Hand son lab Creación Tab en Teams mediante SPFx 1.7

Olga Martí

@olgaamarti

v1

Contenido

[Objetivos del laboratorio 2](#_Toc530094724)

[Prerequisitos 2](#_Toc530094725)

[Creación del WebPart en SPFx 2](#_Toc530094726)

[Consumir el context de Teams y SharePoint 3](#_Toc530094727)

[Depurar nuestro WebPart 5](#_Toc530094728)

[Publicar el WebPart 6](#_Toc530094729)

[Añadir App a Microsoft Teams 7](#_Toc530094730)

# Objetivos del laboratorio

El objetivo del laboratorio es el de proporcionar los conocimientos para crear una pestaña en Microsoft Teams mediante SharePoint Framework

# Prerequisitos

Será necesario disponer:

* Disponer de una suscripción de Office 365 en Target Release con administrador Global.
* Microsoft Teams Escritorio Instalado.
* Herramientas de desarrollo de SPFx (Instrucciones: <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/set-up-your-development-environment> )

# Creación del WebPart en SPFx

Abrimos nuestra terminal y navegamos a la carpeta donde deseemos guardar nuestra solución:

1. Creamos la carpeta que contendrá nuestra solución

md teams-tab-webpart

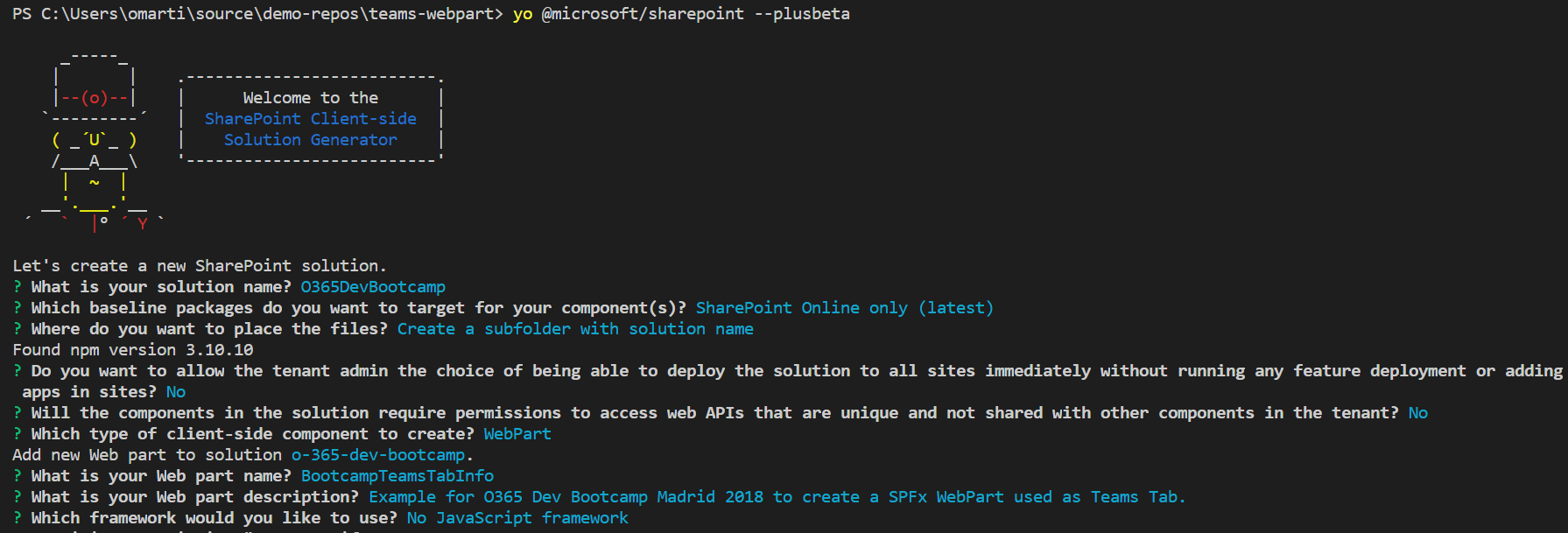
1. Navegamos a la carpeta

cd teams-tab-webpart

1. A continuación, ejecutamos el generador de Yeoman de SharePoint. Importante tener la última versión del generador instalada.

Lo ejecutamos en modo beta para obtener la versión 1.7 plus beta.

yo @microsoft/sharepoint --plusbeta

1. Completamos el Wizard de generación del proyecto:

Nombre de la solución: O365DevBootcamp

Package Baseline: SharePoint Online only.

Place the files: Subfolder with solution name.

Deploy Solution to All sites: No

Unique Permissions: No

Type of component: WebPart

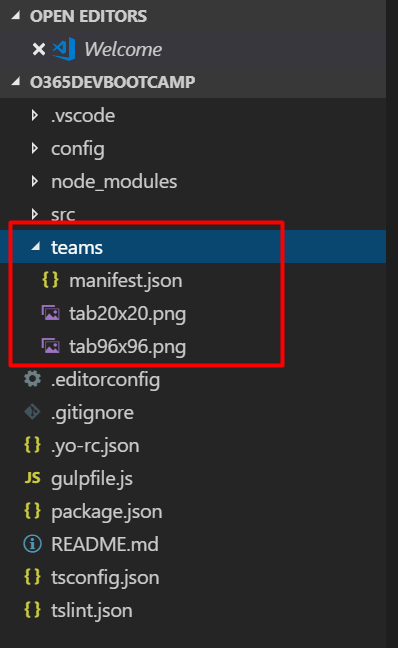
WebPArt Name: BootcampTeamsTabInfo

Description: Example for 0365Dev Bootcamp Madrid 2018.

Framework: No JavaScript Framework

# Consumir el context de Teams y SharePoint

Una vez tenemos la solución creada, podemos observar una carpeta llamada Teams, que es donde se encuentra nuestro Manifest.json y los iconos de nuestra App.



1. Nos dirigimos a: **\src\webparts\bootcampTeamsTabInfo\BootcampTeamsTabInfoWebPart.ts**

Donde dentro de nuestra clase declararemos la siguiente variable:

// This variable has been added

private \_teamsContext: microsoftTeams.Context;

1. Inicializamos:

protected onInit(): Promise<any> {

let retVal: Promise<any> = Promise.resolve();

if (this.context.microsoftTeams) {

retVal = new Promise((resolve, reject) => {

this.context.microsoftTeams.getContext(context => {

this.\_teamsContext = context;

resolve();

});

});

}

return retVal;

}

1. Sustituimos el render por el siguiente:

public render(): void {

let title: string = '';

let subTitle: string = '';

let siteDescription: string = '';

if (this.\_teamsContext) {

// We have teams context for the web part

title = "Welcome to " +this.\_teamsContext.teamName+ "Teams!";

subTitle = "You're in the " + this.\_teamsContext.channelName + " channel.";

siteDescription = "This Team is about: " + this.properties.description;

}

else

{

// We are rendered in normal SharePoint context

title = "Welcome to "+ this.context.pageContext.web.title +" SharePoint!";

subTitle = "This sites is connected to a Microsoft Teams.";

siteDescription = "This SharePoint Site is about: " + this.properties.description;

}

this.domElement.innerHTML = `

<div class="${styles.bootcampTeamsTabInfo}">

<div class="${ styles.container }">

<div class="${ styles.row }">

<div class="${ styles.column }">

<span class="${ styles.title }">${title}</span>

<p class="${ styles.subTitle }">${subTitle}</p>

<p class="${ styles.description }">${siteDescription}</p>

<a href="https://aka.ms/spfx" class="${ styles.button }">

<span class="${ styles.label }">Learn more</span>

</a>

</div>

</div>

</div>

</div>`;

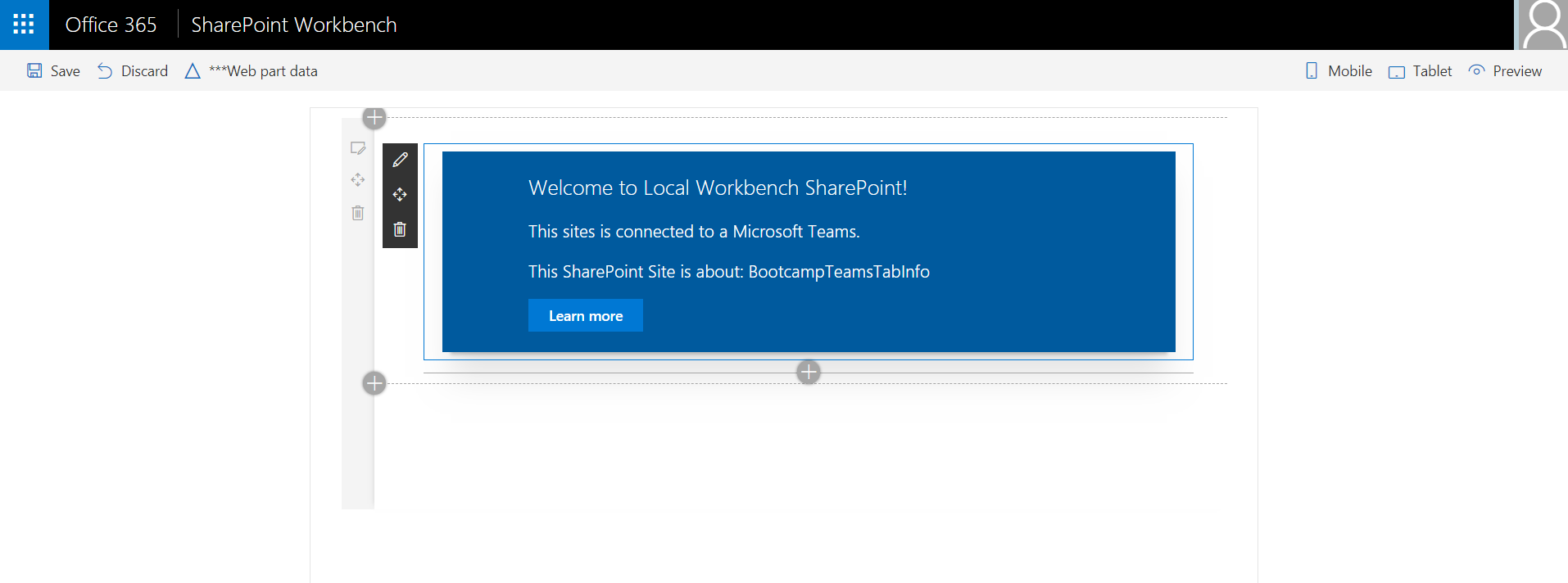
}

# Depurar nuestro WebPart

Para depurar, utilizaremos el Workbench de SharePoint. Ejecutaremos el comando:

gulp serve

Se nos abrirá el Workbench local de SharePoint donde podremos depurar.

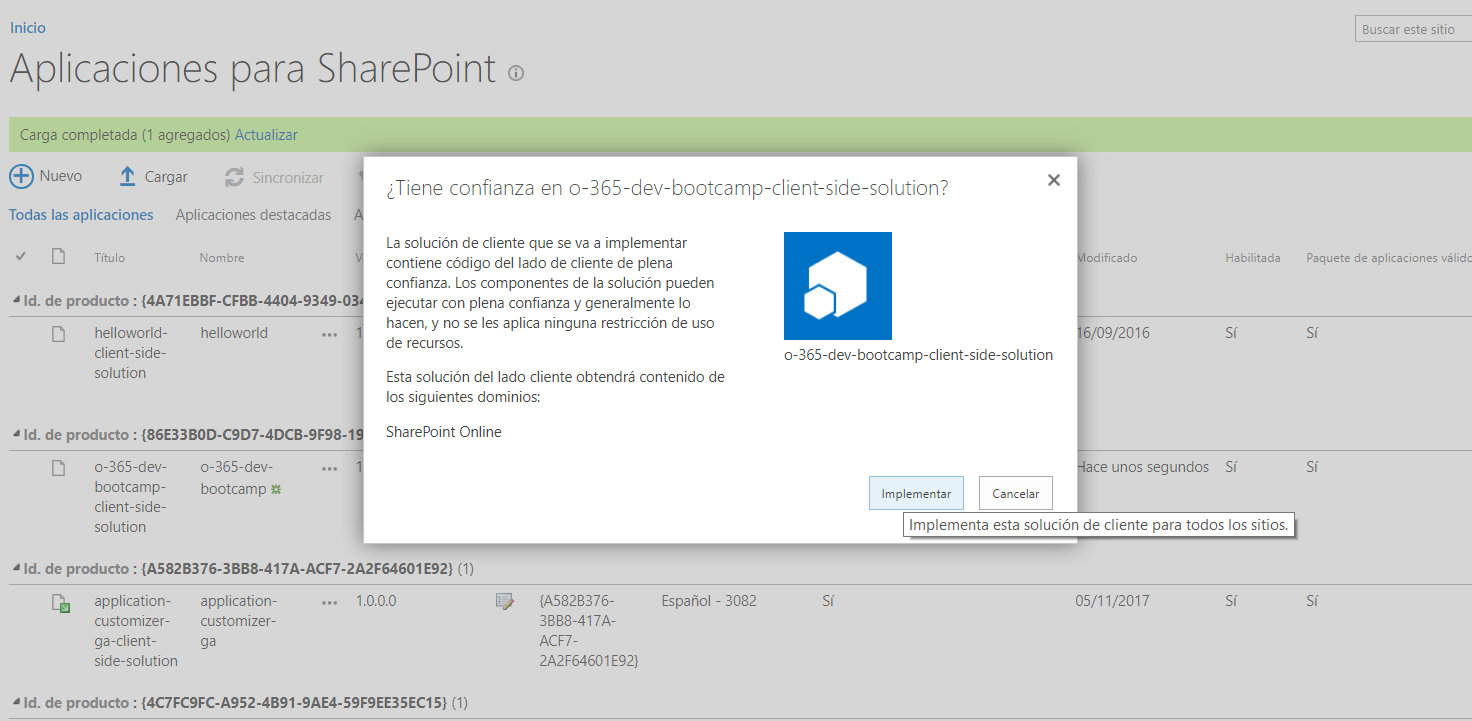


Si queremos obtener el contexto de SharePoint, tendremos que ir al Workbench de SharePoint.

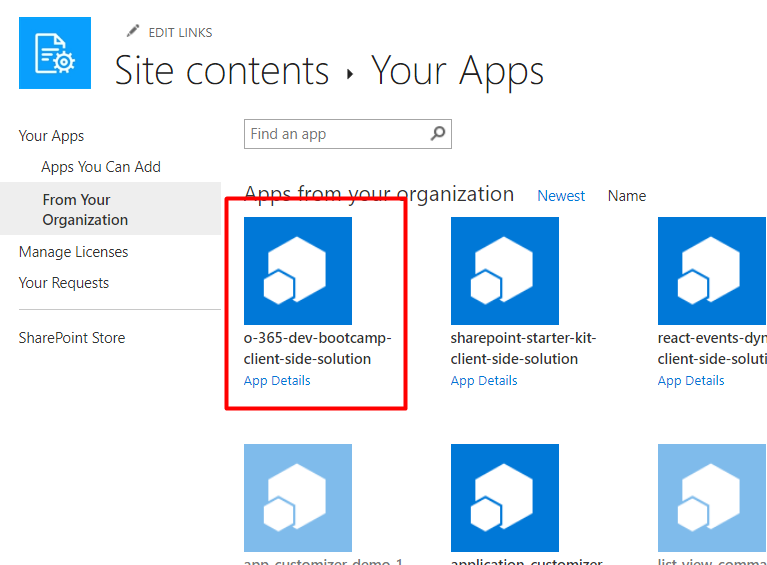
https://your-sharepoint-tenant.sharepoint.com/\_layouts/workbench.aspx

# Publicar el WebPart

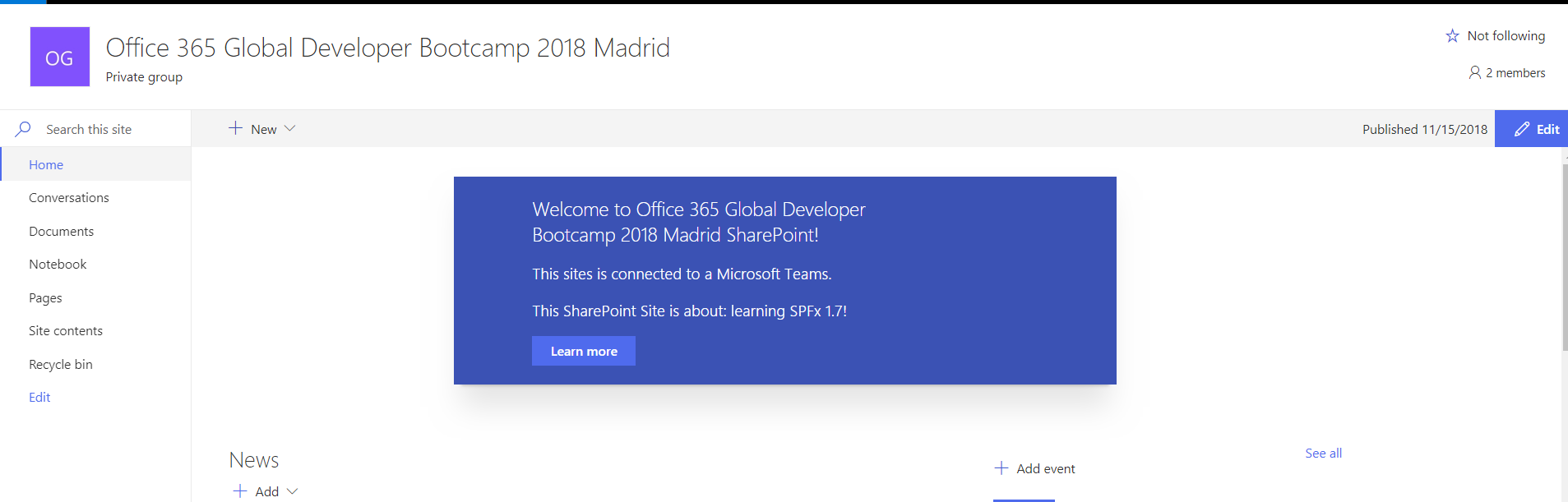
Antes de insertar nuestro WebPart como pestaña, deberemos publicar el WebPart en el App Catalog de SharePoint como si fuera un WebPart de SPFx corriente.



He instalamos nuestra aplicación en el SharePoint Site de nuestro Microsoft Teams.



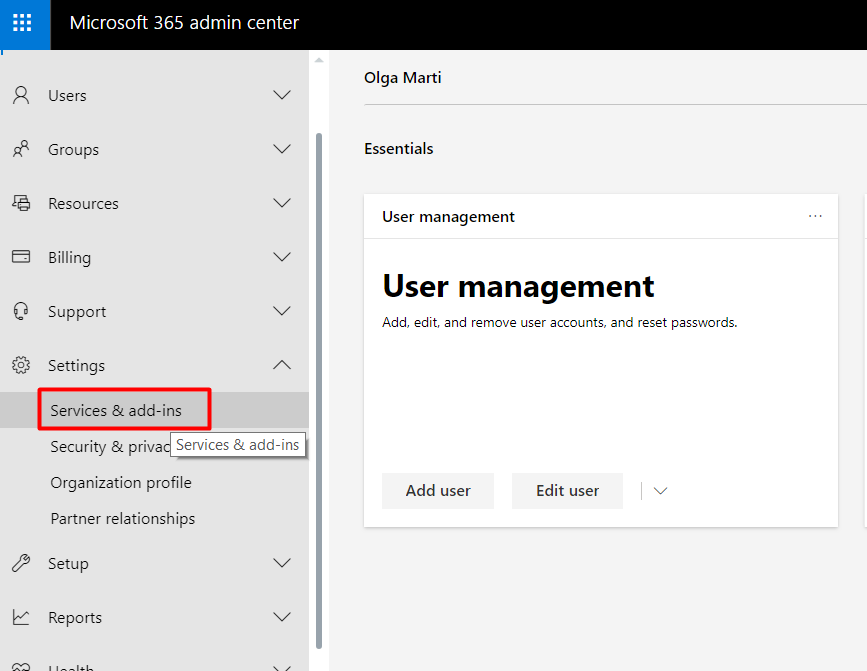
Al cabo de unos minutos, podremos añadir nuestro WebPart en una página de SharePoint:



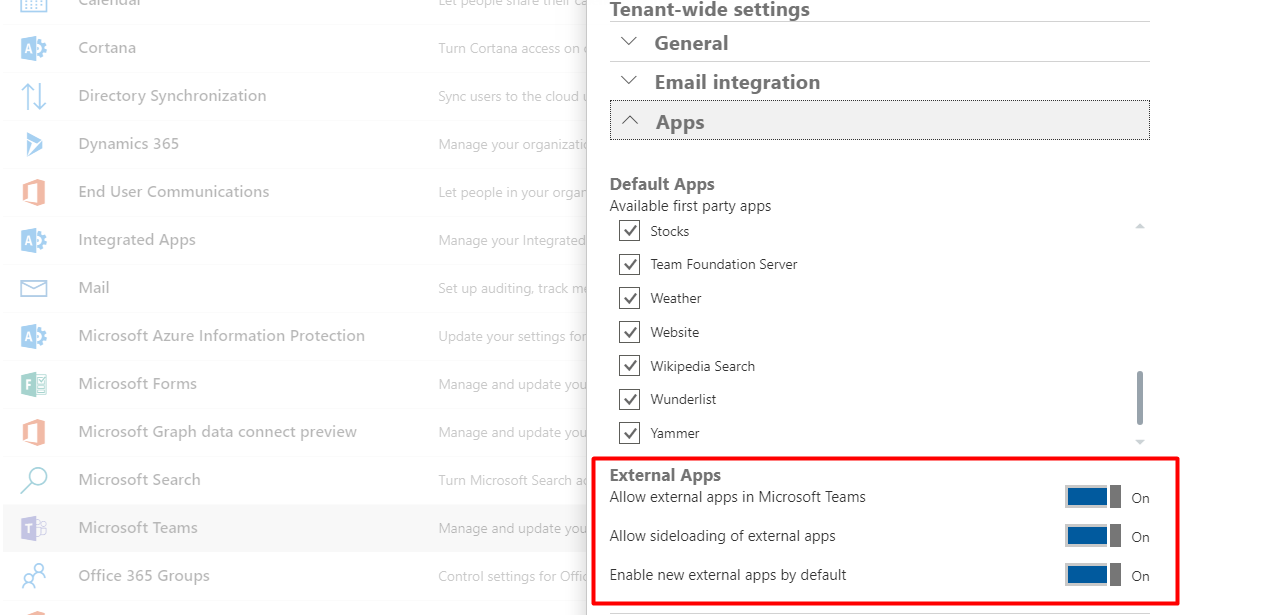
# Añadir App a Microsoft Teams

Para añadir una App propia a Microsoft Teams deberemos tener habilitado en la Administración de Office 365 el SideLoading de Apps.

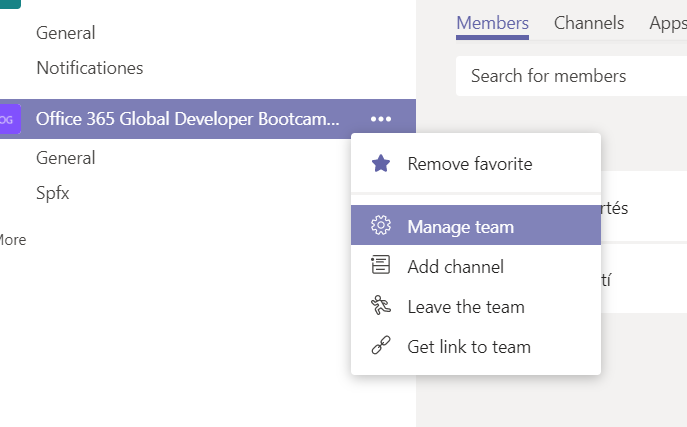
1. Vamos a la Administración Central de Microsoft 365 y seleccionamos Settings -> Services & Addins.



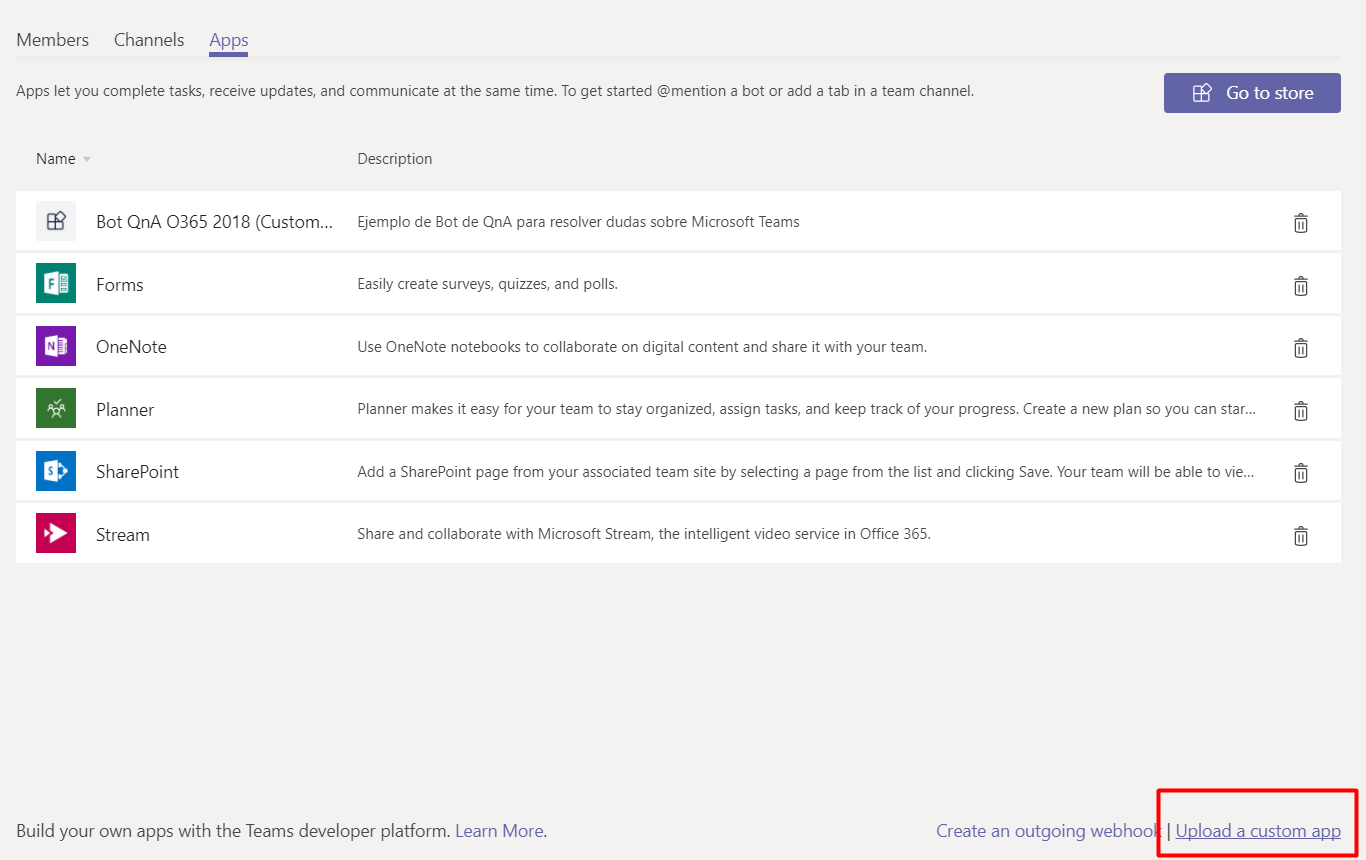
1. Buscamos Microsoft Teams y habilitamos las siguientes opciones dentro de la sección de Apps:

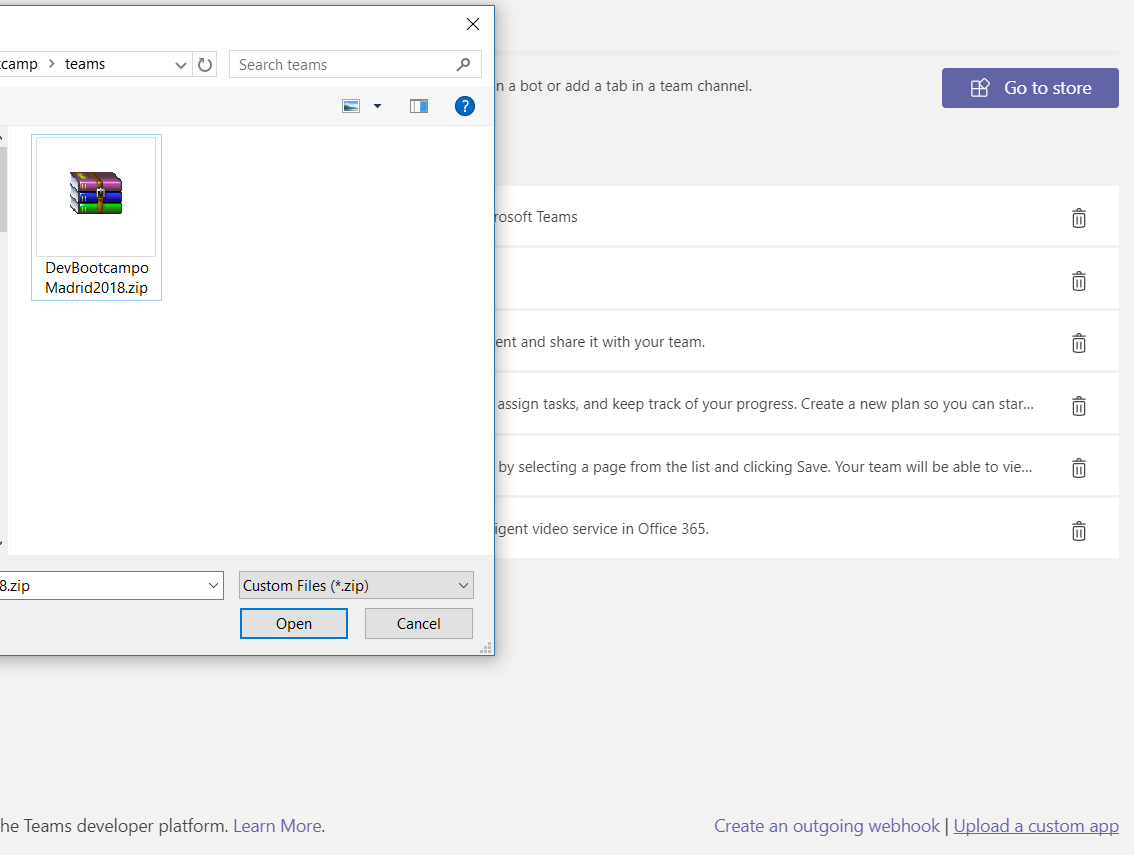


1. Una vez hemos guardado la configuración, nos dirigmos al Microsoft Teams y selecciomos “Manage Team” para ir a la configuración de las Apps.

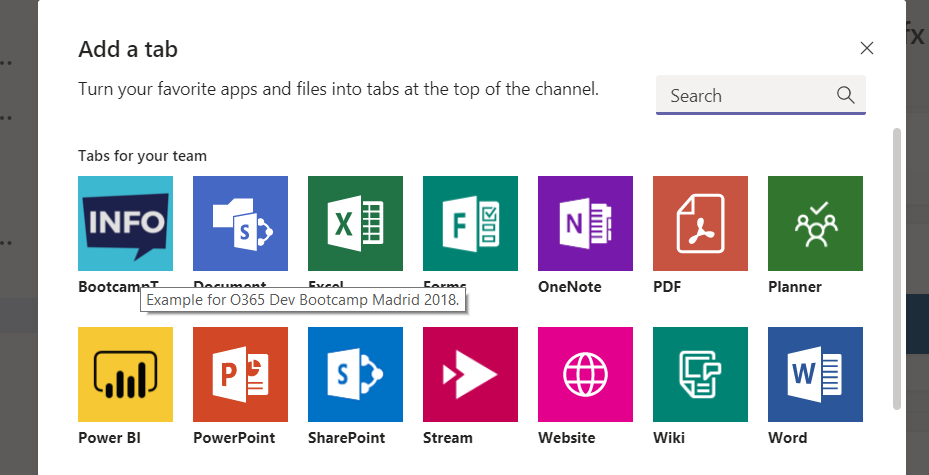


1. En la esquina inferior derecha, encontraremos un enlace que nos permitirá cargar nuestro paquete, compuesto por las dos imágenes y el manifest.json.





1. Si la carga ha sido correcta, cuando vayamos a añadir una nueva pestaña en un canal, nos aparecerá nuestra aplicación:



1. Nos pedirá la configuración de la pestaña, que se trata del panel de propiedades del WebPart:

