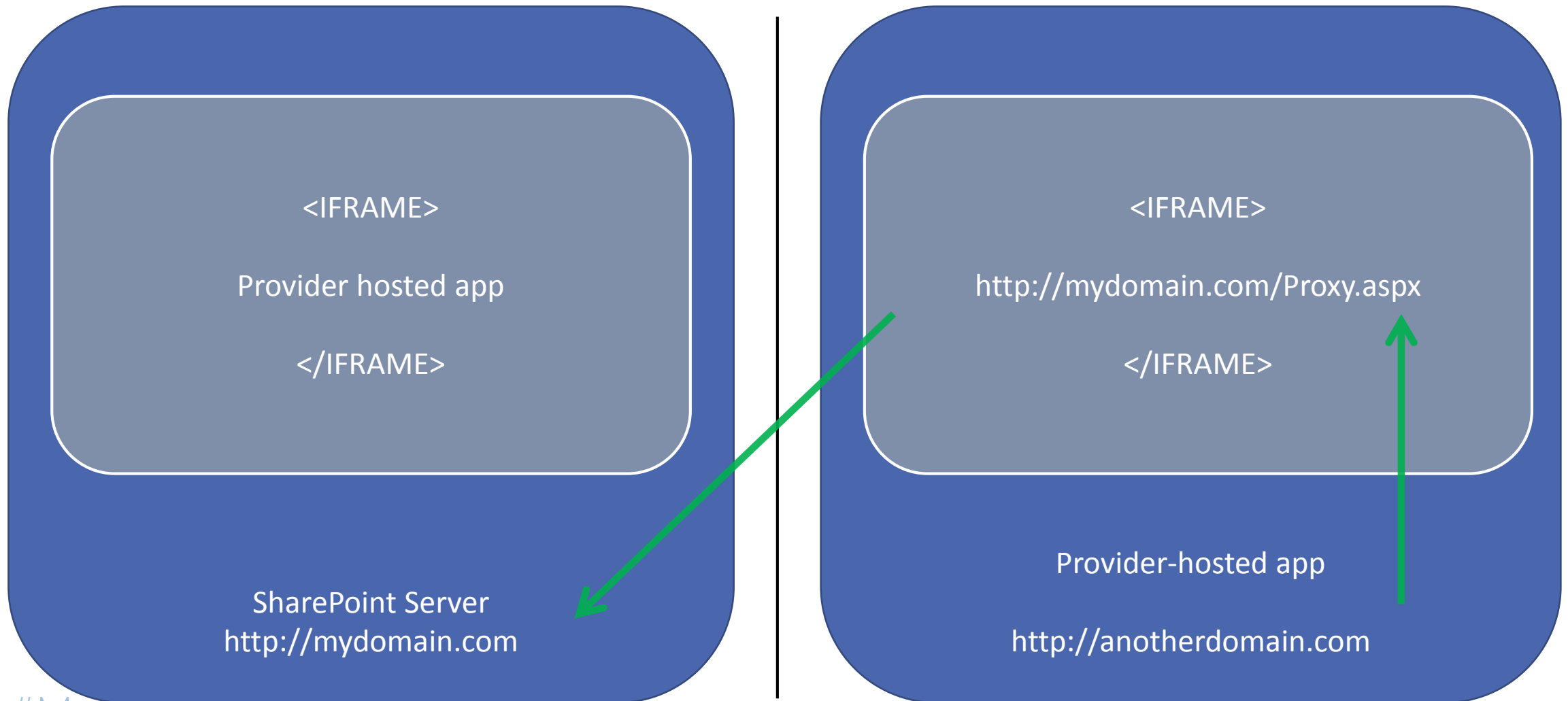


# SP.RequestExecutor.js





# Habilitando las llamadas cross-domain



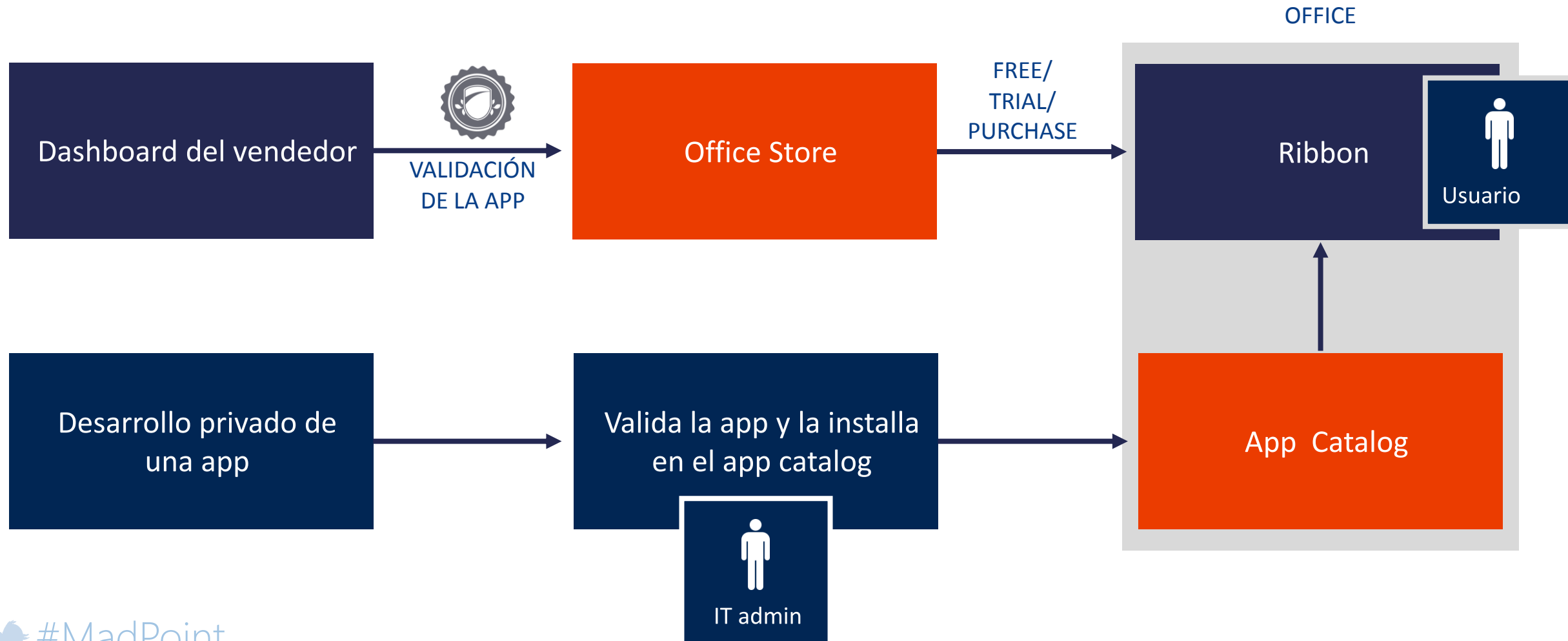
# Instalando apps y comprando en la tienda de SharePoint





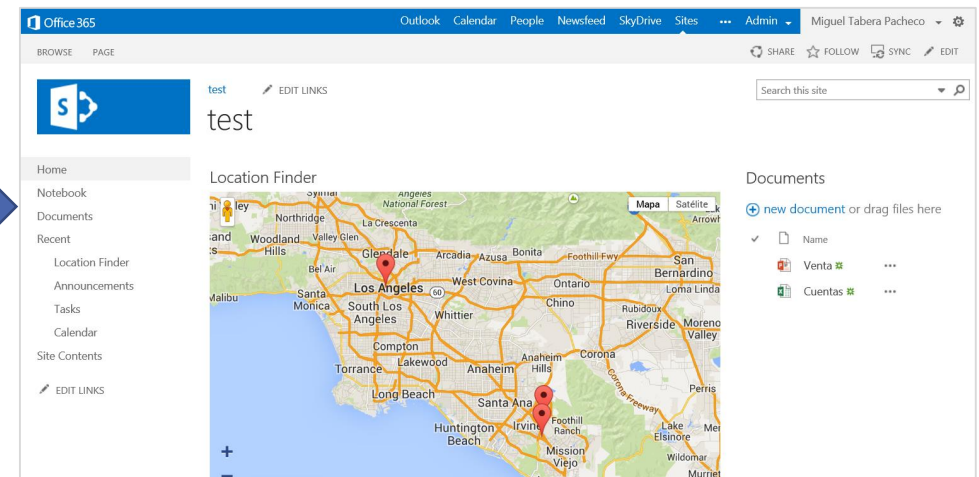
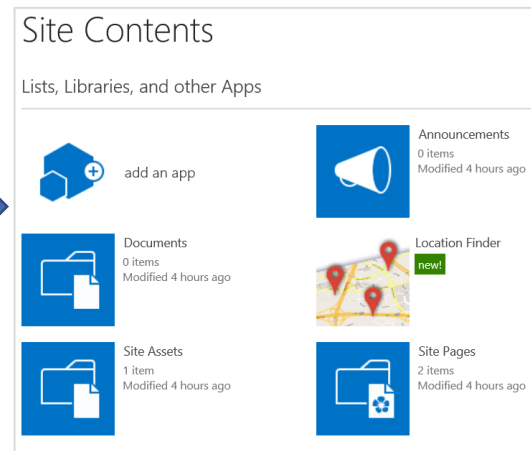
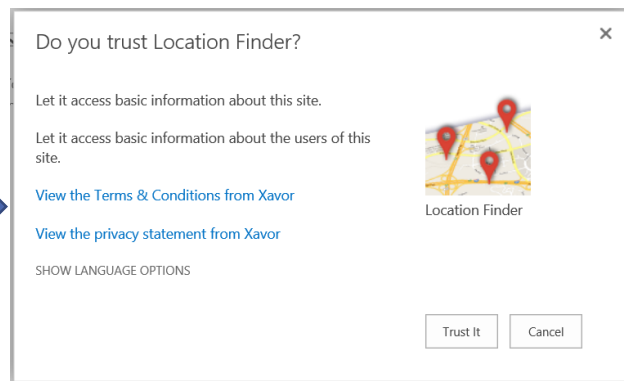
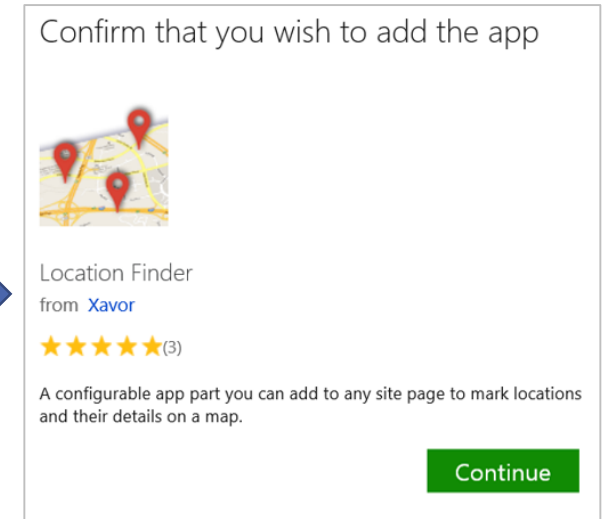
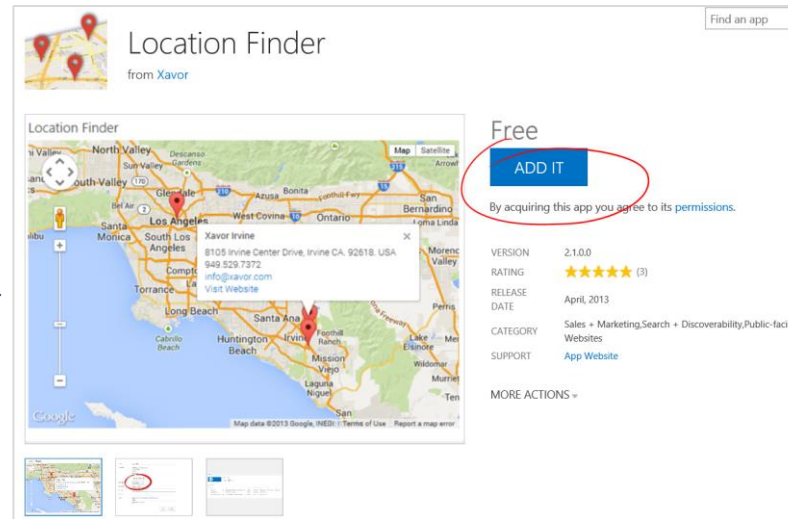
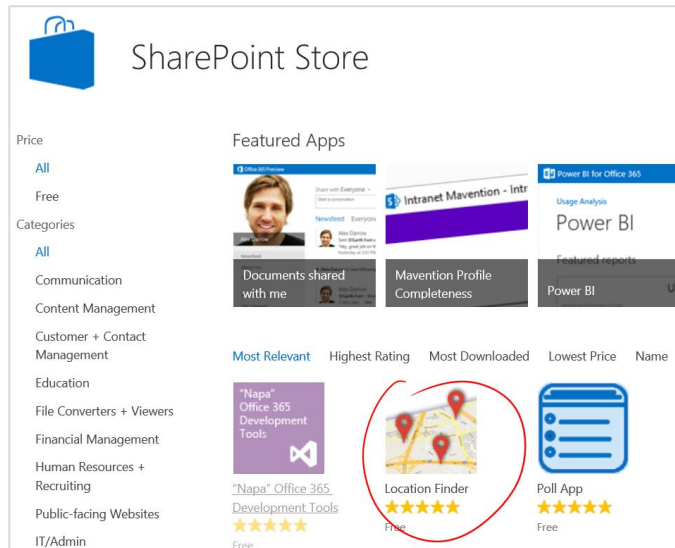


# Ciclo de instalación de una app





# Comprando en la tienda



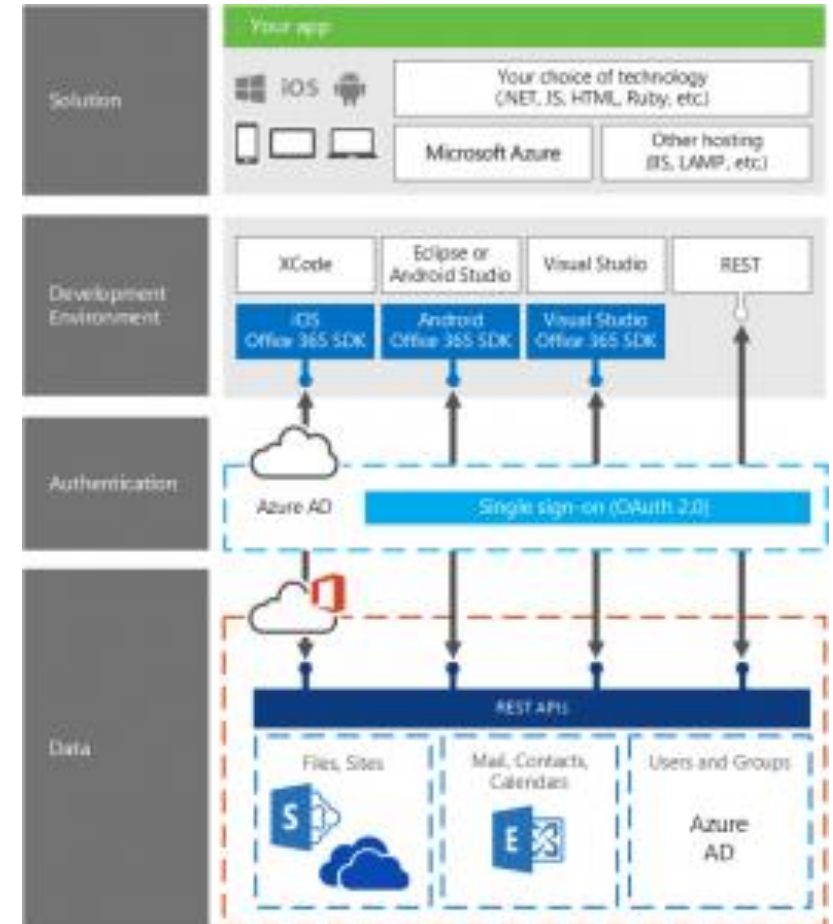
# Apps de Office 365





# Introducción a apps de Office 365

- Apps que trabajan con toda la plataforma de Office 365
- Se exponen diferentes servicios REST transversales para toda la plataforma:
  - Archivos, sitios y OneDrive for Business
  - Correo, contactos y calendarios
  - Usuarios y grupos
- Se pueden desarrollar con cualquier tecnología
- Autenticación basada en Azure Active Directory (AAD)







# Azure Active Directory

- Office 365 usa Azure AD.
- Puede generarse una suscripción de Azure a través del Azure AD de Office 365
- De esta forma se pueden crear apps en Azure que trabajen con Office 365

Office 365 admin center

Search users, admin tasks and more

DASHBOARD | ACTIVE USERS

Single sign-on: [Set up](#) | [Learn more](#)  
Active Directory® synchronization: [Set up](#) | [Learn more](#)  
[Manage mail contacts](#) | [Manage resource mailboxes](#) | [Manage shared mailboxes](#) in Exchange  
Change the password expiration policy for your users: [Change now](#)  
Set Multi-factor authentication requirements: [Set up](#) | [Learn more](#)

Select a view: All users

	DISPLAY NAME	USER NAME	STATUS
<input type="checkbox"/>	Joe	joe@wtofficedevtraining.onmicrosoft.com	In cloud
<input type="checkbox"/>	Paul	paul@wtofficedevtraining.onmicrosoft.com	In cloud

Microsoft Azure

CREDIT STATUS

wingtip office developer training

USERS GROUPS APPLICATIONS DOMAINS DIRECTORY INTEGRATION CONFIGURE REPORTS

LICENSES

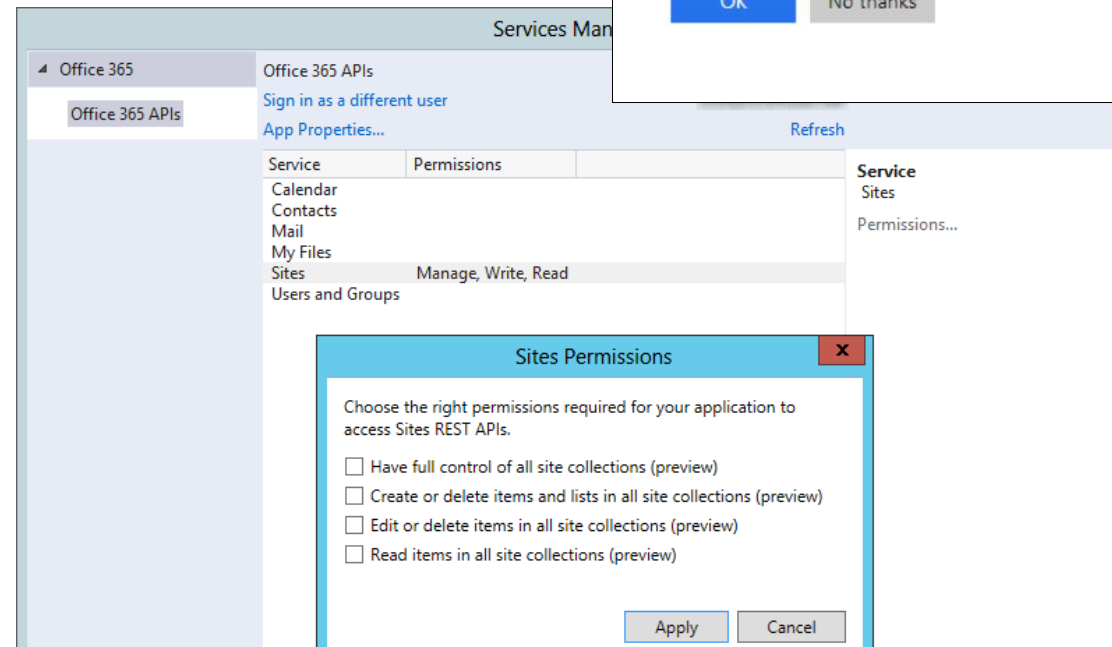
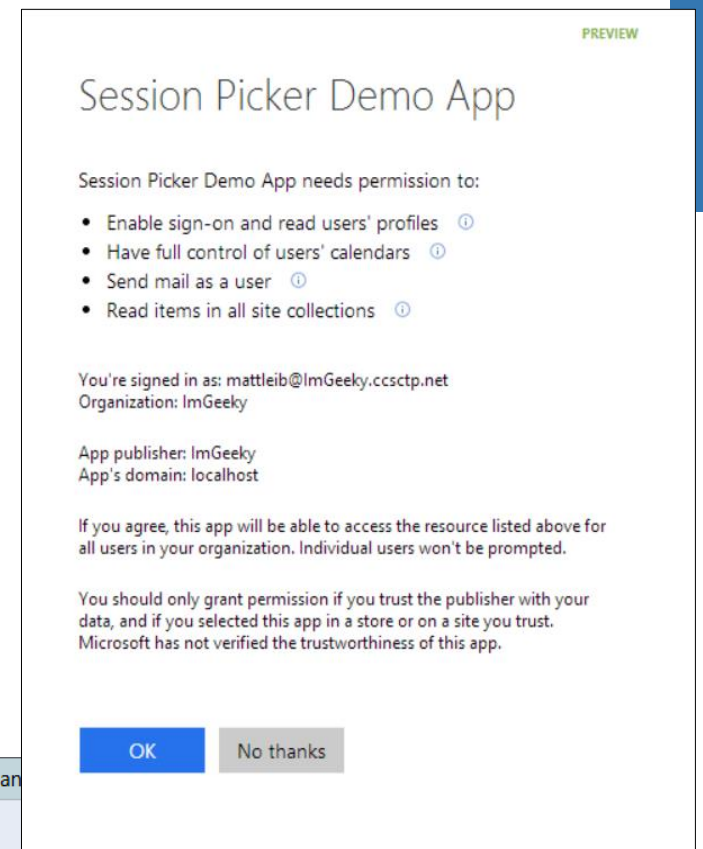
DISPLAY NAME	USER NAME	SOURCED FROM
Joe	joe@wtofficedevtraining.onmicrosoft.com	Windows Azure Active Directory
Paul	paul@wtofficedevtraining.onmicrosoft.com	Windows Azure Active Directory



# Oauth de AAD en Office 365

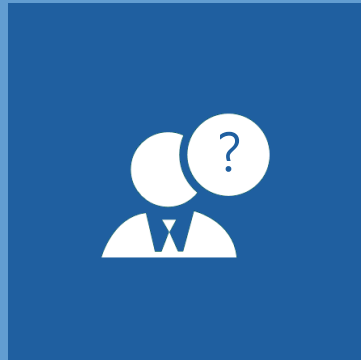
- Oauth 2.0
- Las aplicaciones no necesitan almacenar credenciales
- Azure AD Graph, Exchange y SharePoint

- Permisos en Visual Studio





MadPoint  
SharePoint Technical  
Community Madrid



Q&A

Colaboran:



Plain Concepts



**spenta**



Microsoft



**Beezy**  
Social at work

 #MadPoint





MadPoint  
SharePoint Technical  
Community Madrid

# Thank you!

Colaboran:



Plain Concepts



**spenta**



Microsoft



**Beezy**  
Social at work



#MadPoint