Imanol Iza

Modern Workplace Lead en Necsia

[Imanol.iza@gmail.com](mailto:Imanol.iza@gmail.com)

[Linkedin](https://www.linkedin.com/in/imanoliza/) [Twitter](https://twitter.com/ima_iza)

bot de soporte con microsoft powerapps

Office 365 Developer Bootcamp Hands-on Lab

Introducción

Tiempo estimado para completar este laboratorio

60 minutos

Prerrequisitos

Para este laboratorio se requiere una cuenta activa de Office 365. Puedes usar el tenant proporcionado por el Office 365 Bootcamp o crearte un tenant Office 365 de prueba <http://go.microsoft.com/fwlink/p/?LinkID=403802>

Objetivos

Después de haber completado este laboratorio serás capaz de:

* Conocer el servicio cognitivo QnAMaker
* Crear y configurar un conector personalizado para extender Microsoft PowerApps
* Crear una aplicación en Microsoft PowerApps
* Consumir la API de QnAMaker mediante Microsoft PowerApps

Descripción del Laboratorio

Muchas veces, cuando pensamos en Microsoft PowerApps solemos pensar en aplicaciones empresariales sencillas, vinculadas a las listas de SharePoint y con flujos por debajo sin complejidad. Este laboratorio pretende servir para conocer como poder extender Microsoft PowerApps a través de los conectores personalizados consumiendo uno de los Servicios Cognitivos de Microsoft más conocidos, QnAMaker.

Ejercicio 1: Creación de servicio QnA Maker

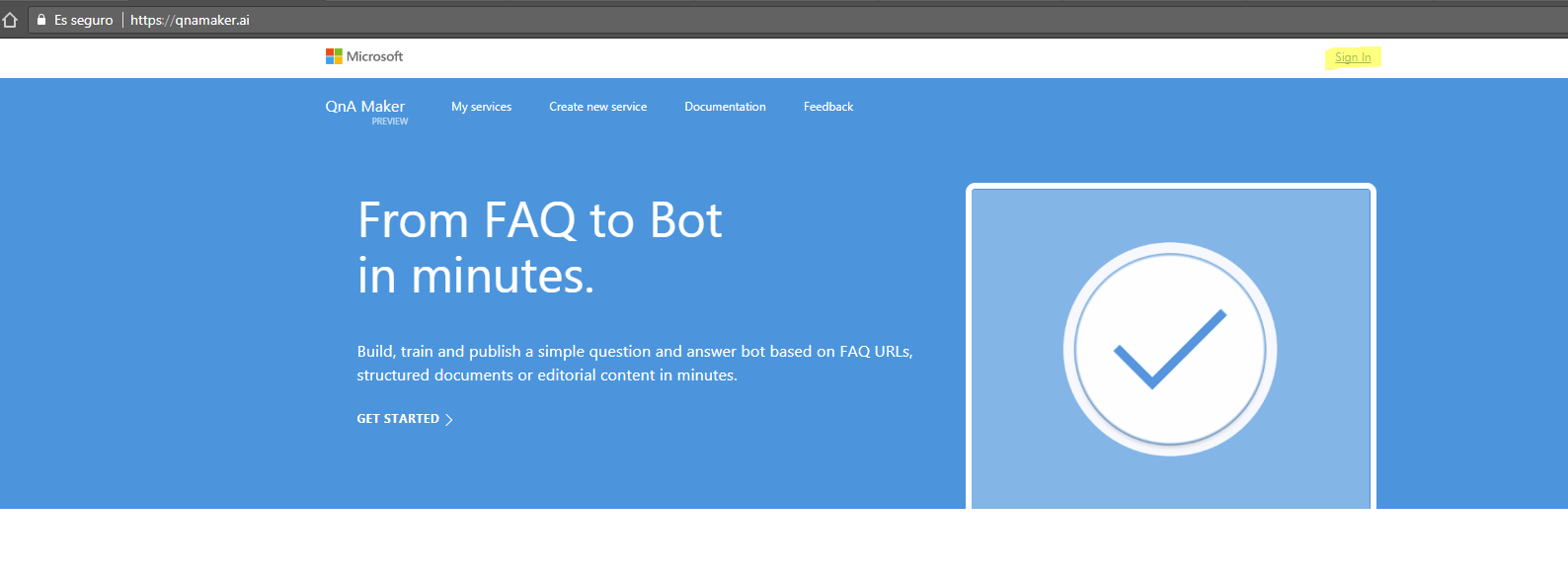
Ejercicio 2: Creación de un conector personalizado en Microsoft PowerApps

Ejercicio 3: Creación de una aplicación en Microsoft PowerApps

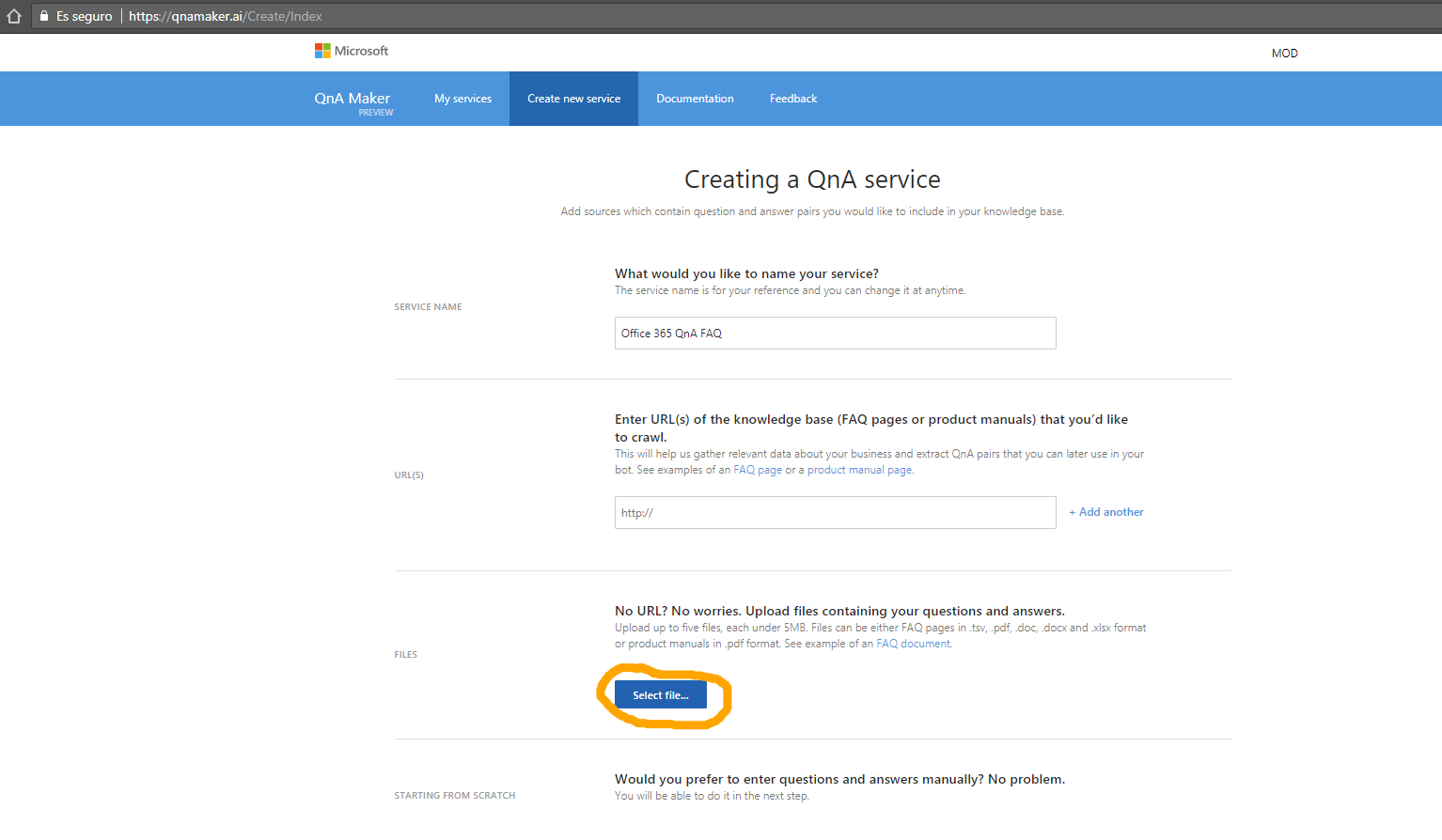
Ejercicio 1: Creación de servicio QnA Maker

En el siguiente ejercicio crearemos un nuevo servicio de QnA Maker desde el portal del servicio:

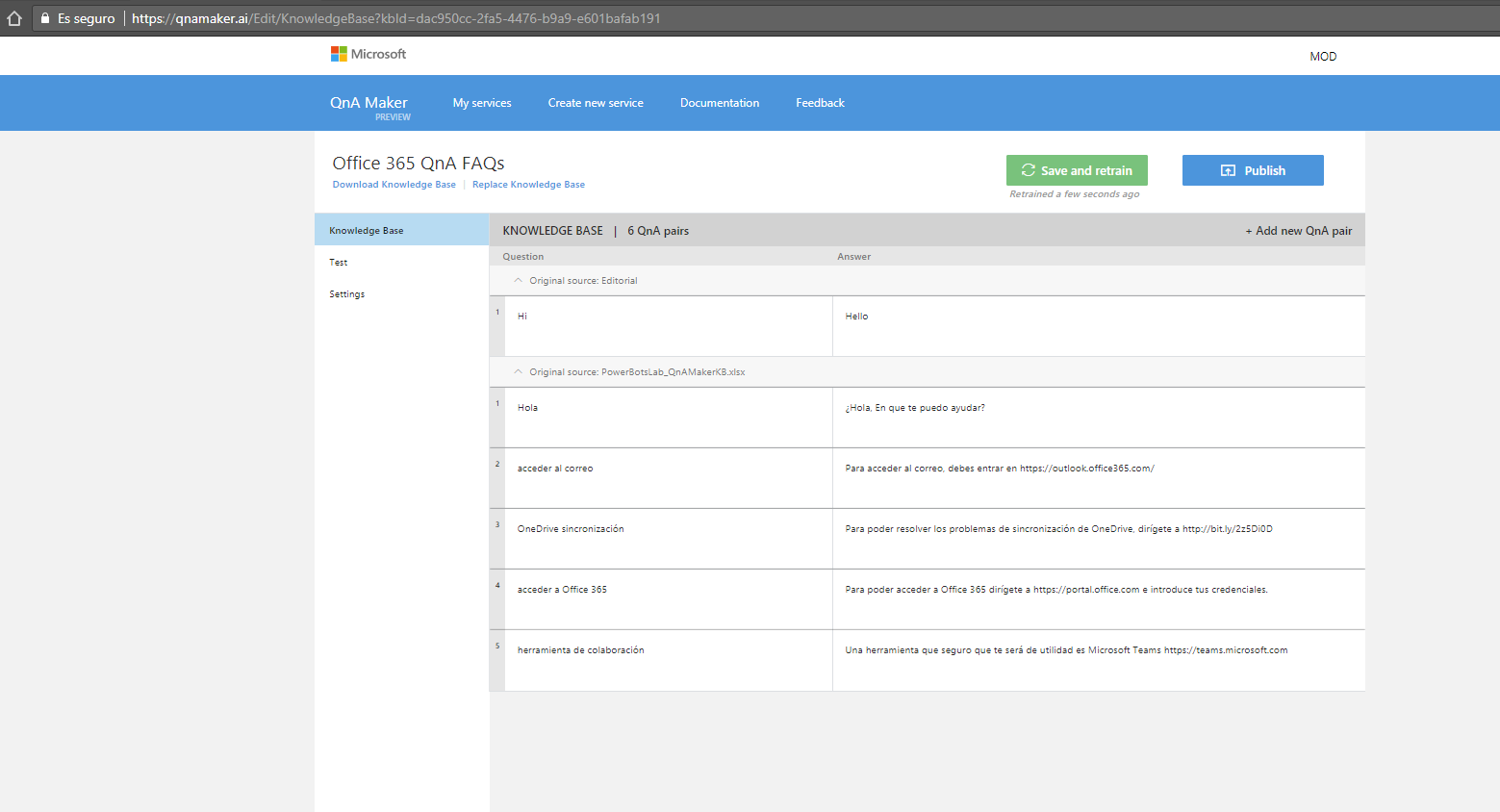
1. Dirígete a <https://qnamaker.ai/> e introduce tus credenciales proporcionadas para el Bootcamp



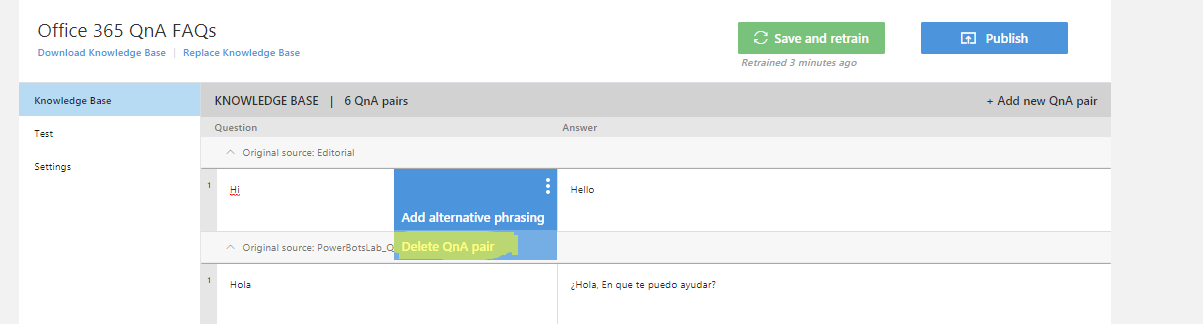
1. La primera vez te pedirá autorización, adelante. Vamos a crear un nuevo servicio. Selecciona “Crear nuevo servicio”. Ponle el nombre “Office 365 QnA FAQ”
2. Selecciona adjuntar un fichero y adjunta el fichero “PowerBotsLab\_QnAMakerKB.xslx” que está en la carpeta del laboratorio. A continuación, selecciona Crear.



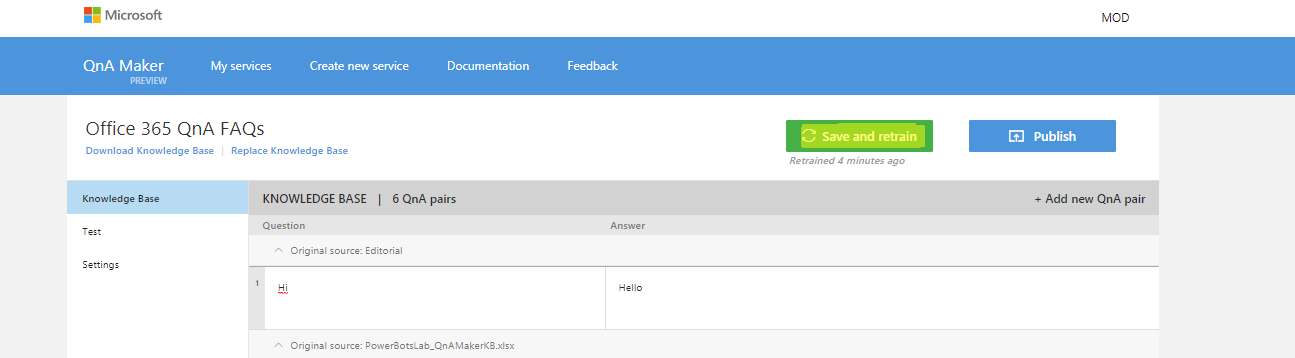
1. Veras como se han añadido varios registros a nuestra base de datos de conocimiento.



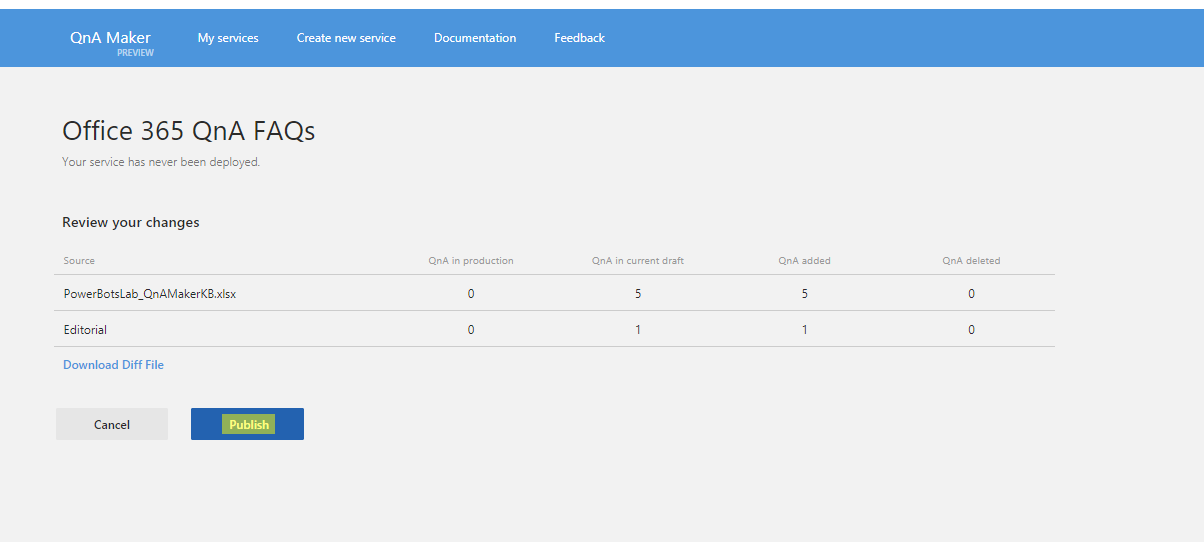
1. Elimina el primer registro ya que no lo vamos a necesitar.



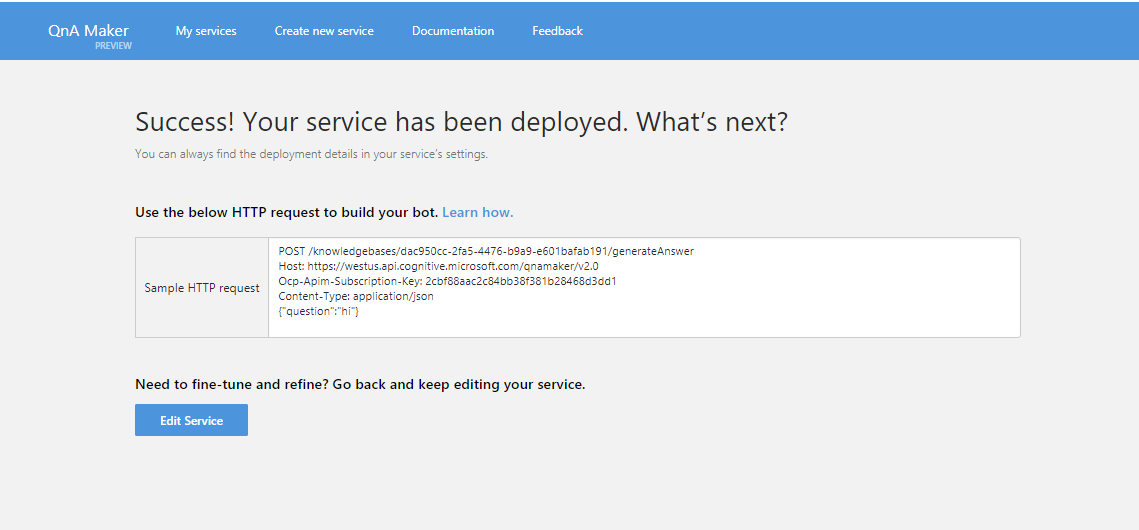
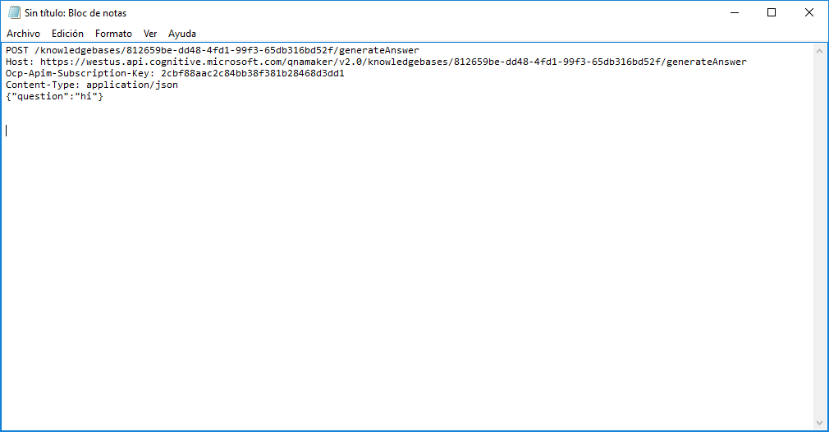
1. Selecciona Sabe and retrain



1. Y finalmente Publish, para publicar el servicio.



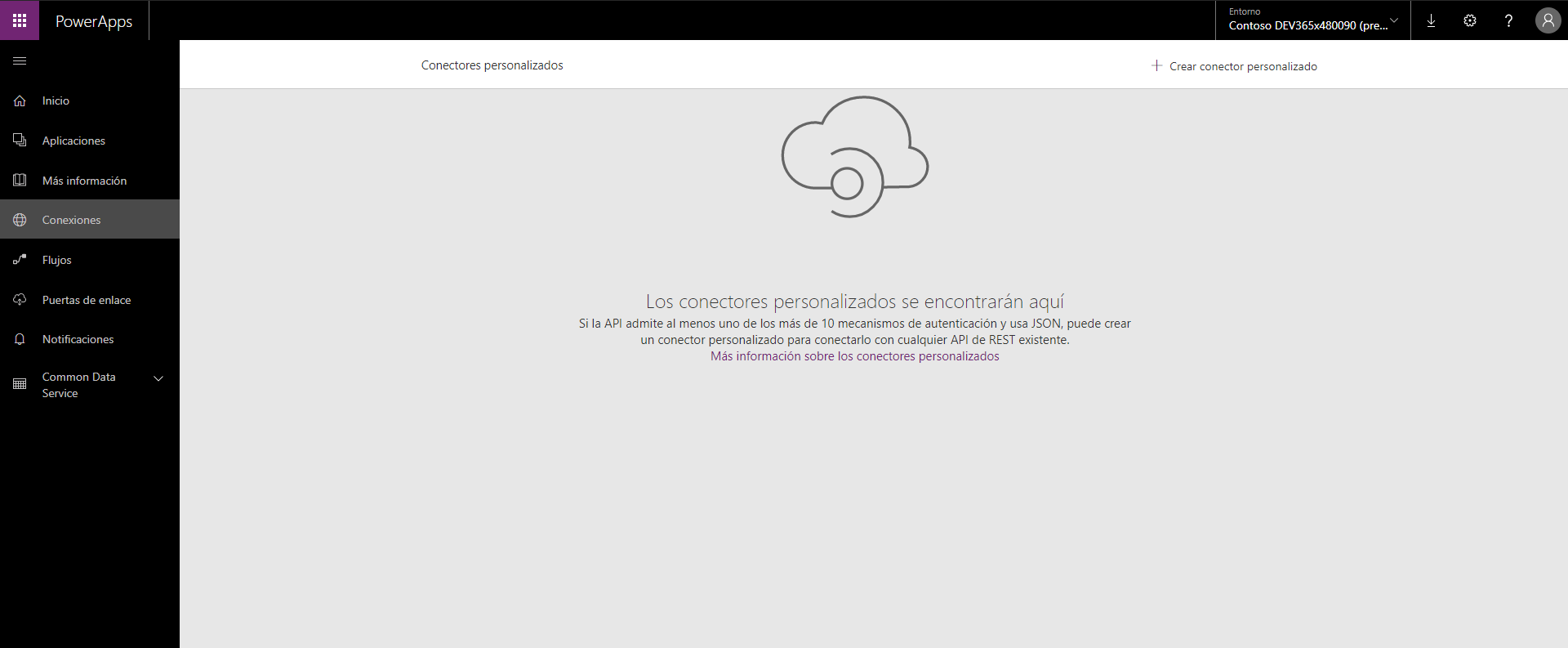
1. Guárdate la información proporcionada por el servicio en un NotePad, por que la necesitaremos mas adelante



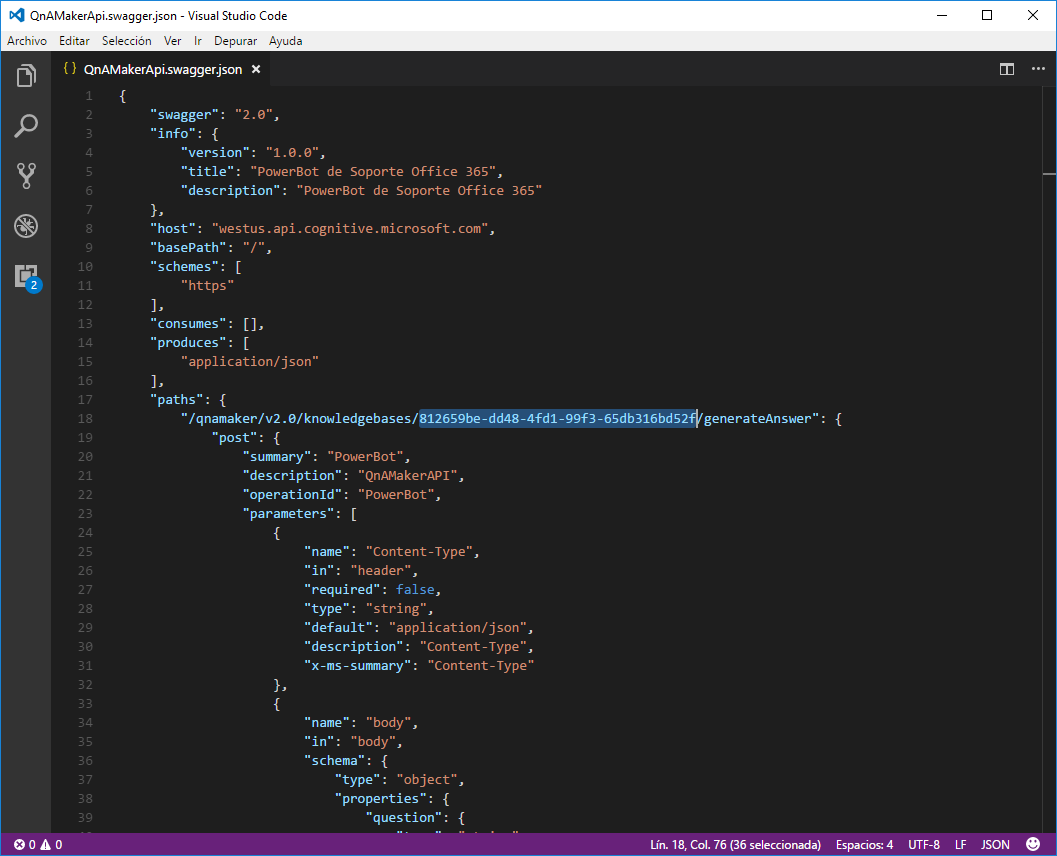
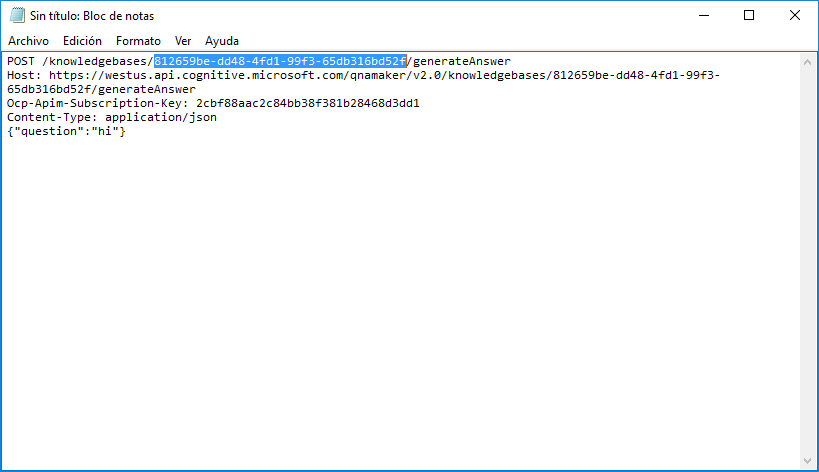
Ejercicio 2: Creación de un conector personalizado en Microsoft PowerApps

En el siguiente ejercicio vamos a crear un conectar personalizado que nos permitirá consumir datos de la API de QnA Maker.

1. Dirígete a <https://web.powerapps.com/> e introduce tus credenciales proporcionadas para el Bootcamp
2. Una vez dentro, accede al apartado de “Conexiones”.



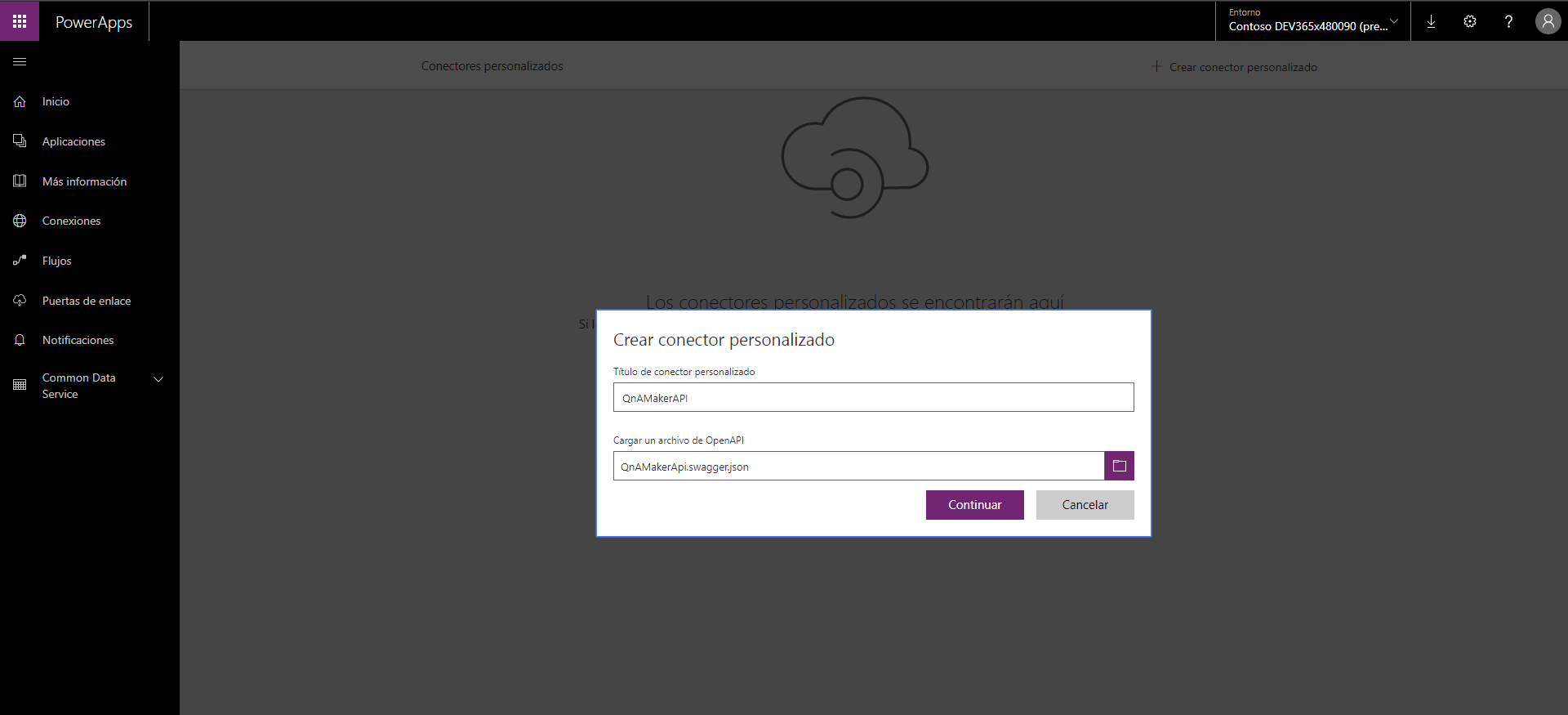
1. En la carpeta del laboratorio buscamos el archivo “QnAMakerApi.swagger” y lo editamos con VSCode o VStudio. Una vez lo tengamos abierto, sustituimos el guid de la imagen por el guid que nos hemos guardado anteriormente en el notepad al crear el servicio y guardamos el archivo con los cambios.



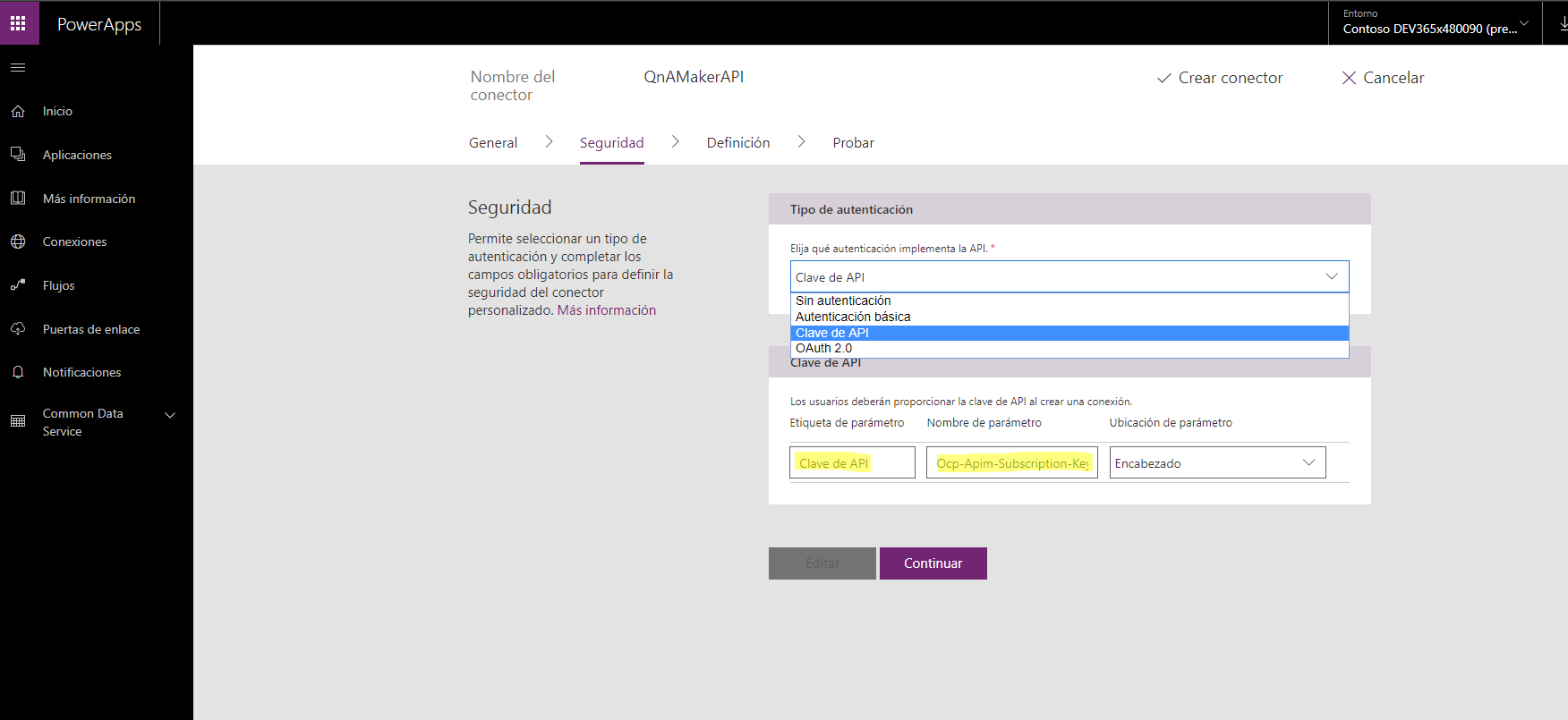
1. Volvemos a la ventana de los conectores personalizados y vamos a seleccionar la opción de Importar un archivo de Open API



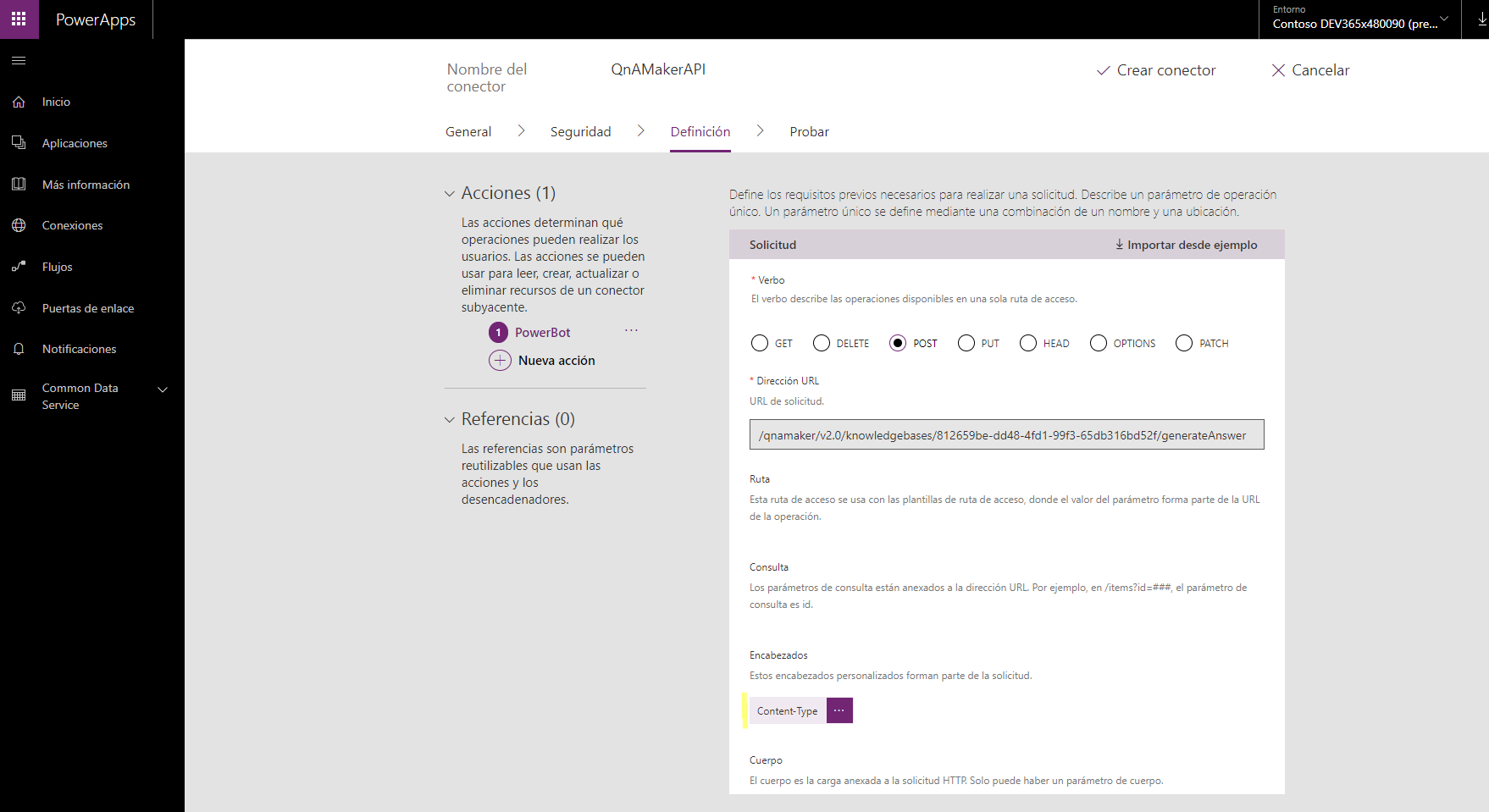
1. Introducimos como nombre “QnAMakerAPI” y adjuntamos el archivo swagger que acabamos de editar



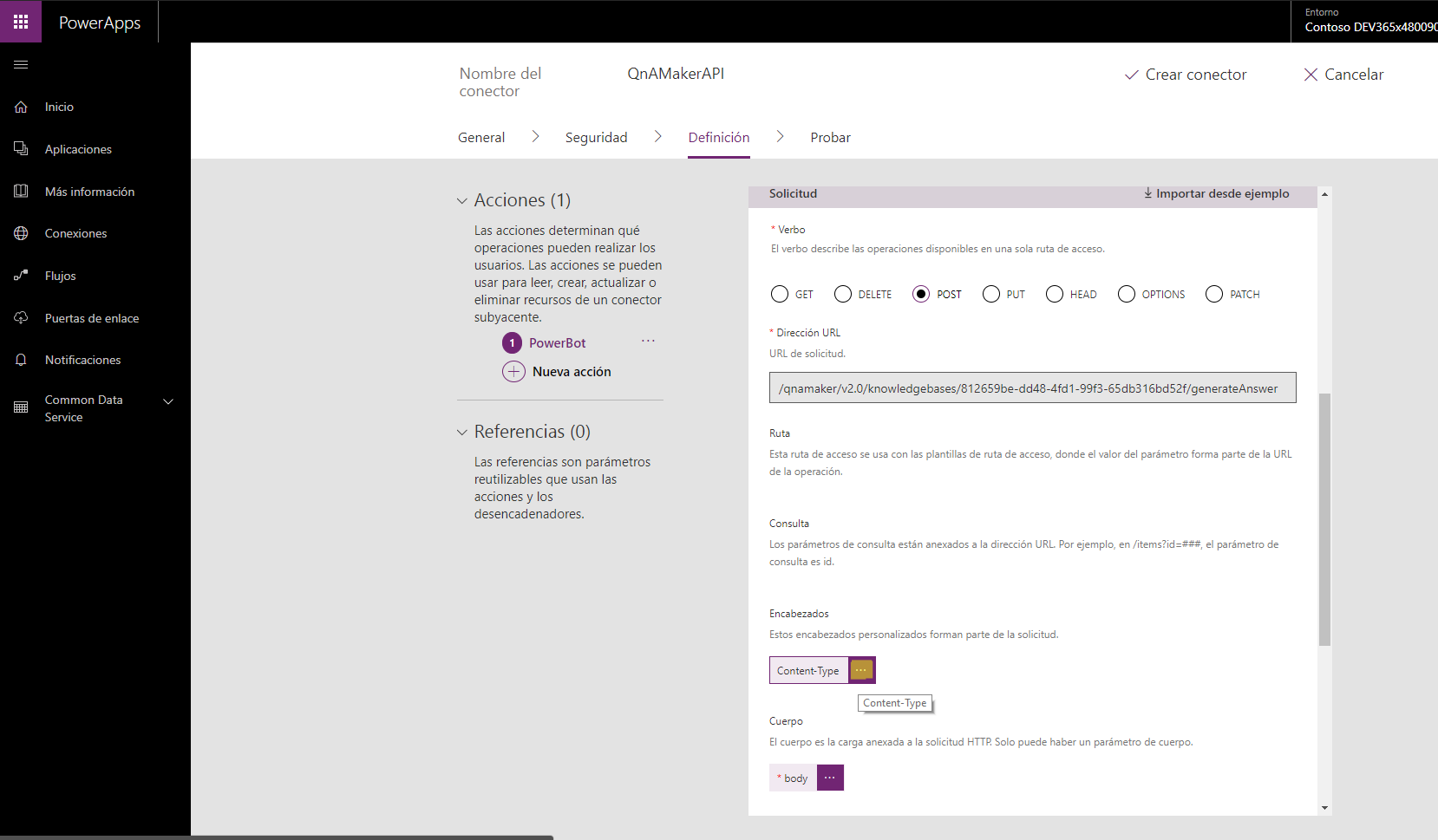
1. En el apartado de Seguridad, seleccionamos Clave de API, como tipo de autenticación y rellenamos los campos de la Clave de API con los valores “Clave de API” y “Ocp-Apim-Subscription-Key” tal y como aparecen en la imagen.



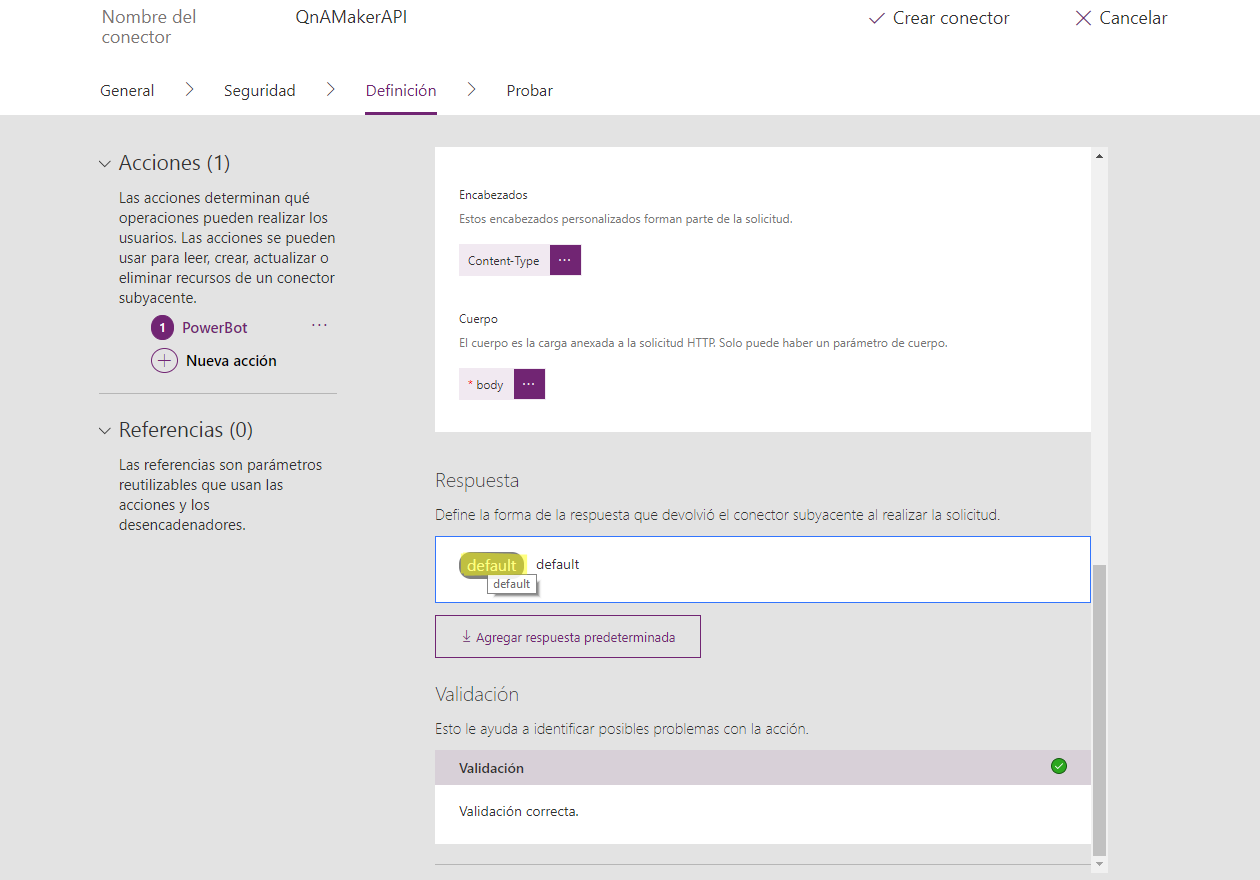
1. En el siguiente paso Definición, en el apartado Solicitud, eliminar el encabezado “Ocp-Apim-Subscription-Key” porque ya lo hemos informado en la pantalla anterior.



1. Una vez eliminado, seleccionar para editar al encabezado “Content-Type” y seleccionar la opción de obligatorio a NO

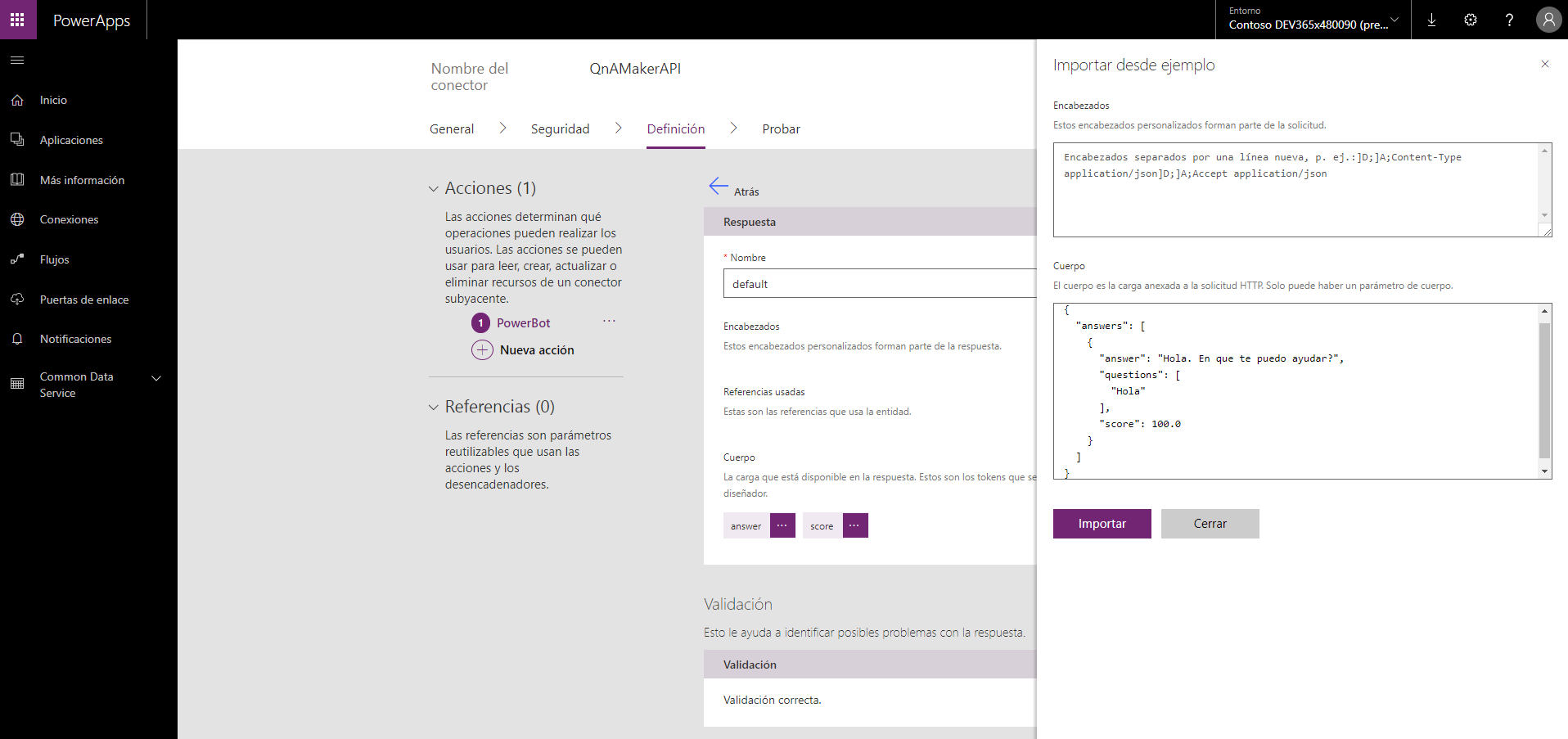


1. En el apartado de Respuesta, selecciona la opción “default”

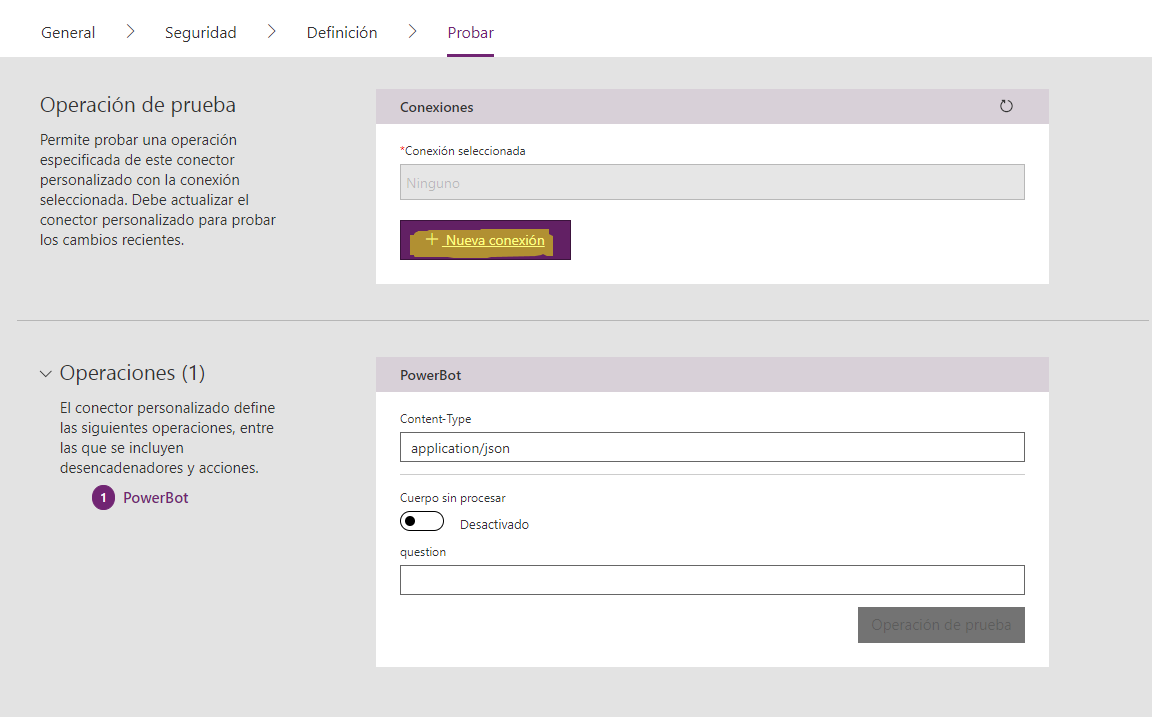


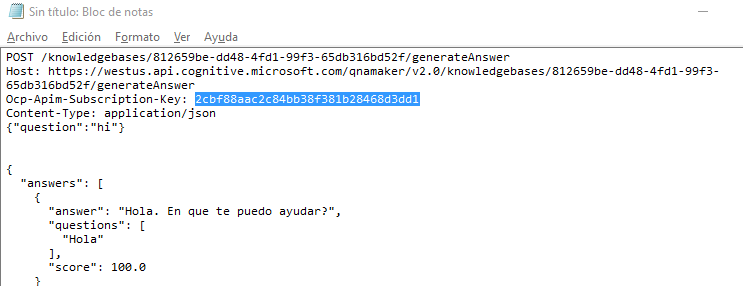
1. Selecciona la opción “Importar desde ejemplo” e introduce el siguiente cuerpo JSON en el apartado Cuerpo

|  |
| --- |
| {  "answers": [  {  "answer": "Hola. En que te puedo ayudar?",  "questions": [  "Hola"  ],  "score": 100.0  }  ]  } |

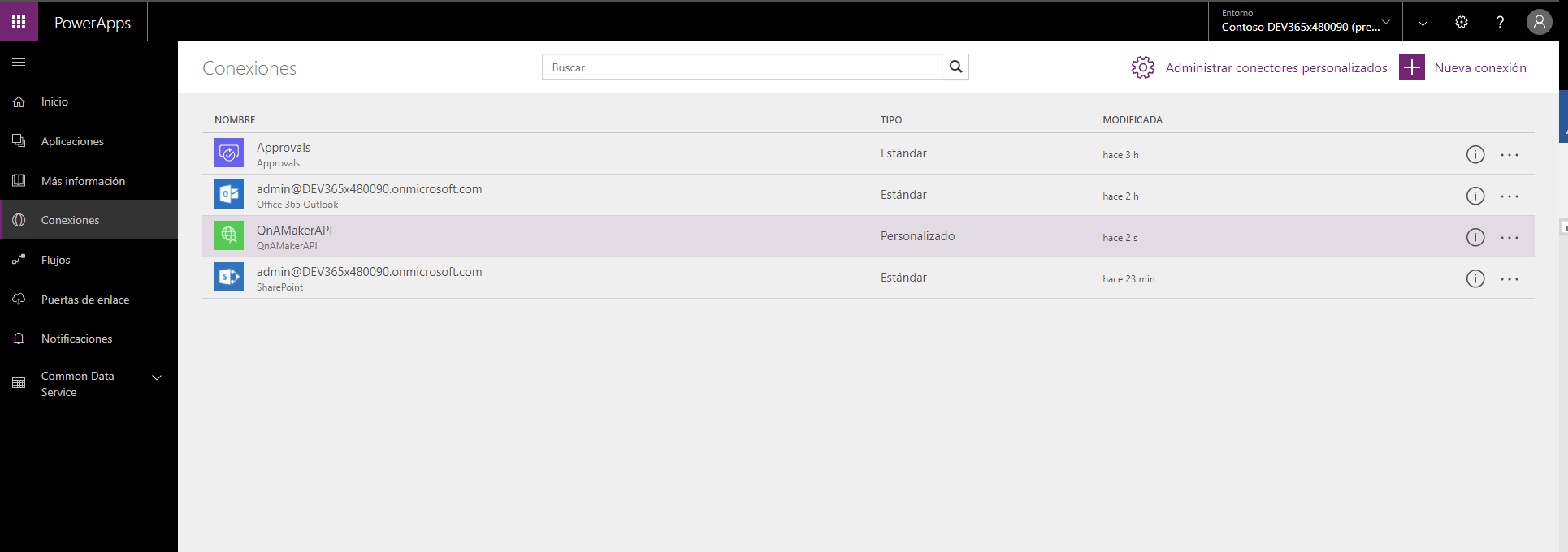


1. Seleccionamos Crear el conector. Una vez nos haya confirmado que se ha creado. Nos dirigimos al apatado “Probar” y creamos una nueva conexión. Nos pedirá una clave de API. Introducimos la clave de API que nos hemos guardado antes en el notepad al crear el servicio.

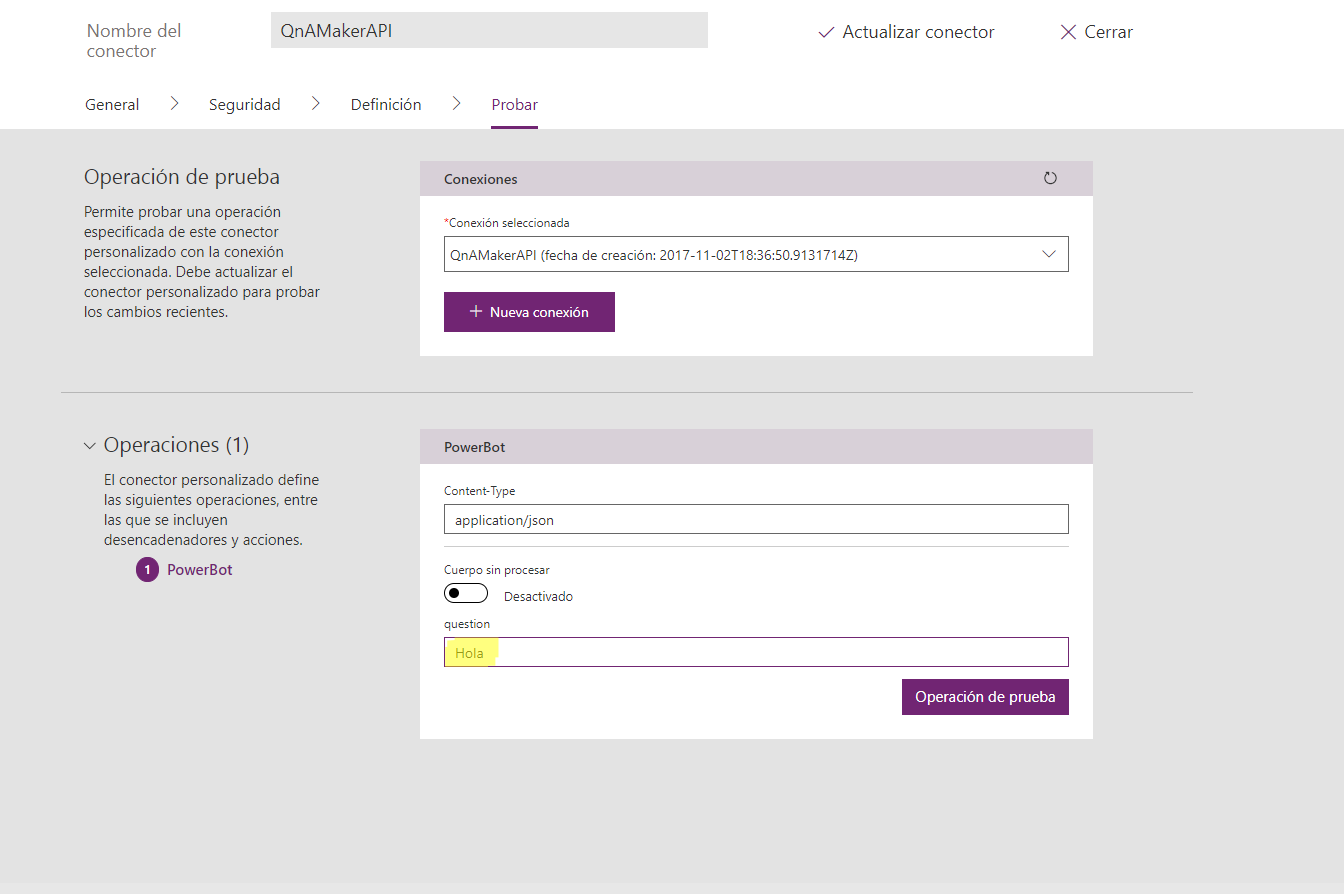




1. Vemos como se nos ha agregado una nueva conexión a Microsoft PowerApps.



1. Seleccionamos de nuevo la opción “Administrar conectores personalizados” y editamos nuestro conector. Nos dirigimos al apartado Probar y probamos nuestro conector indicándole la palabra “Hola”



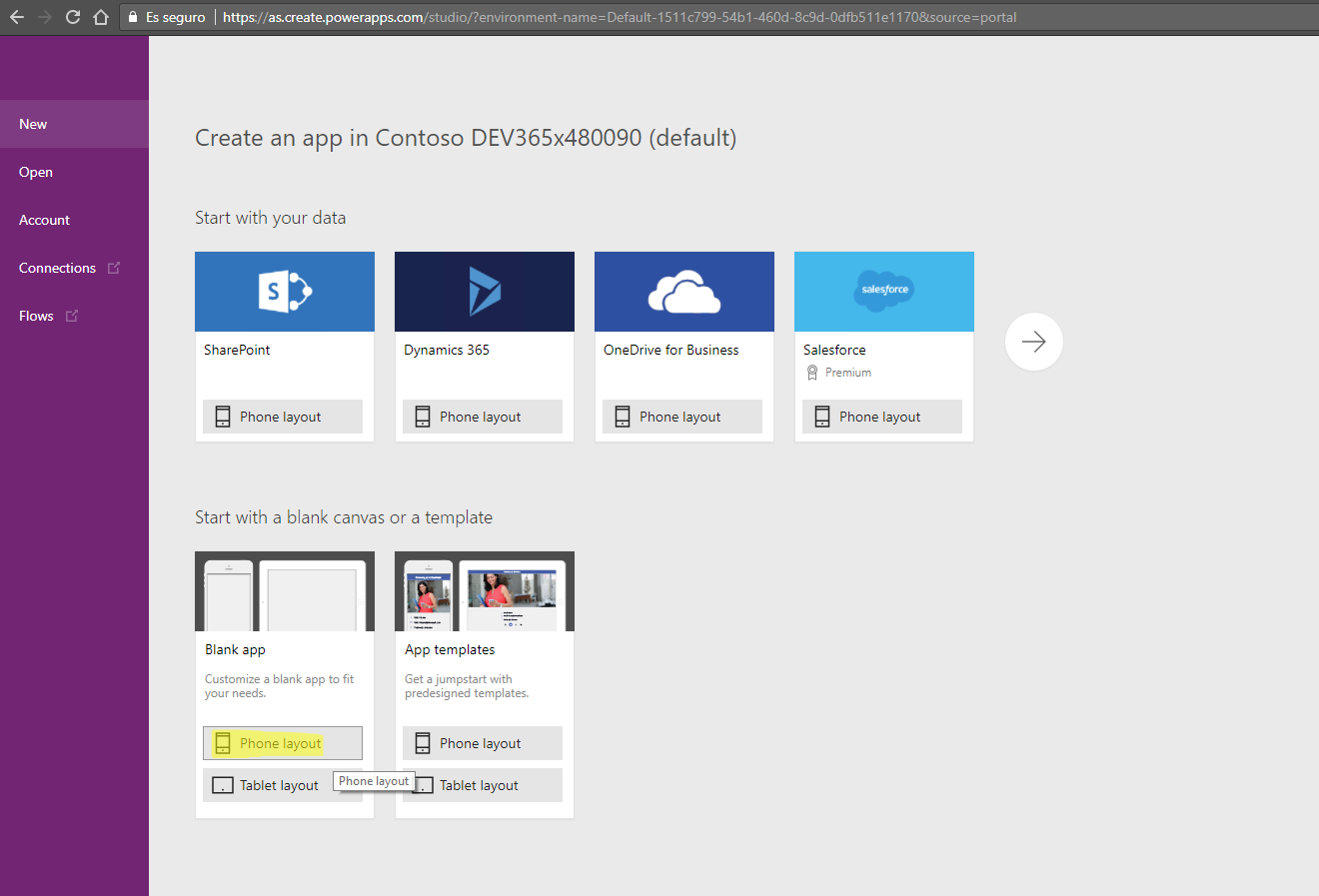
1. Si todo está correcto, el conector nos devolverá el cuerpo del mensaje de forma satisfactoria



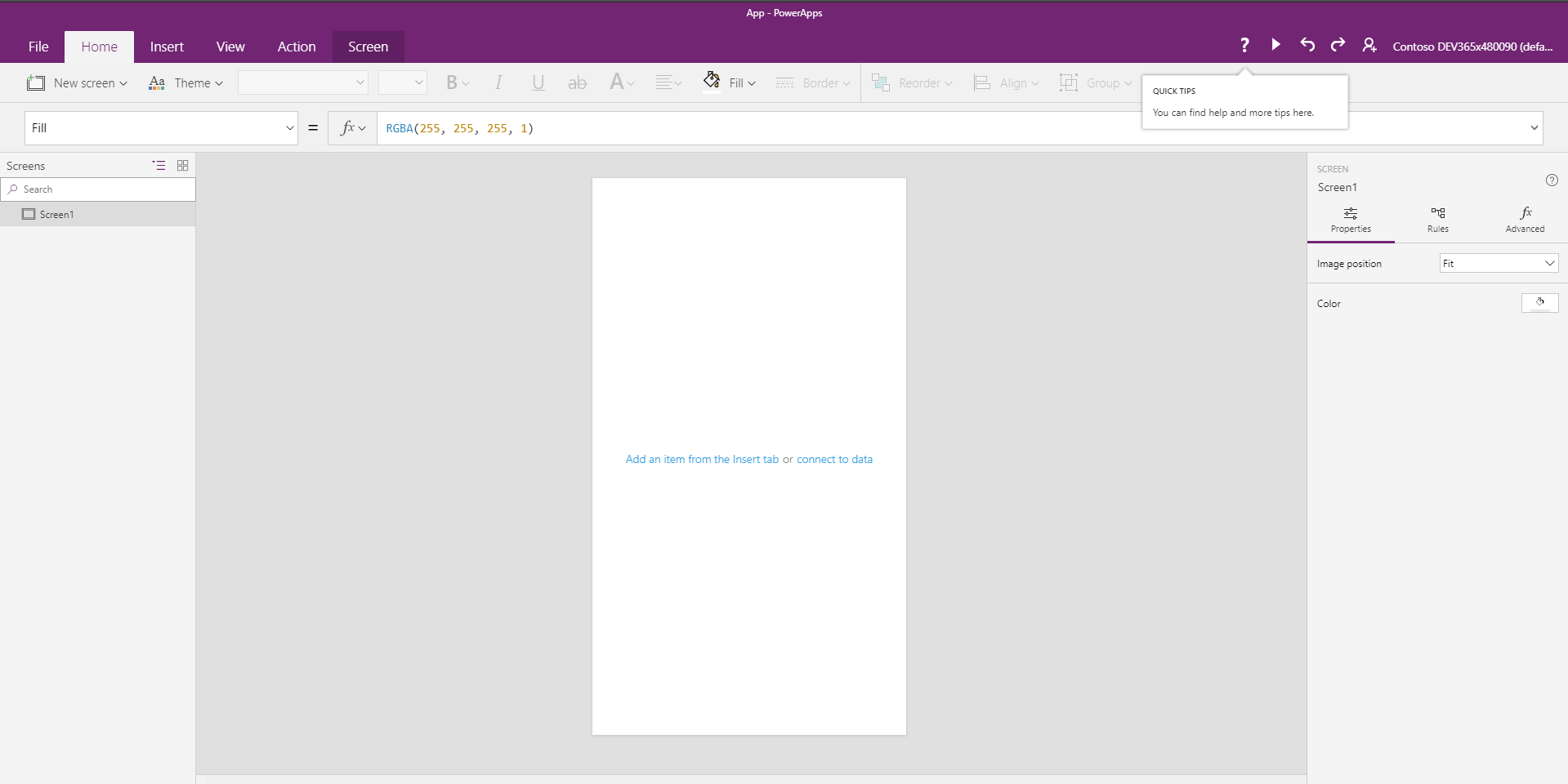
Ejercicio 3: Creación de una aplicación en Microsoft PowerApps

En el siguiente ejercicio veremos cómo podemos crear una aplicación de Microsoft PowerApps que consuma la API de QnAMaker mediante el conector personalizado que acabamos de crear.

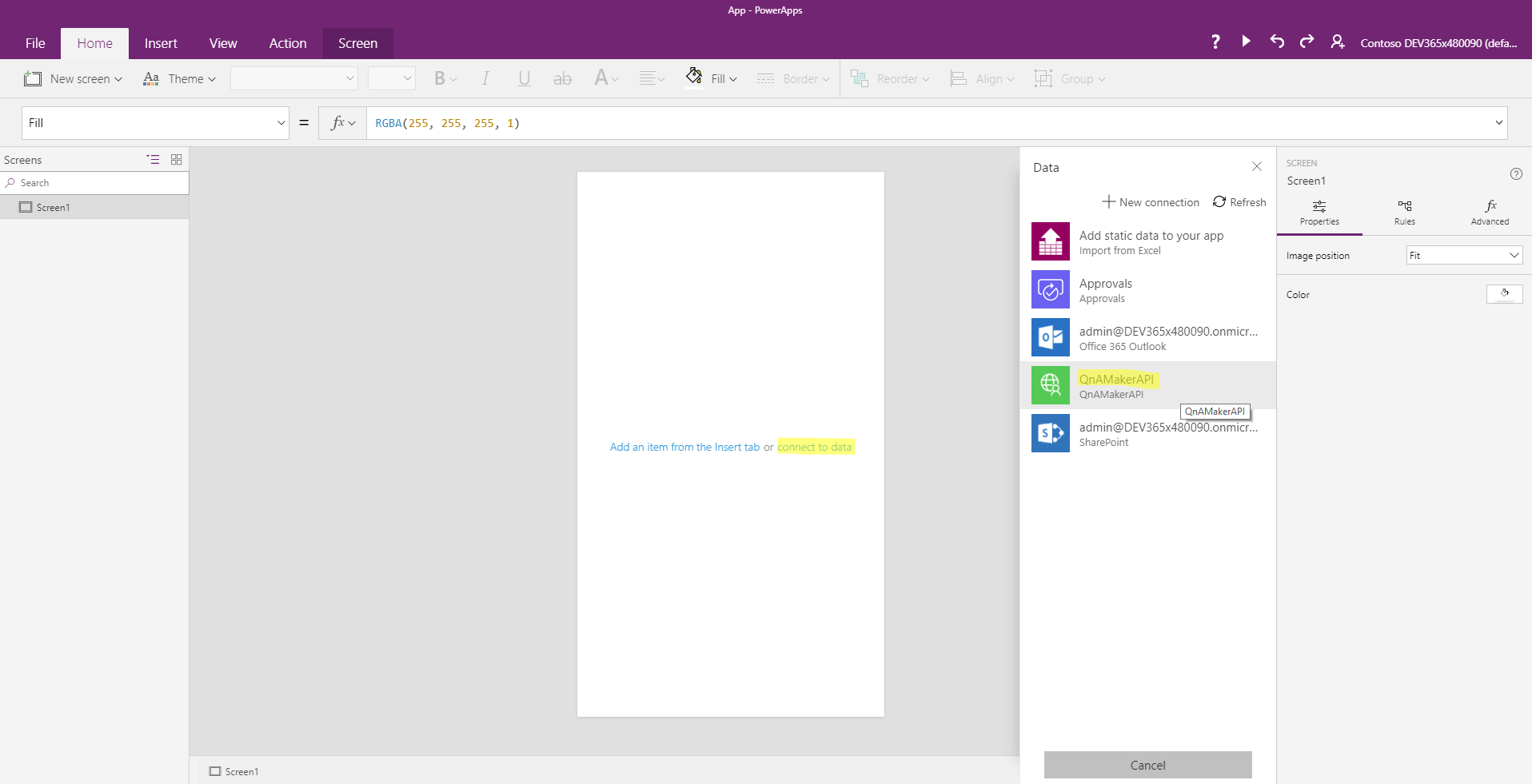
1. Vamos a acceder al apartado de “Aplicaciones” en el portal de PowerApps. <https://web.powerapps.com/>
2. Seleccionamos “Crear una Aplicación”.
3. Seleccionamos crear una aplicación móvil vacía.



1. Se nos abrirá el diseñador de PowerApps con una pantalla en blanco

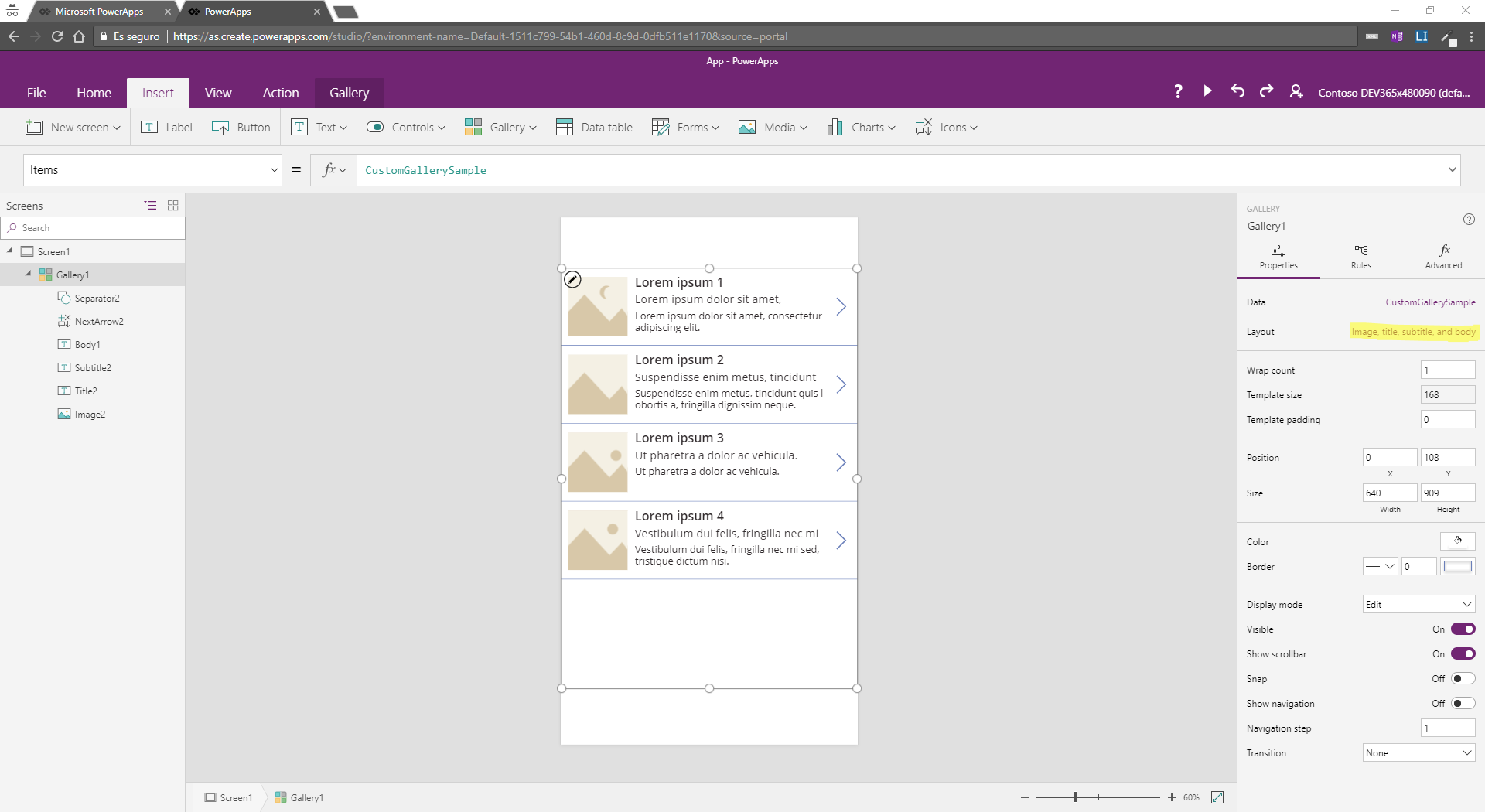


1. Lo primero de todo, vamos a agregar nuestro conector personalizado a nuestra aplicación. Para ello, seleccionamos sobre la opción “Connect to data” y elegimos nuestro conector.

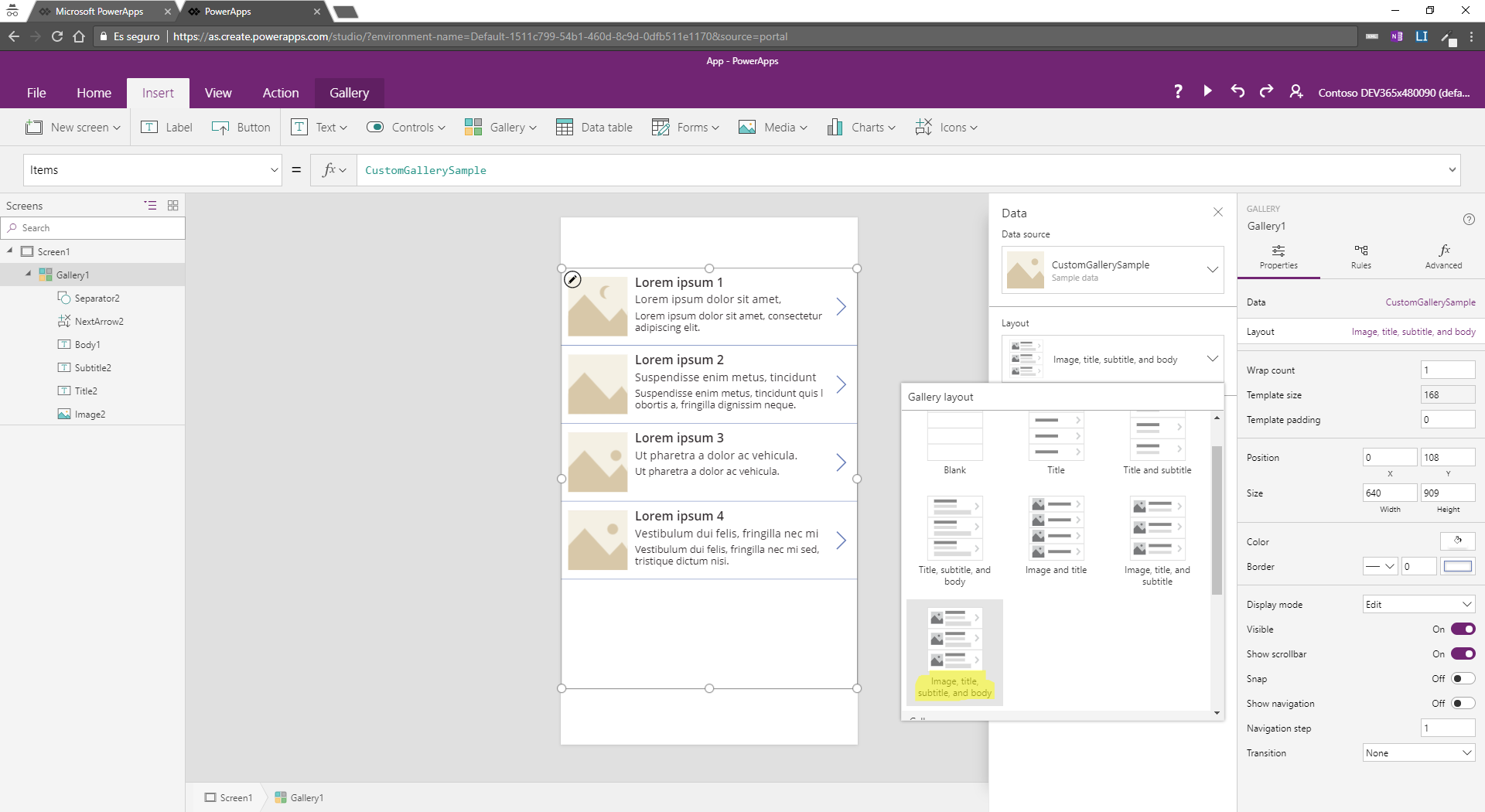


1. Una vez lo hayamos añadido, cerramos la ventana de “Data” y vamos a añadir controles a la pantalla.
2. Lo primero que debemos añadir es una galería vertical, para vamos a la pestaña “Insert” y elegimos “Gallery”
3. Una vez que tengamos la galería agregada a nuestra pantalla de la aplicación, seleccionamos la propiedad

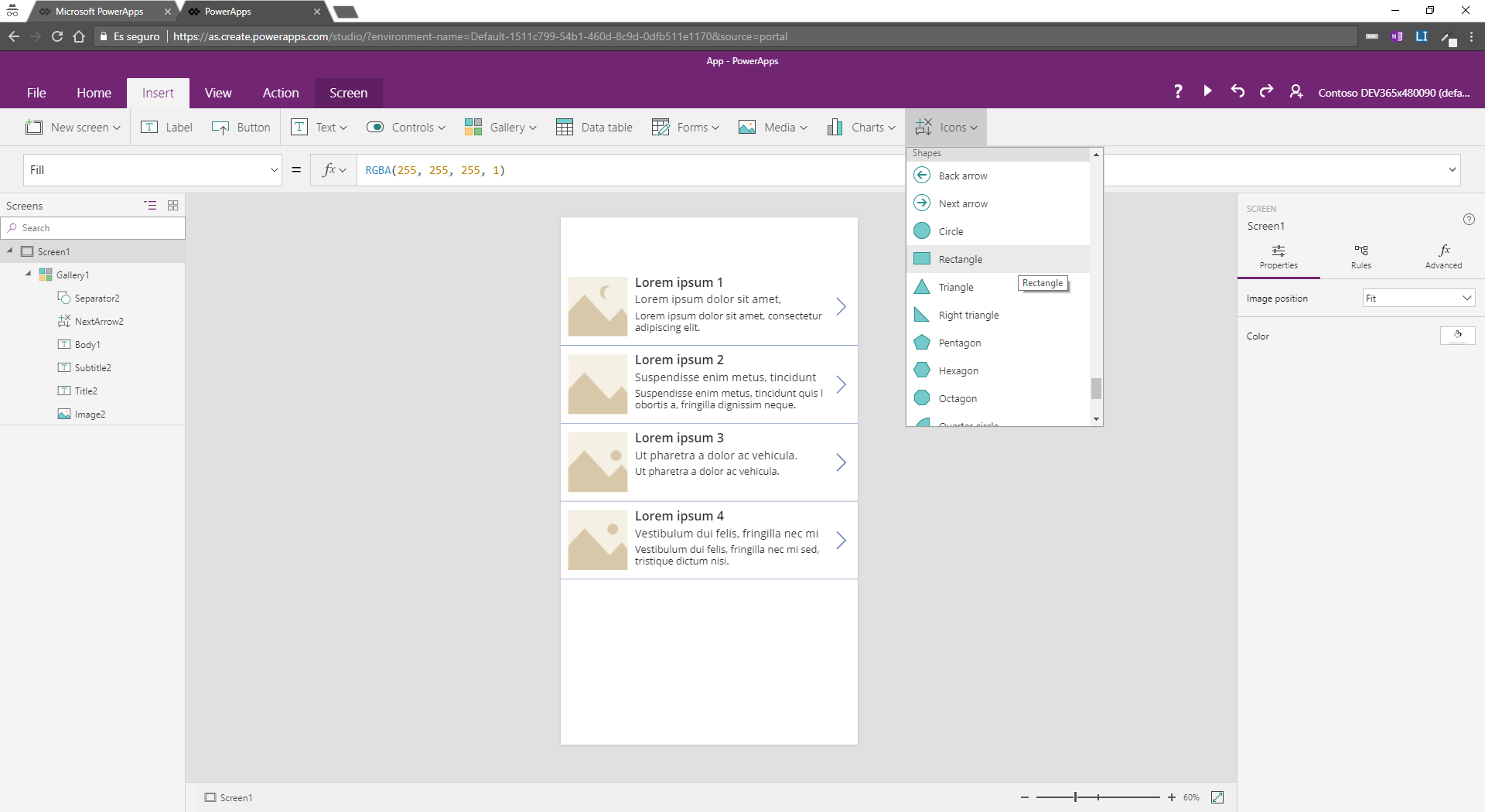
Layout y elegimos un layout diferente.



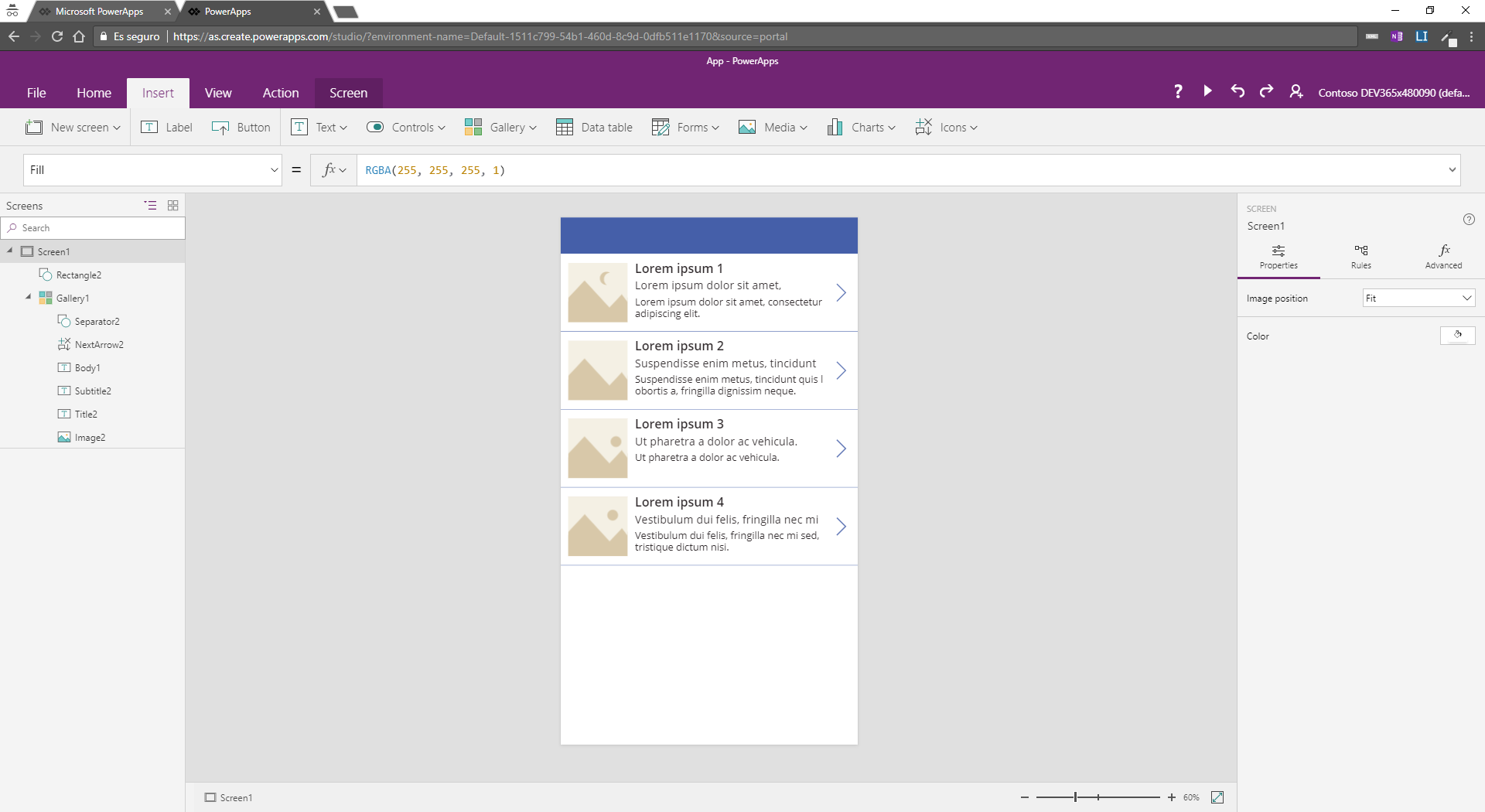
1. Elegimos el layout que aparece en la imagen



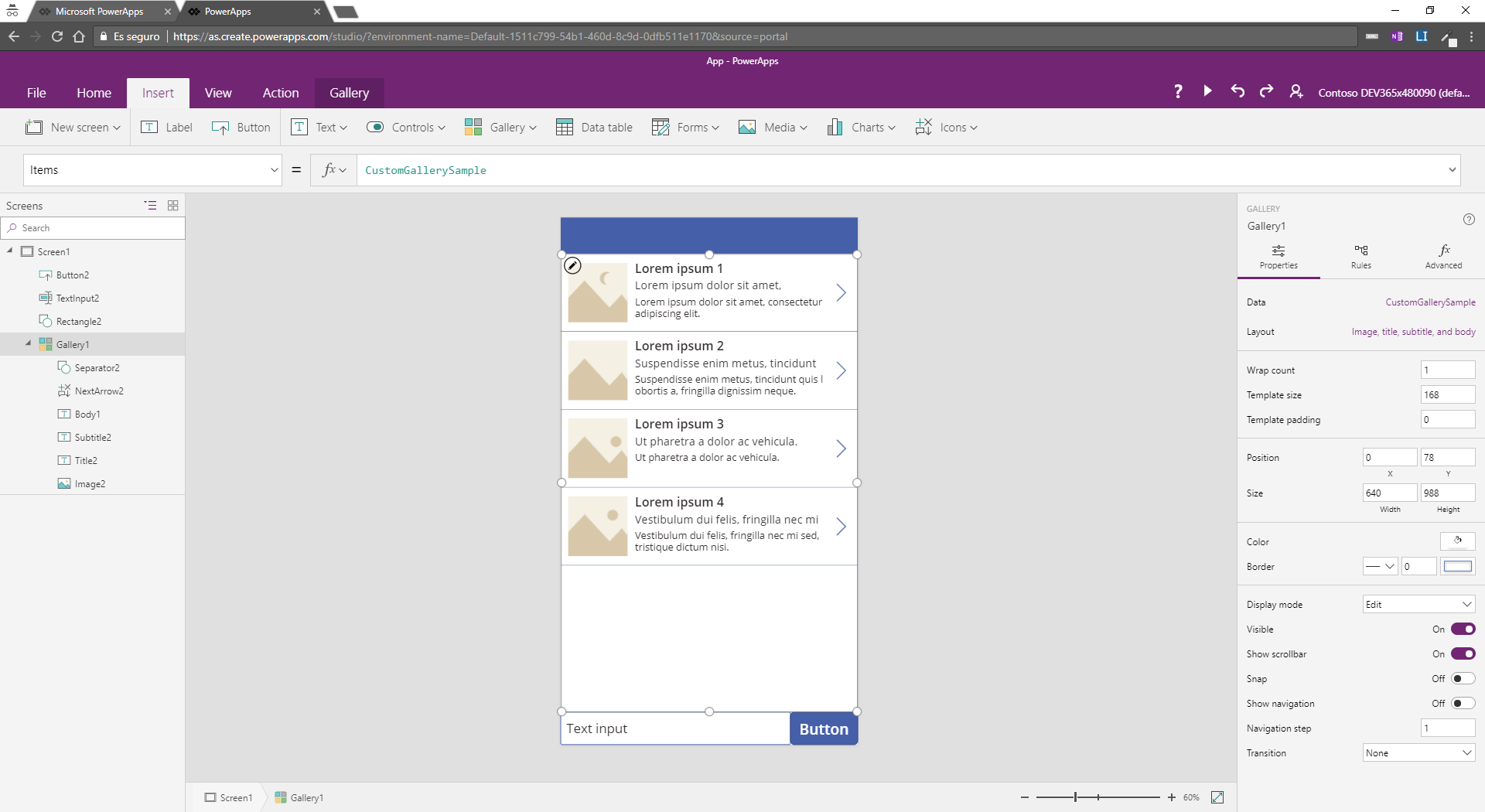
1. Agregamos una barra rectangular en la parte superior, Para ello seleccionamos el icono “Rectangulo” en la opción de “Icons” de la barra de controles



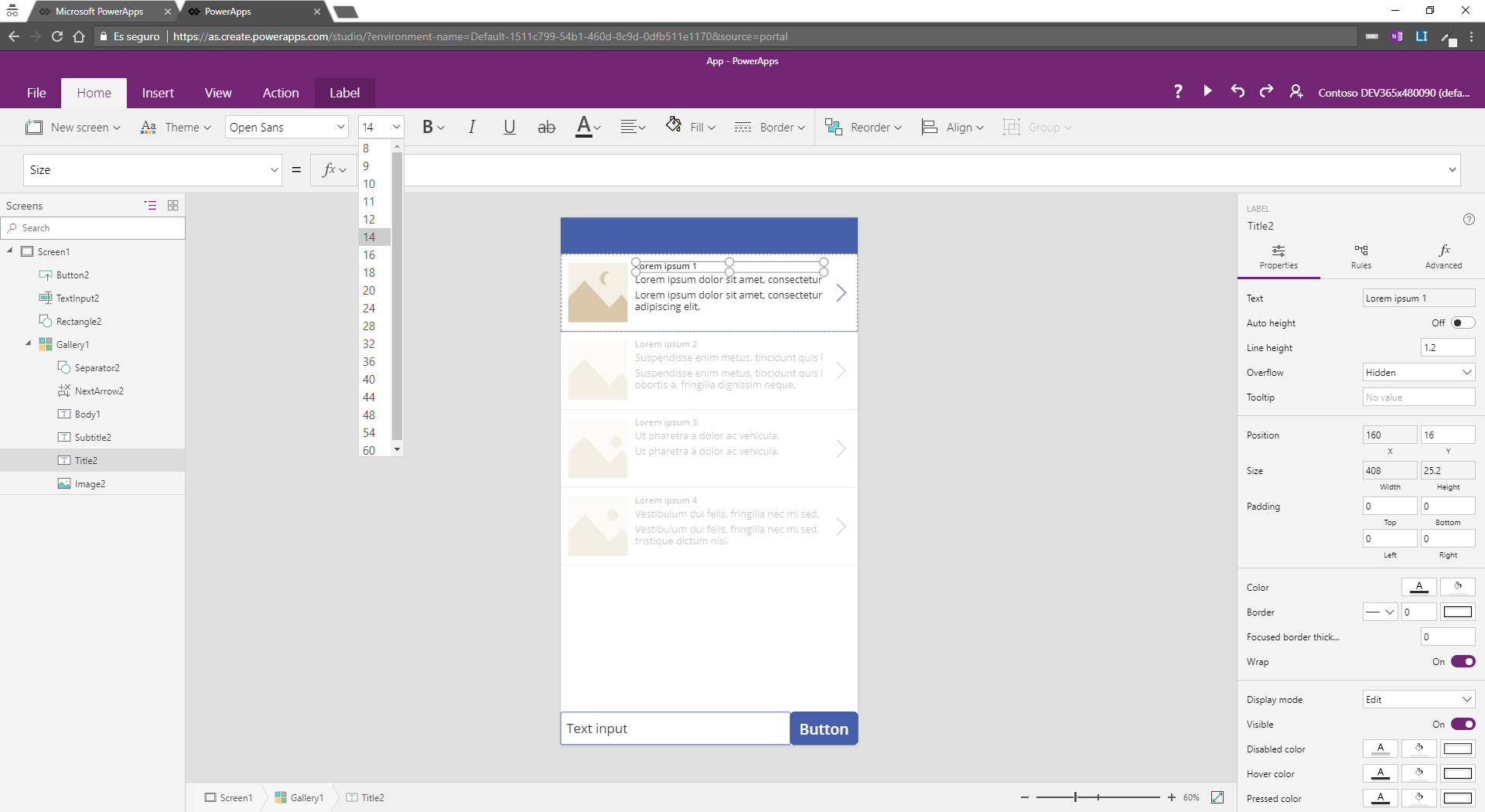
1. El aspecto final tendrá que ser así:



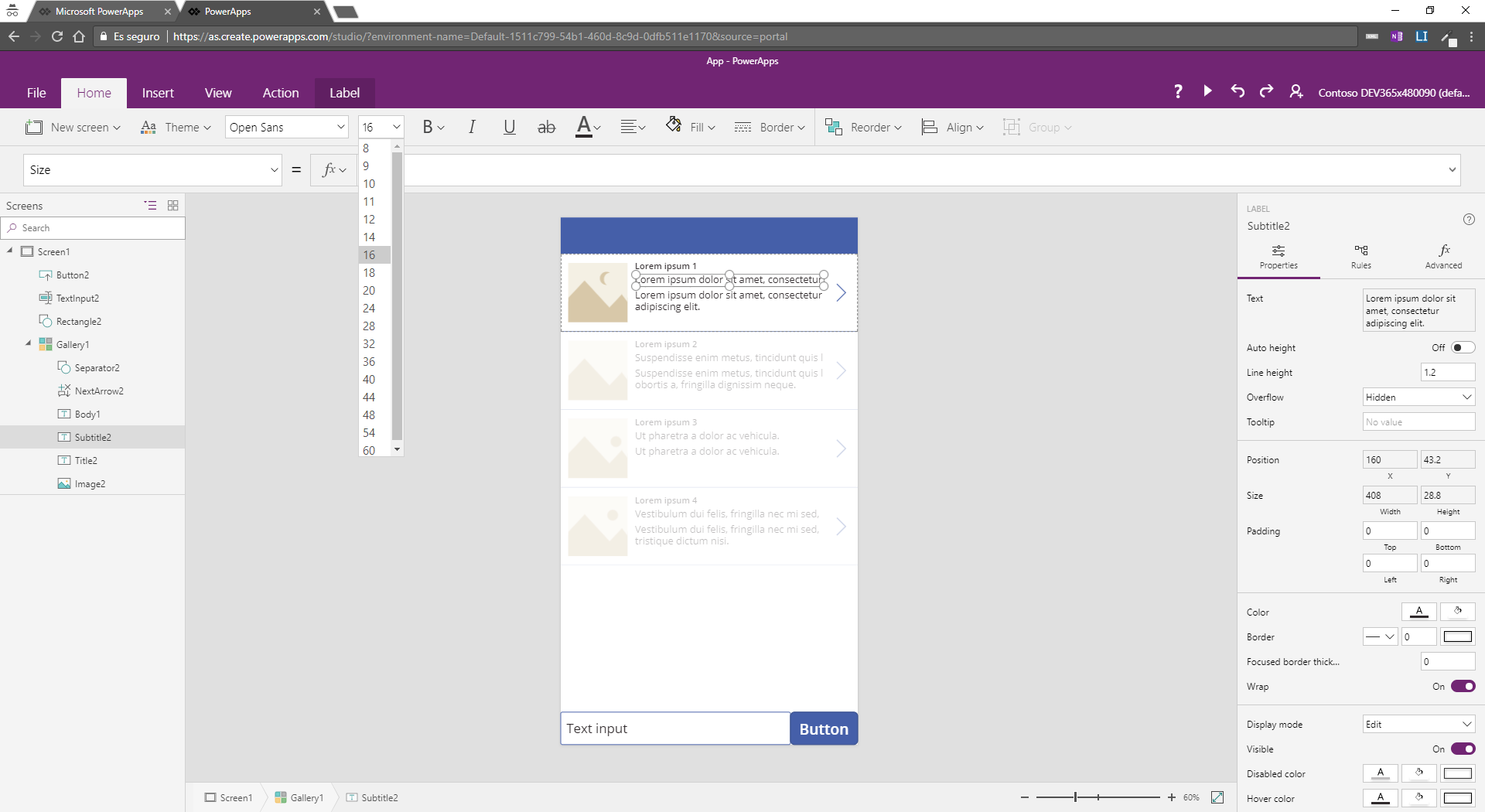
1. Agregamos un “Input Text” y un “Button”, el aspecto final debería ser algo así:



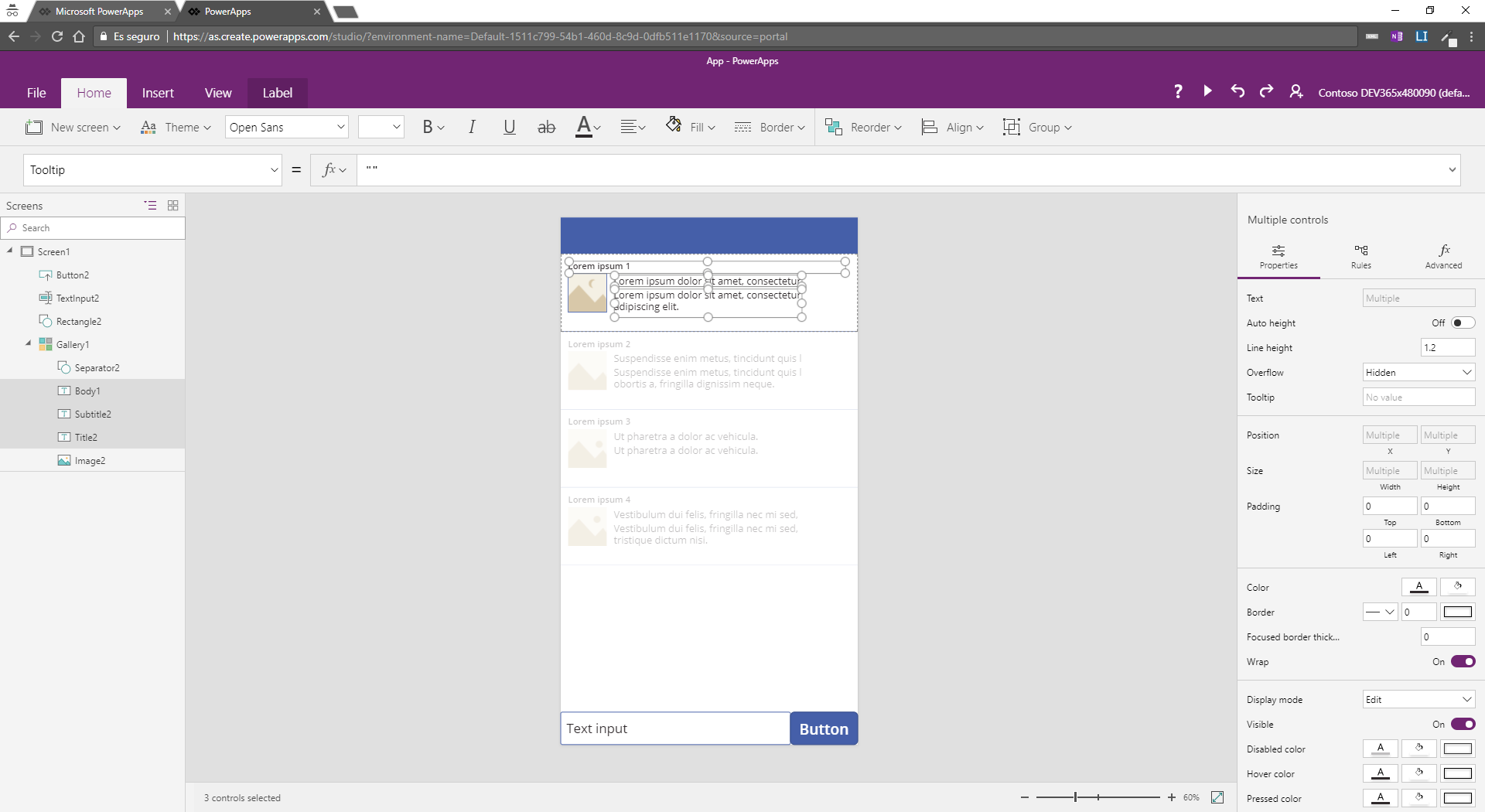
1. Seleccionamos el control Title y bajamos el tamaño de la fuente a 14



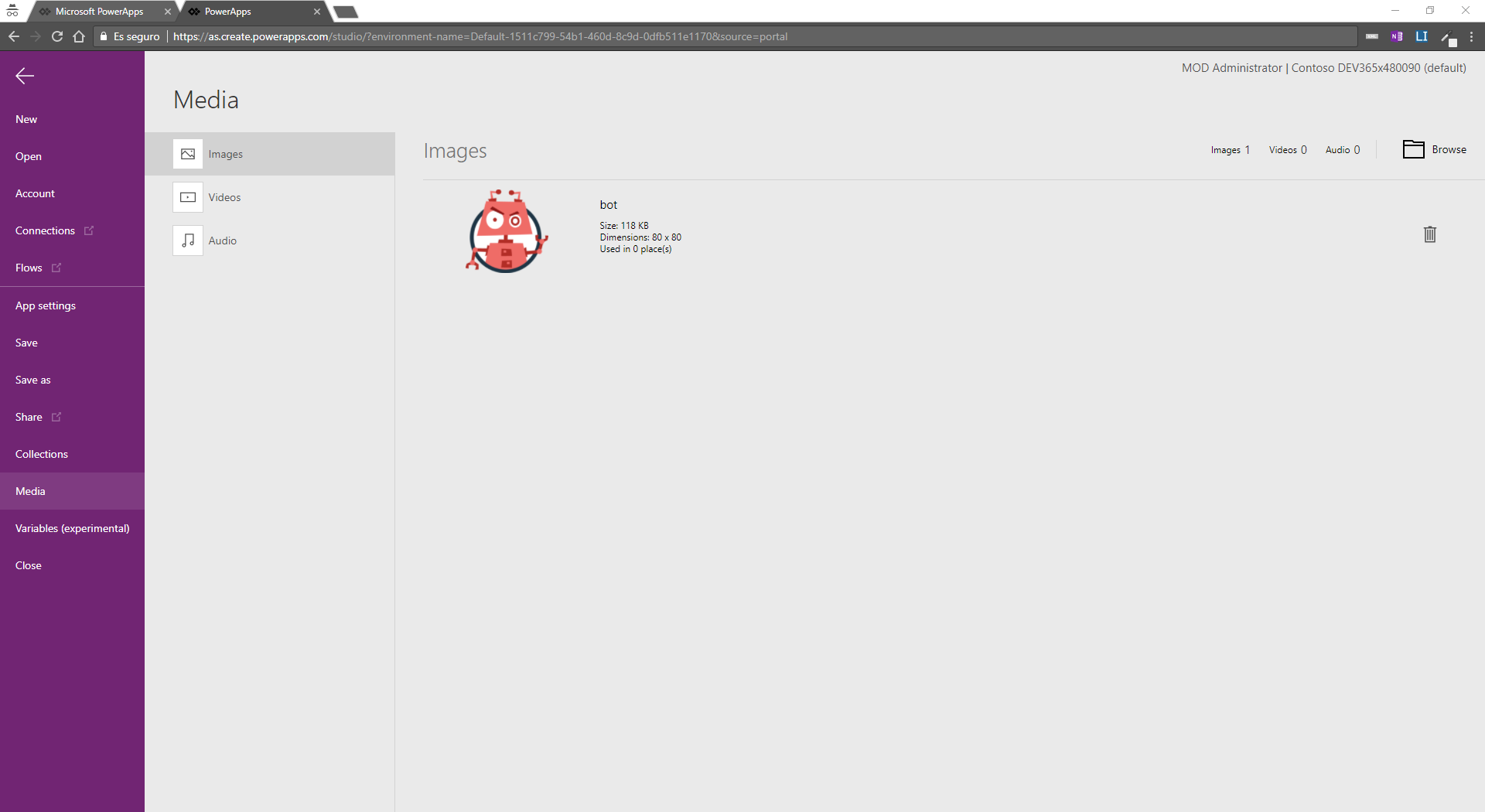
1. Seleccionamos la propiedad Subtitle y bajamos el tamaño de fuente a 16



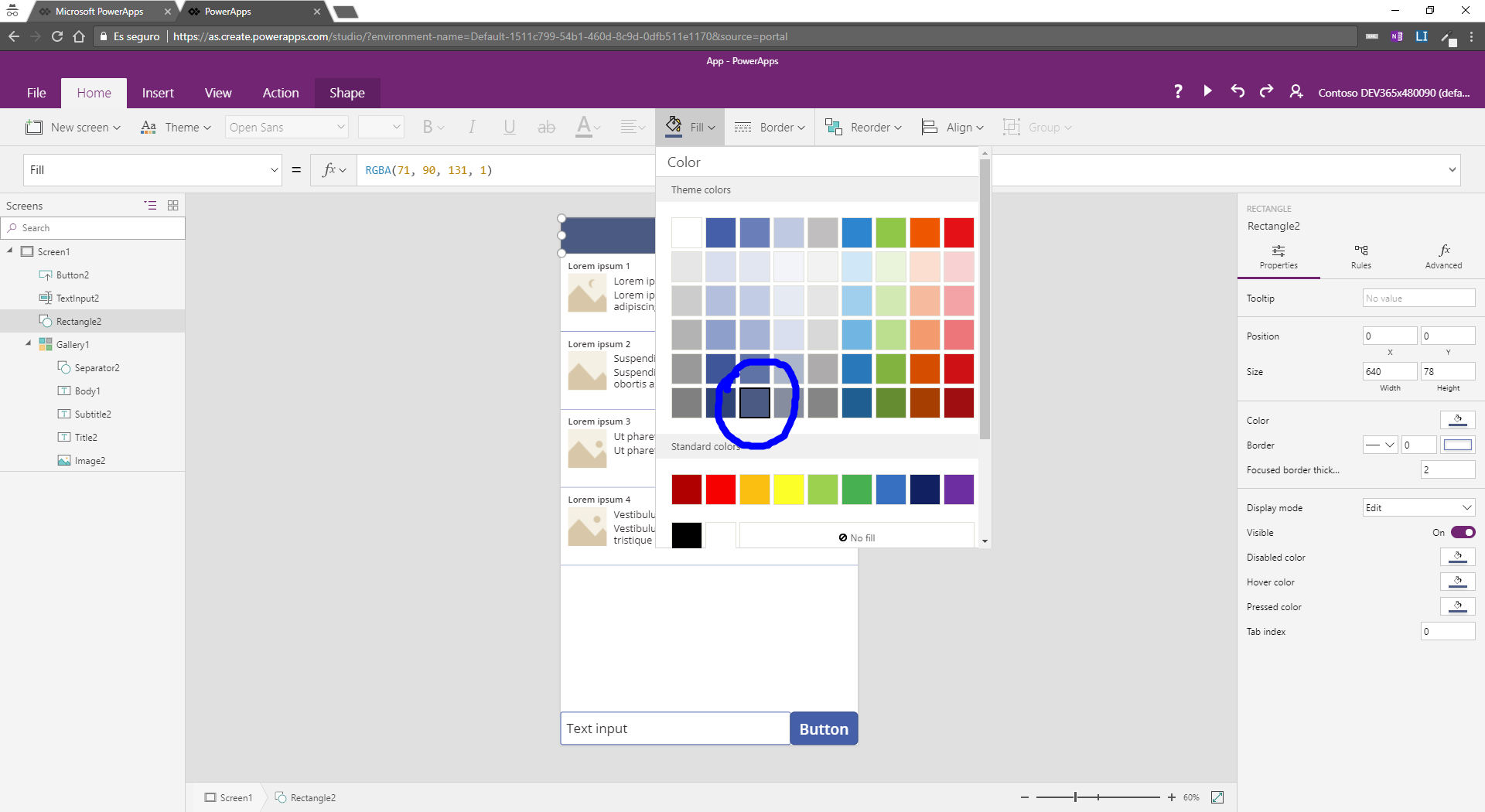
1. Hacemos la imagen más pequeña y alargamos la propiedad Title por encima de ella. Colocamos las propiedades Title, Subtitle, y Body más centradas como aparecen en la imagen y suprimimos la propiedad NextArrow. El aspecto final debería ser este:



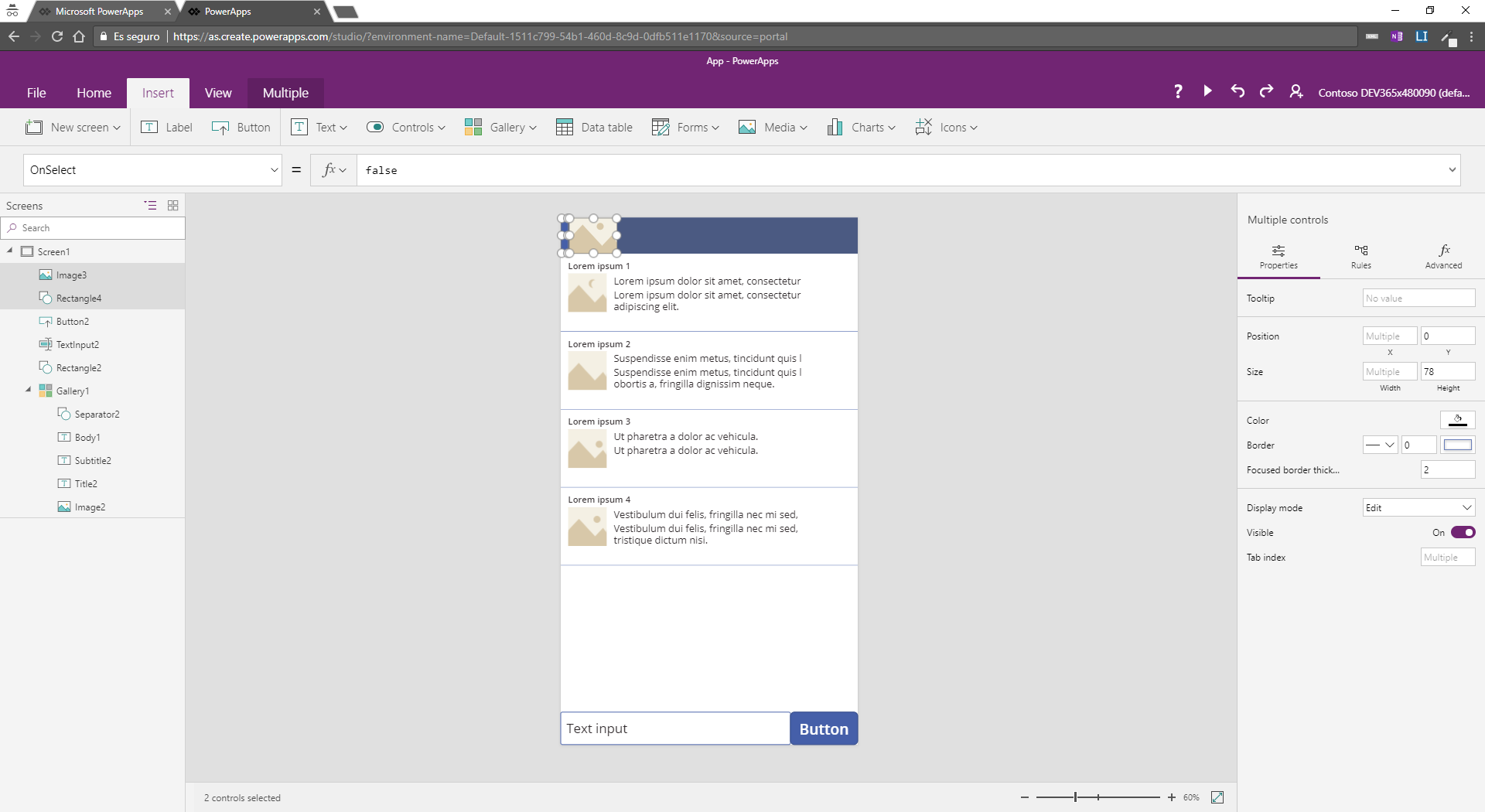
1. Seleccionamos la opción “File” en el menú superior, y accedemos a la sección “Media”. Pulsamos sobre “Browse” y adjuntamos el archivo “bot.png” de la carpeta del laboratorio



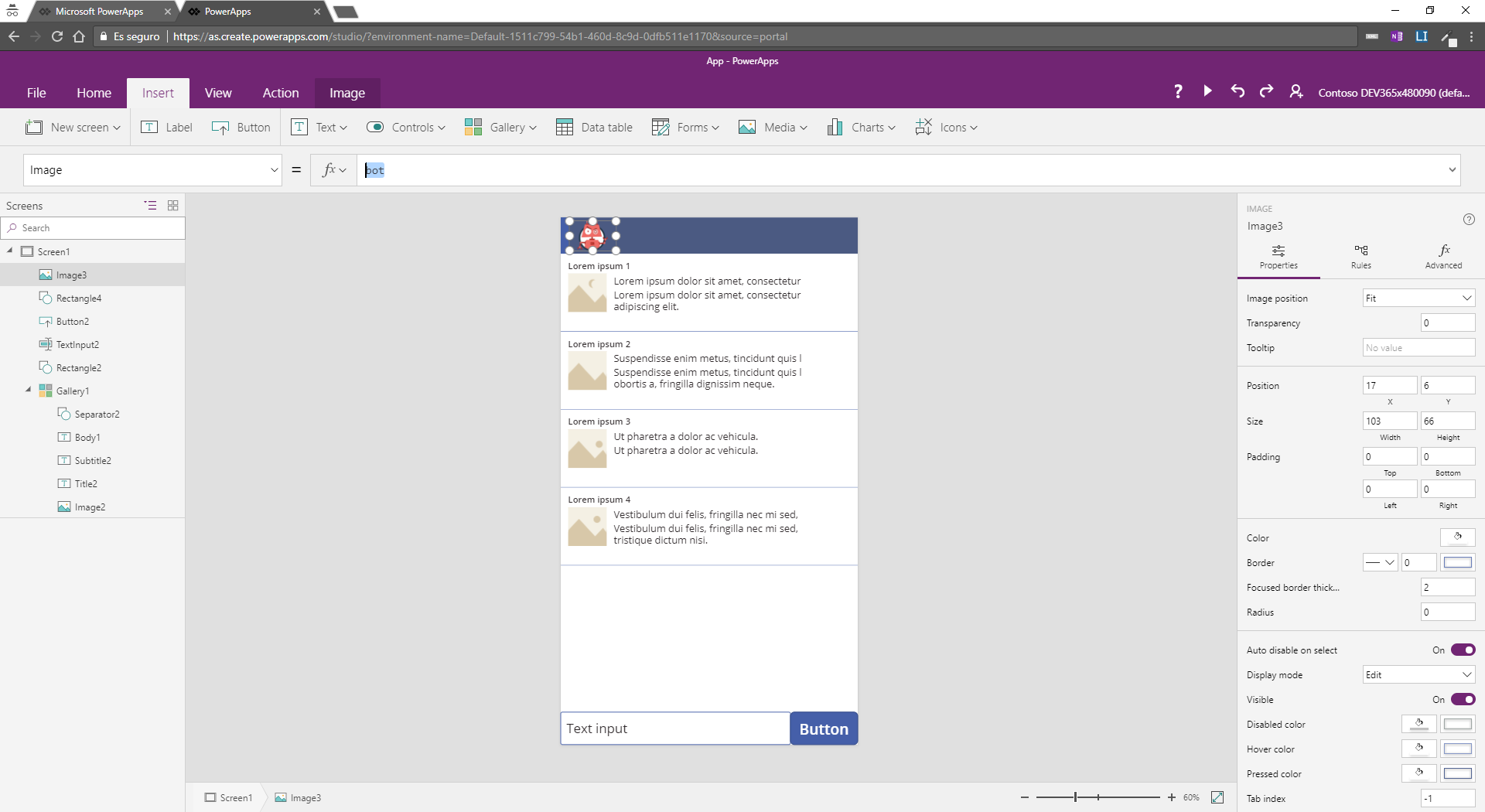
1. Volvemos a la pantalla de nuestra aplicación y seleccionamos el Rectangulo Superior. Cambiamos el color del Rectangulo



1. Agregamos un Rectangulo y una imagen, dando la siguiente apariencia a nuestra pantalla



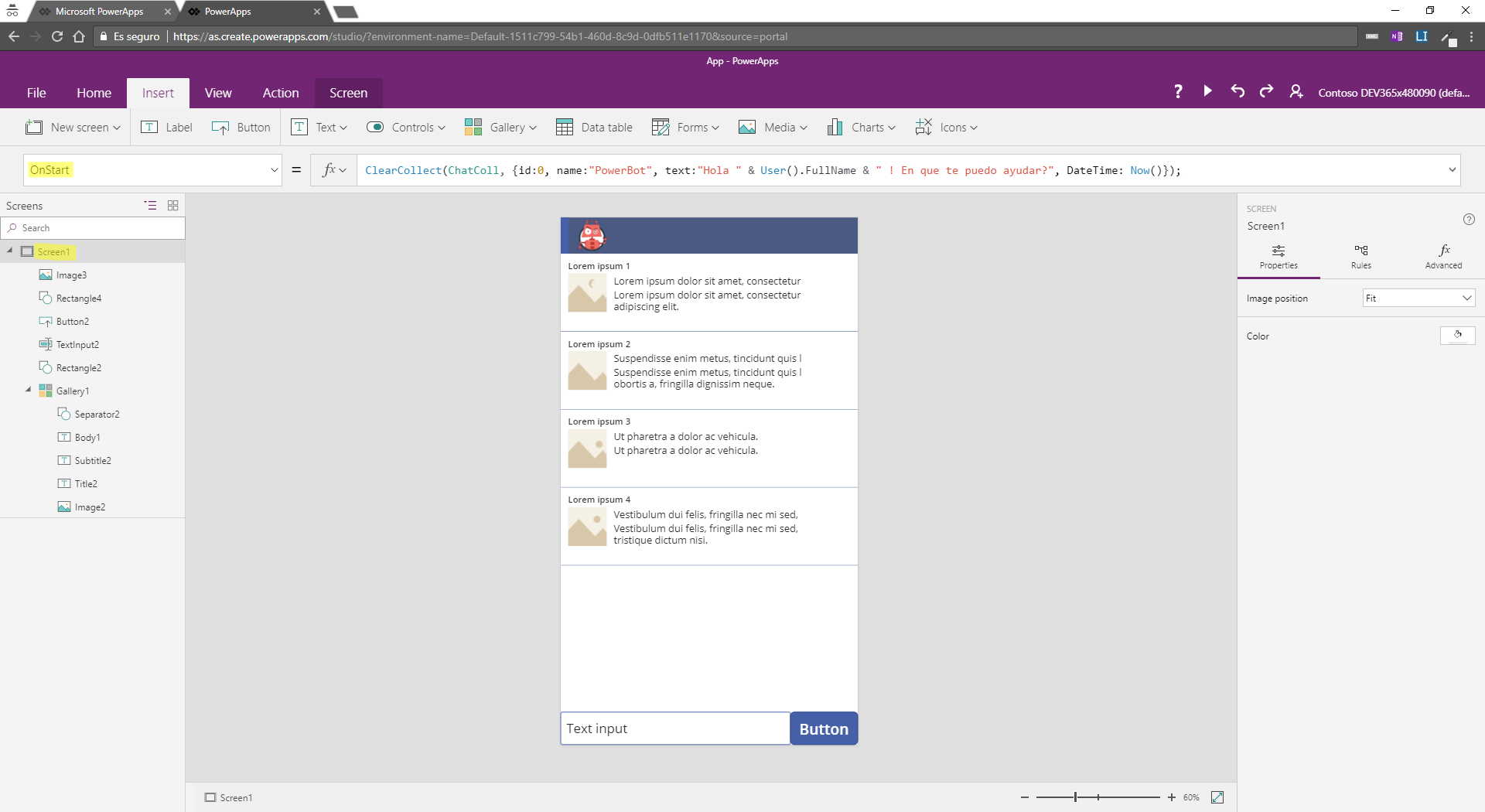
1. En la propiedad “Image” de nuestra imagen, escribir el nombre de la imagen que hemos agregado anteriormente en el apartado de Media



1. Seleccionamos la pantalla Screen1 y en la propiedad OnStart escribimos la siguiente función:

ClearCollect(ChatColl, {id:0, name:"PowerBot", text:"Hola " & User().FullName & " ! En que te puedo ayudar?", DateTime: Now()});

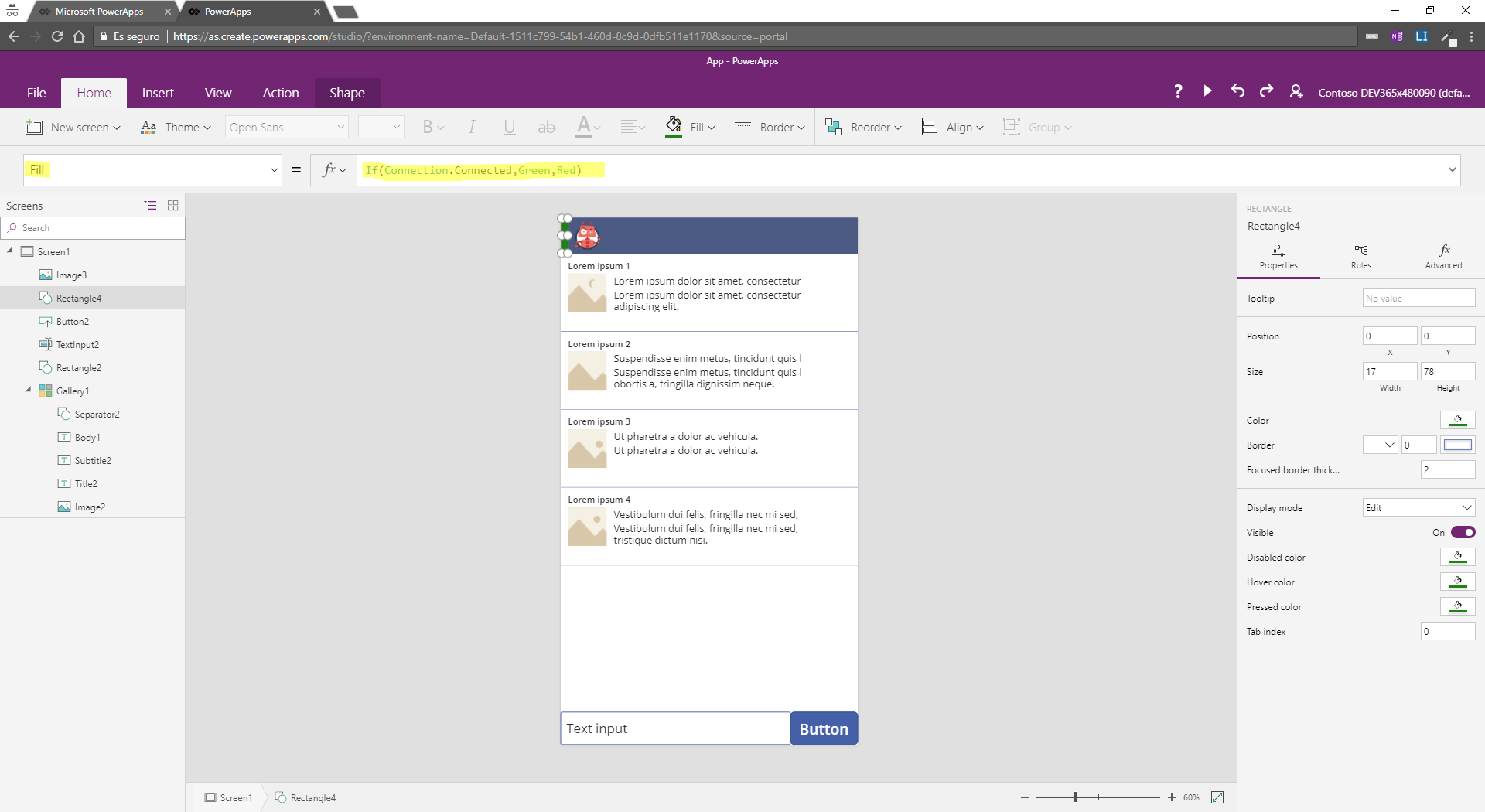
Set(ConversationOrder, SortOrder.Descending)



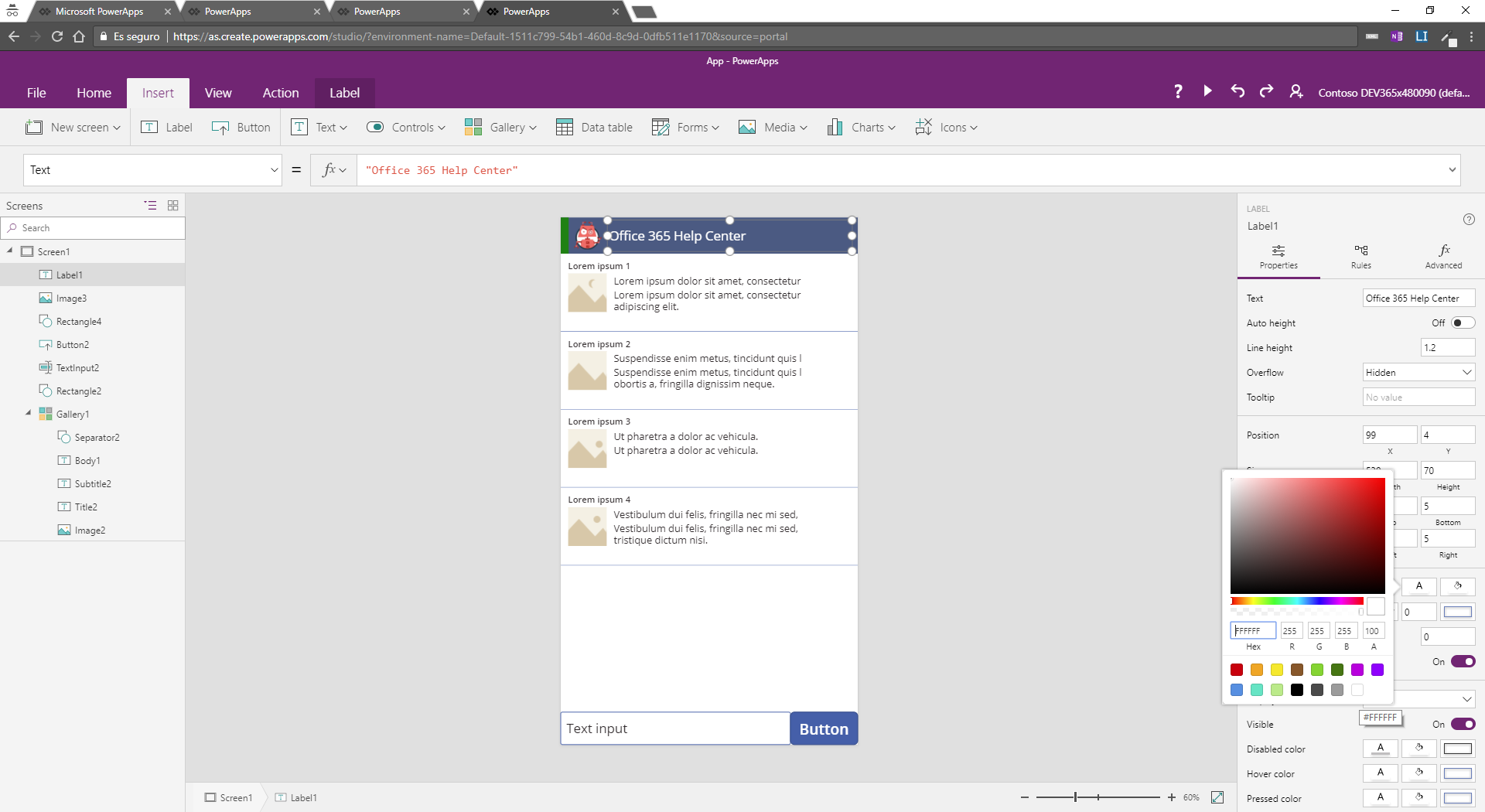
La función ClearCollect, nos creara una colección llamada “ChatColl”, donde almacenaremos nuestras conversaciones y cada vez que se inicie la aplicación esa colección se reseteara.

1. En la propiedad Fill del Rectangulo superior pequeñito que hemos añadido antes, escribir la siguiente función. Esto hará que se visualice en verde cuando estemos conectados a nuestra aplicación y en rojo cuando estemos desconectados.

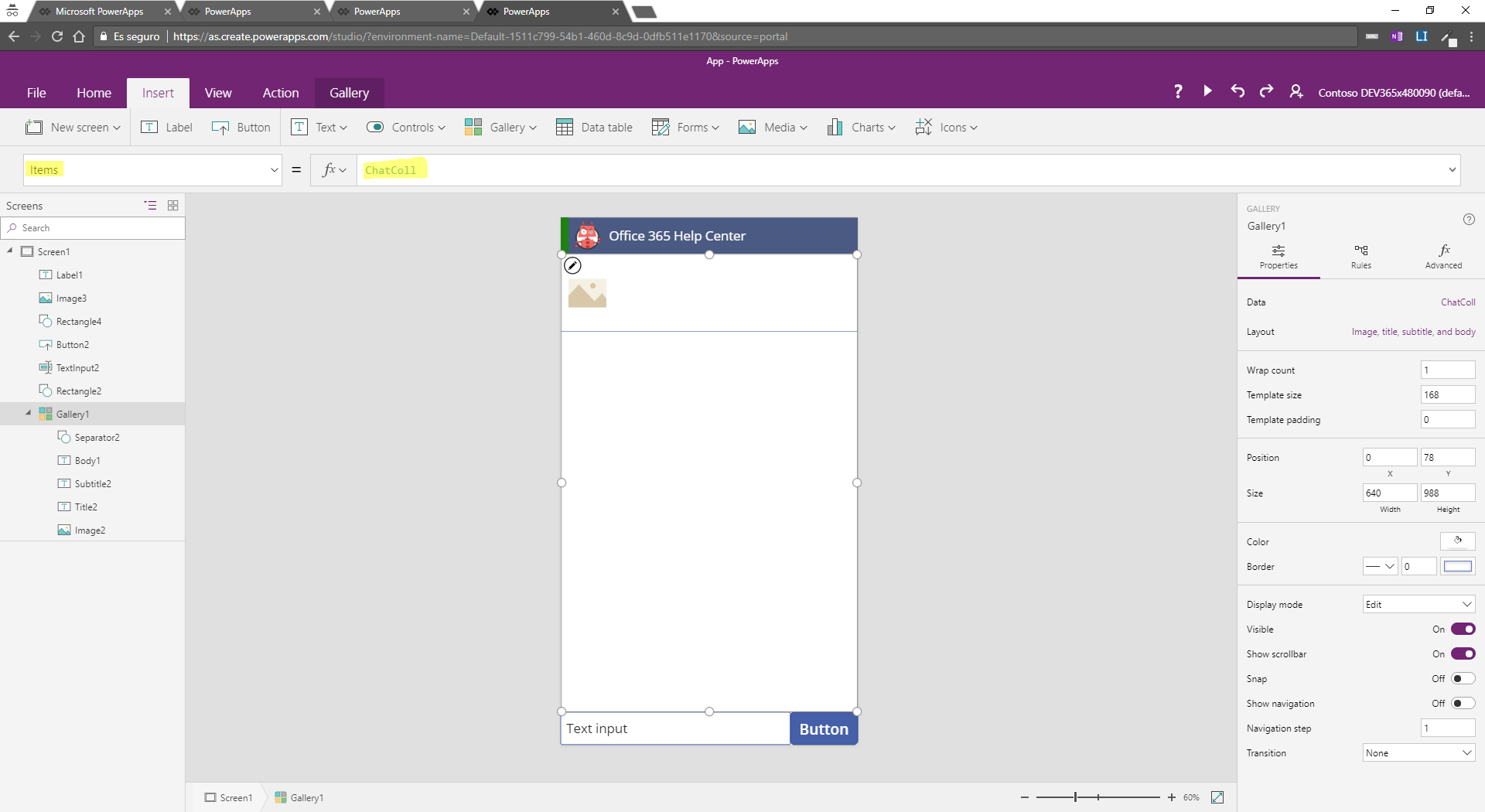
If(Connection.Connected,Green,Red)



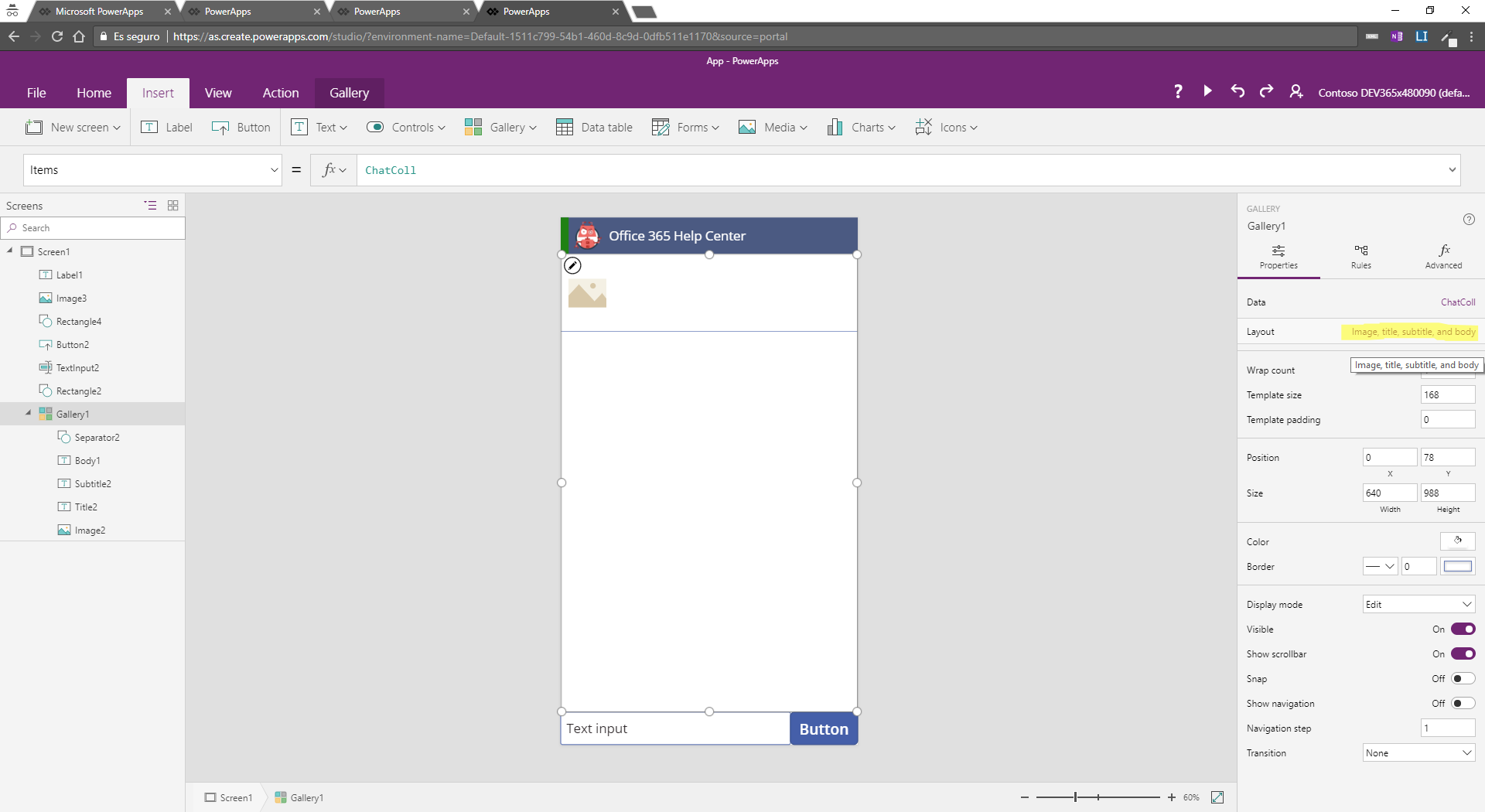
1. Insertamos un label en la cabecera de nuestra aplicación, la centramos y escribimos las letras “Office 365 Help Center”. A continuación cambiamos el color de la fuente a blanco.



1. Es hora de darle un poco de vida a esto. Seleccionamos la galería y en la propiedad Item, escribimos el nombre de la colección que hemos creado antes “ChatColl”



1. A continuación debemos sustituir las propiedades de la galería por los campos de nuestra colección, para ello volvemos a selecciona la opción Layout

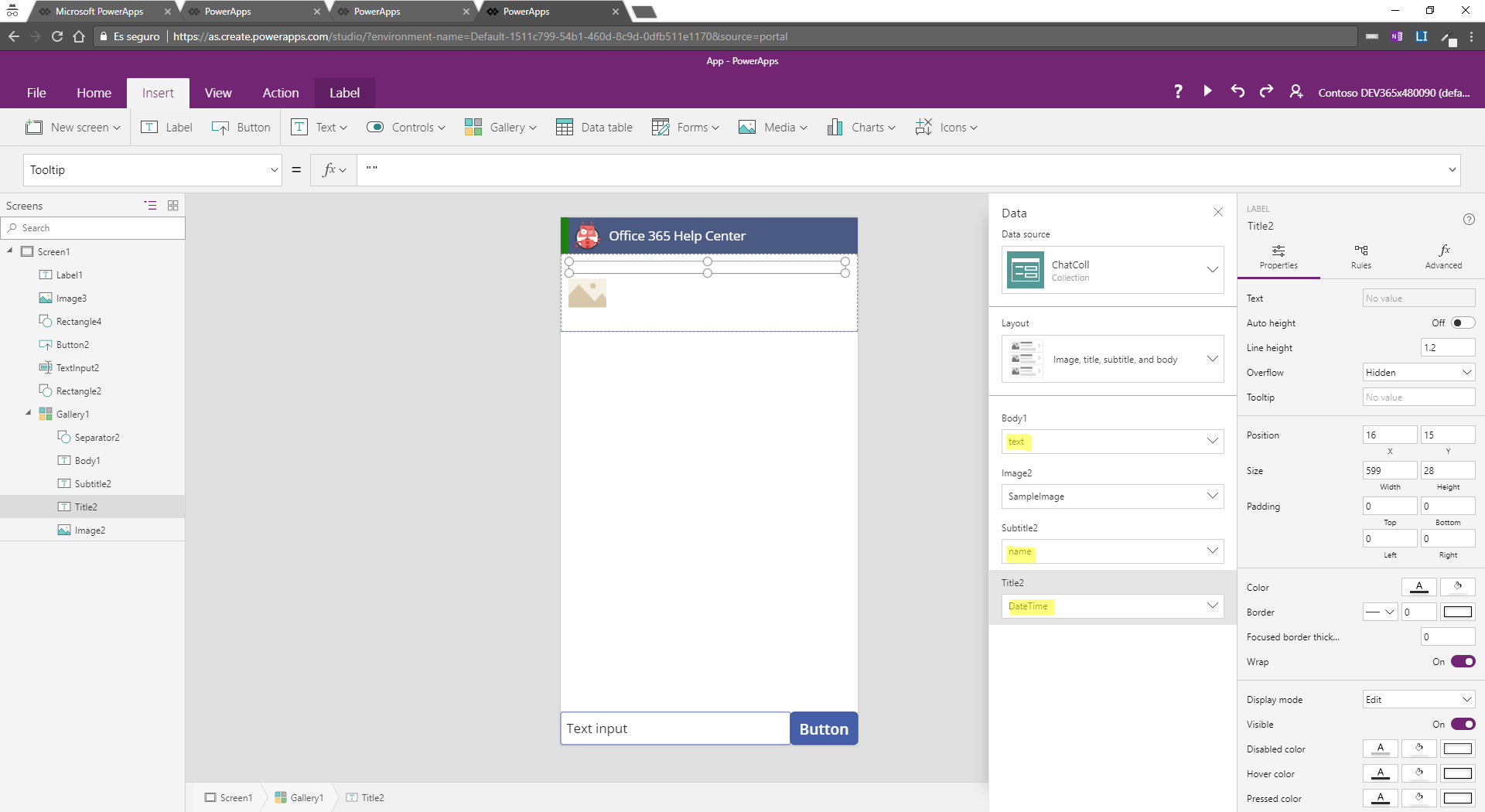


1. Y sustituimos los campos de la siguiente forma, no te preocupes por la imagen, déjalo como esta, ahora vamos a por ella.

Body1: text

Subtitle2: name

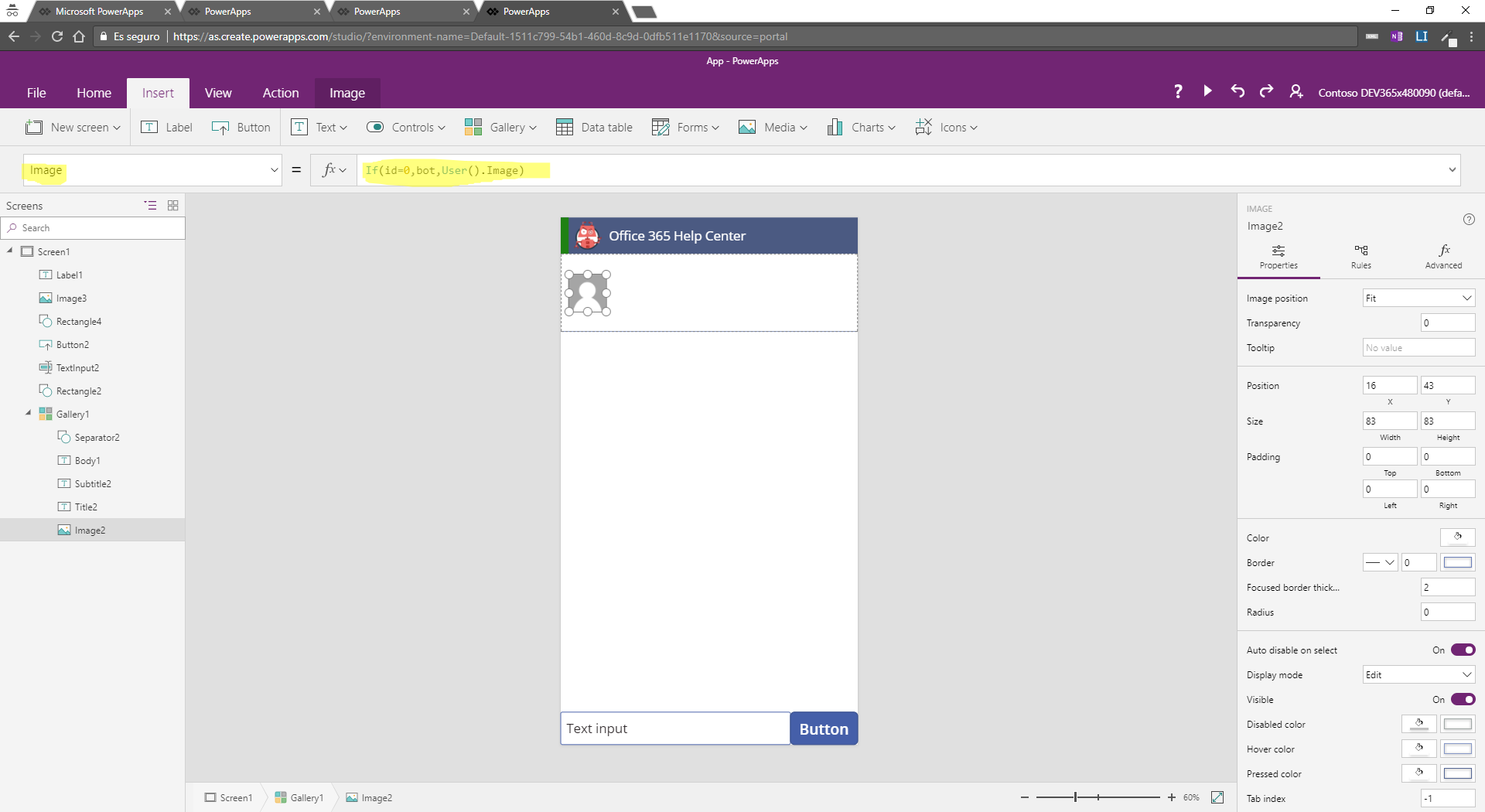
Title2: DateTime



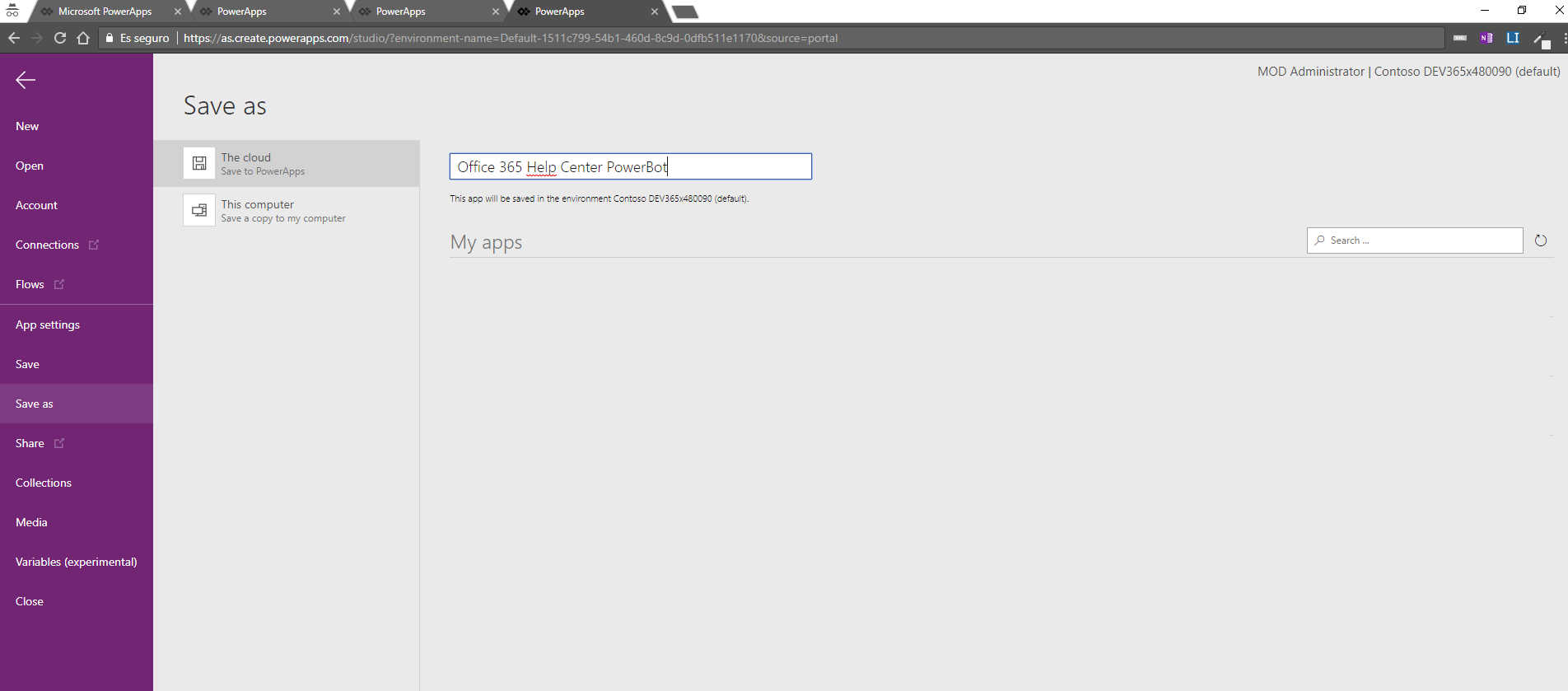
1. Seleccionamos la imagen y en su propiedad “Image” , escribimos lo siguiente:

If(id=0,bot,User().Image)

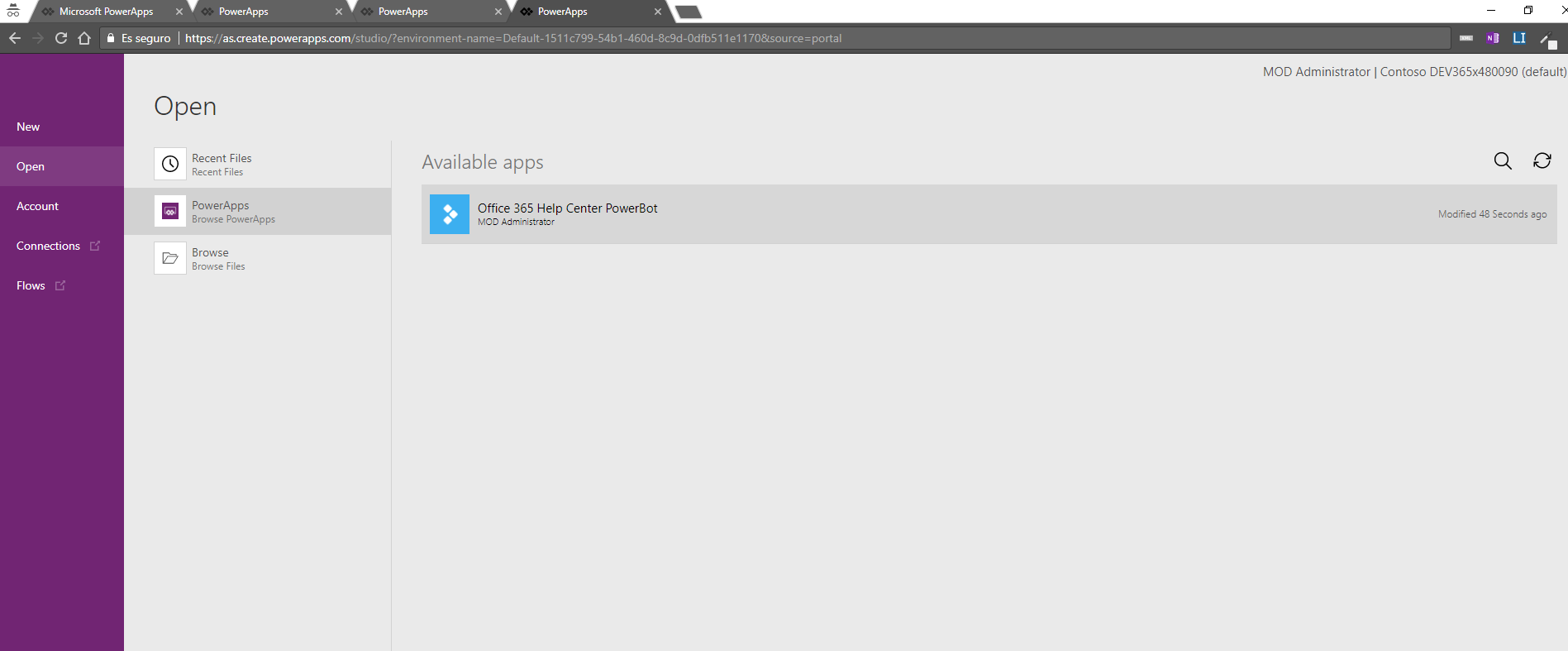
Esto hará que si es el Bot el que escribe mostrara su imagen y si somos nosotros mostrara la nuestra.



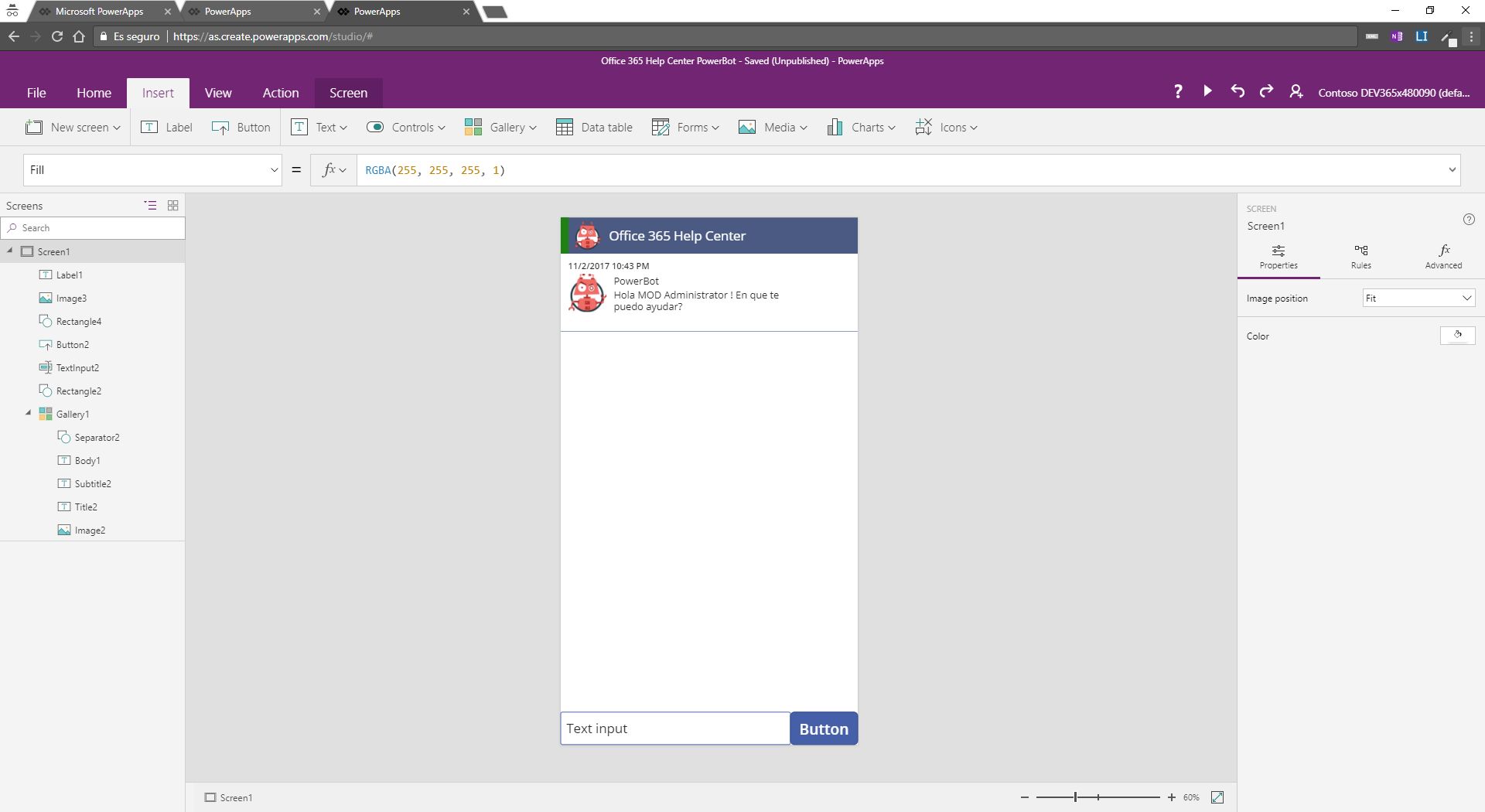
1. Guardamos nuestra solución de Microsoft PowerApps y seleccionamos Close en la ventana actual.



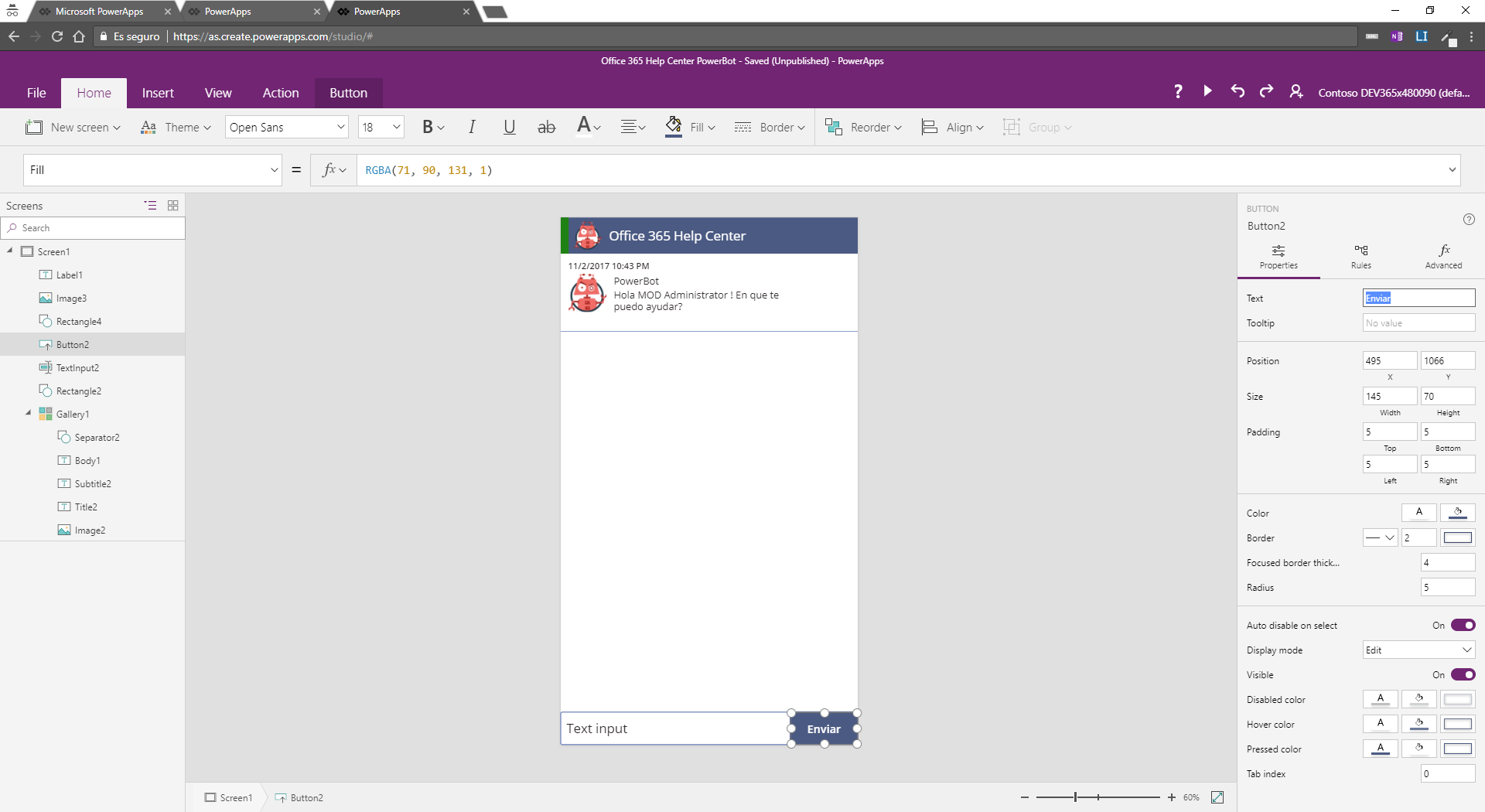
1. A continuación, abrimos de nuevo la aplicación, seleccionando Open -> PowerApps



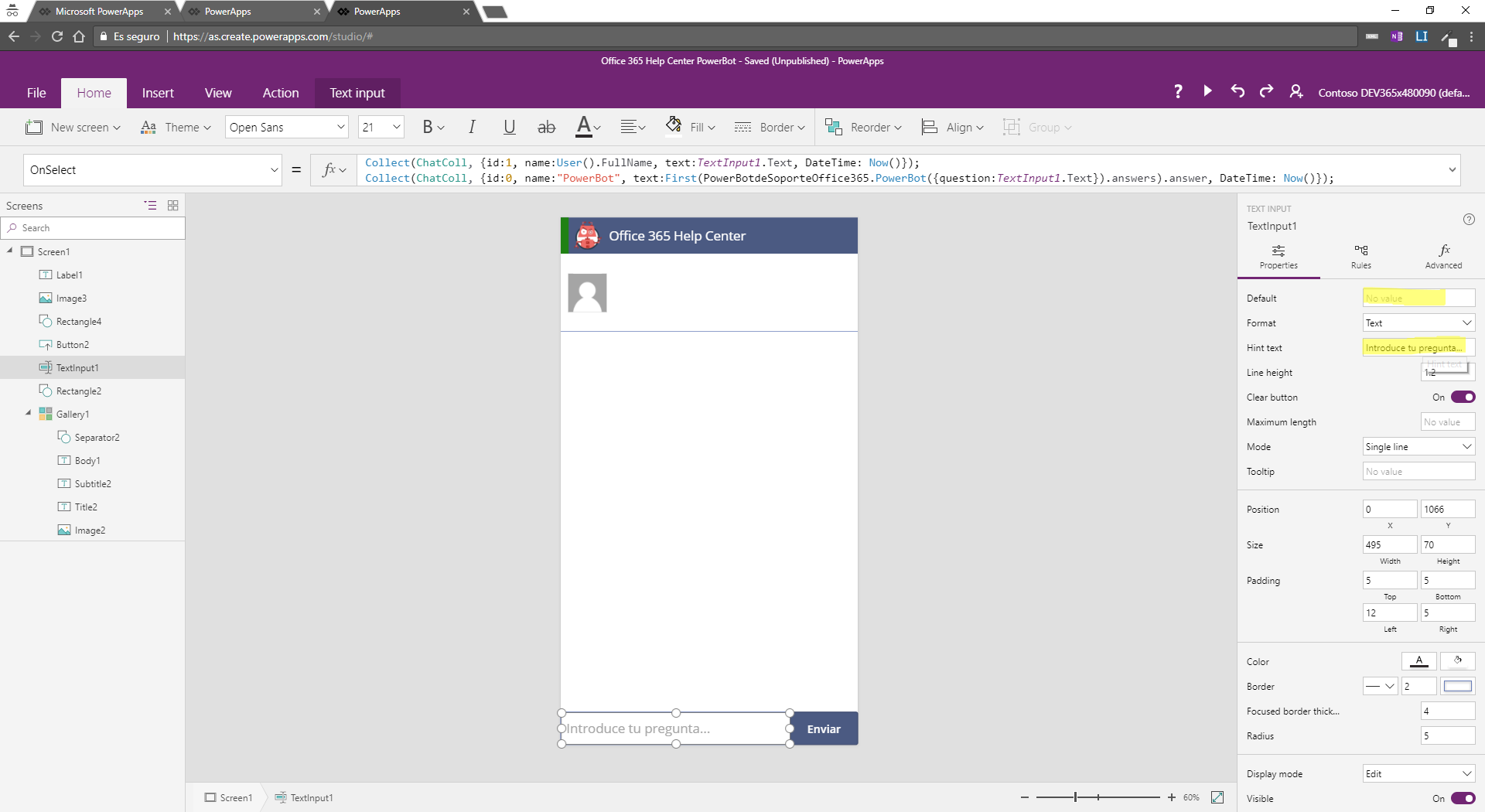
Vemos como ahora ya la aplicación muestra un mensaje, porque la pantalla se ha inicializado mostrando el primer registro de la colección.



1. Seleccionamos el control de botón y modificamos sus propiedades Text: “Enviar”, tamaño de la fuente: 18, propiedad Fill: con el color de la barra superior.



Modificar las propiedades Default y Hint Text del control TextInput.



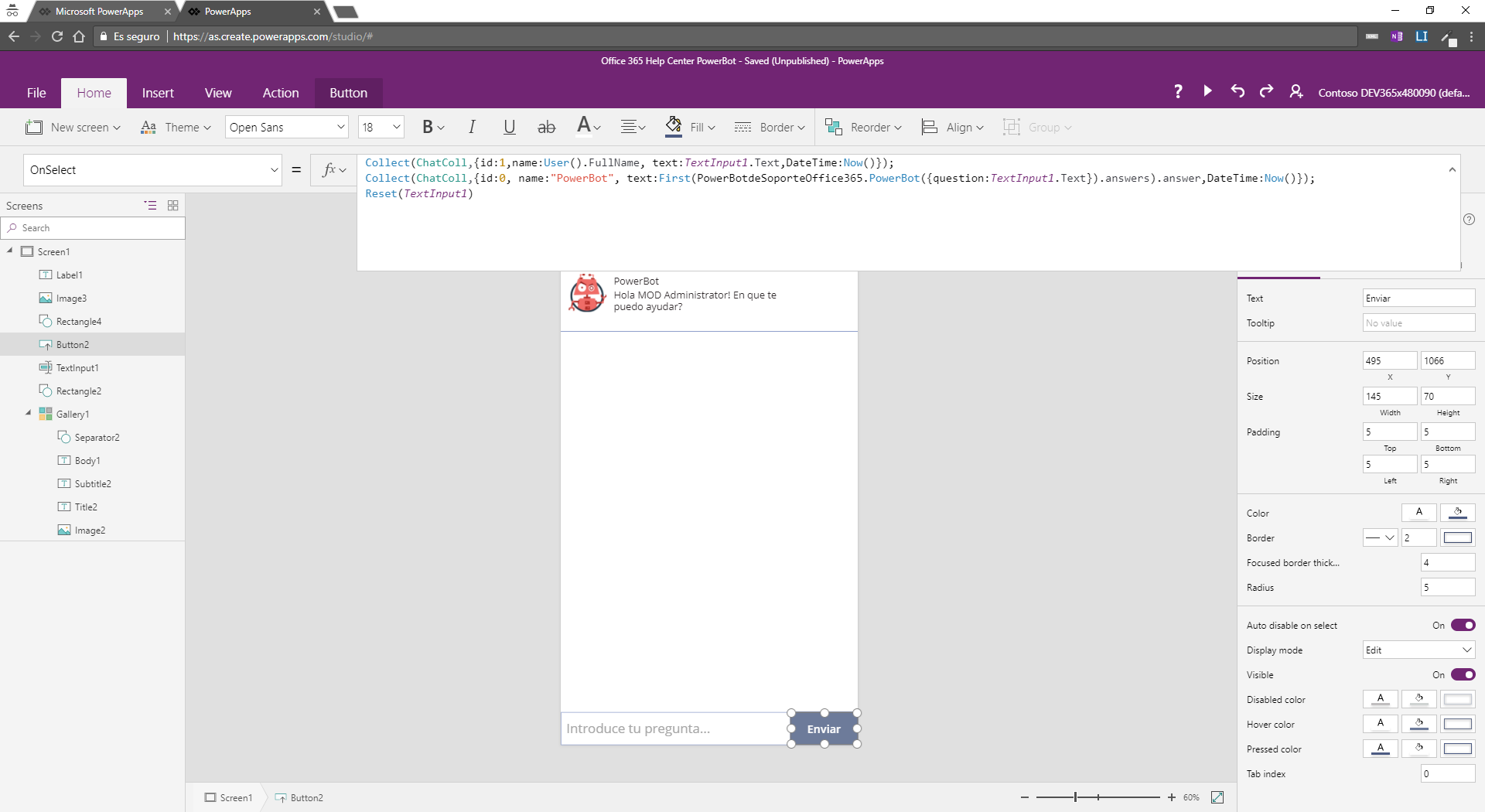
1. A continuación, debemos agregar una función para que nuestro servicio de QnA Maker reciba nuestras peticiones. Para ello, seleccionamos el botón de “Enviar” y en la propiedad OnSelect agregamos el siguiente código de función.

Collect(ChatColl, {id:1, name:User().FullName, text:TextInput1.Text, DateTime: Now()});

Collect(ChatColl, {id:0, name:"PowerBot", text:First(PowerBotdeSoporteOffice365.PowerBot({question:TextInput1.Text}).answers).answer, DateTime: Now()});

Reset(TextInput1)

La primera función Collect guardara nuestra pregunta enviada al servicio QnA Maker, la segunda función Collect guardara la respuesta que nos devuelva el servicio QnA Maker. Finalmente, la función Reset, limpiara el campo donde escribimos cada vez que pulsamos el botón enviar.



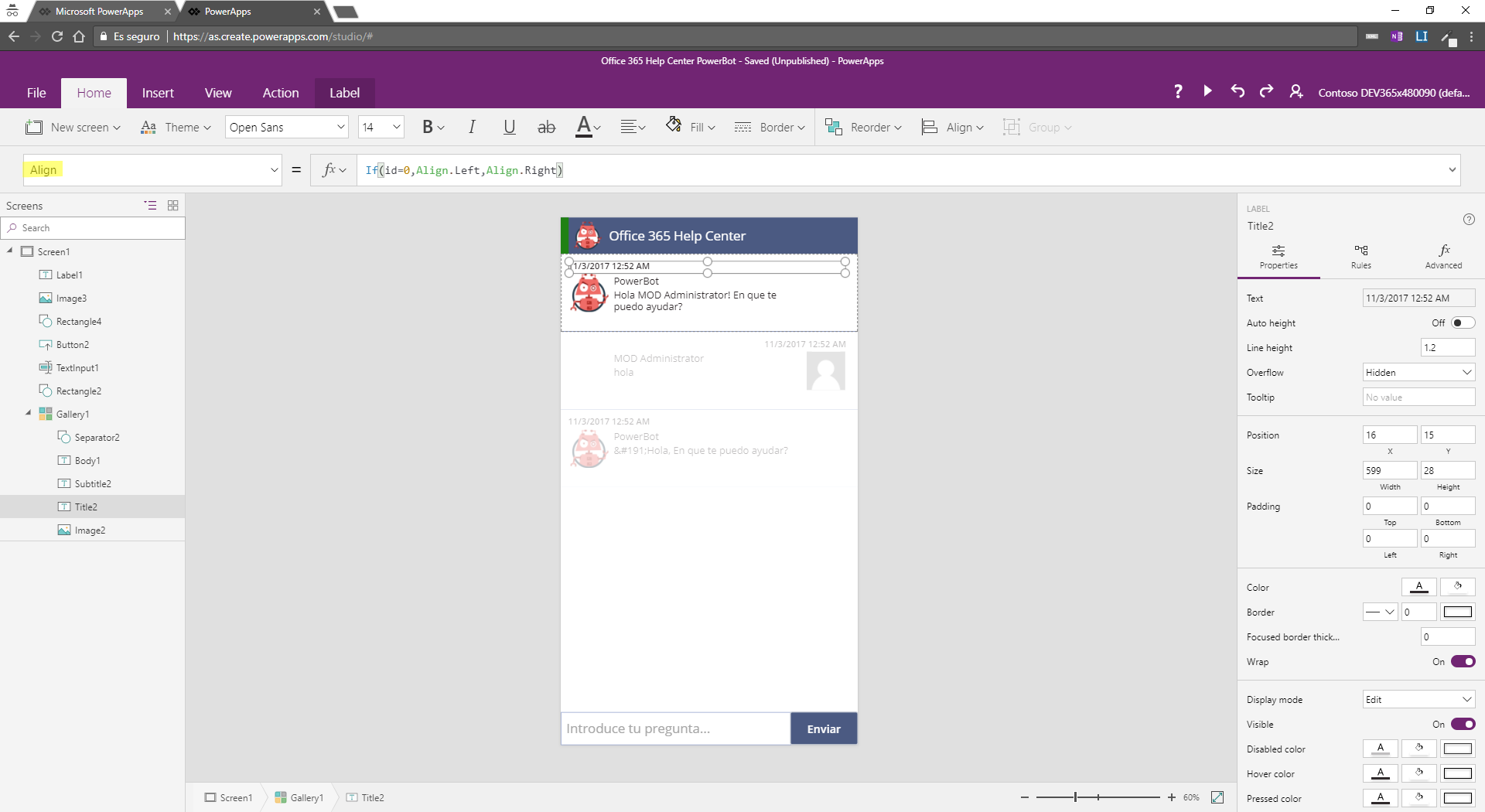
1. Ejecutamos la aplicación pulsando F5, y escribimos “Hola”. El Bot nos contestara. Vamos a hacer algunos ajustes mas. Vamos a seleccionar la imagen del bot y en la propiedad X, agregamos el siguiente código.

If(id=0,20,530)

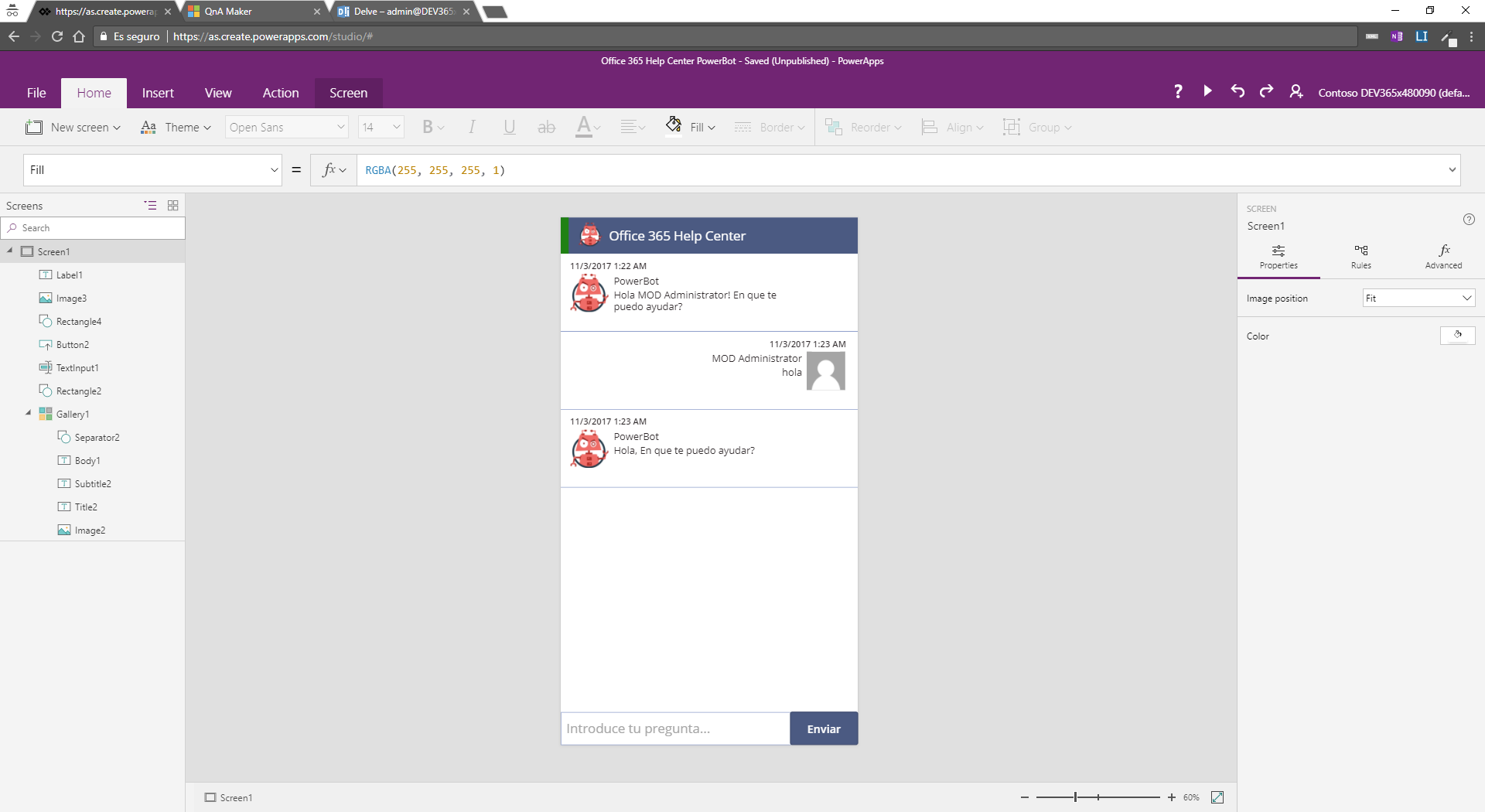
Este código, modificara la alineación de la imagen, si es el bot el que está escribiendo o somos nosotros. Lo mismo debemos hacer con el resto de controles.

Selecciona el control Title1 y en su propiedad “Align” agrega el siguiente código. Esto mostrara el texto alineado a la derecha si somos nosotros los que escribimos y lo alineara a la izquiera si es el bot el que escribe.

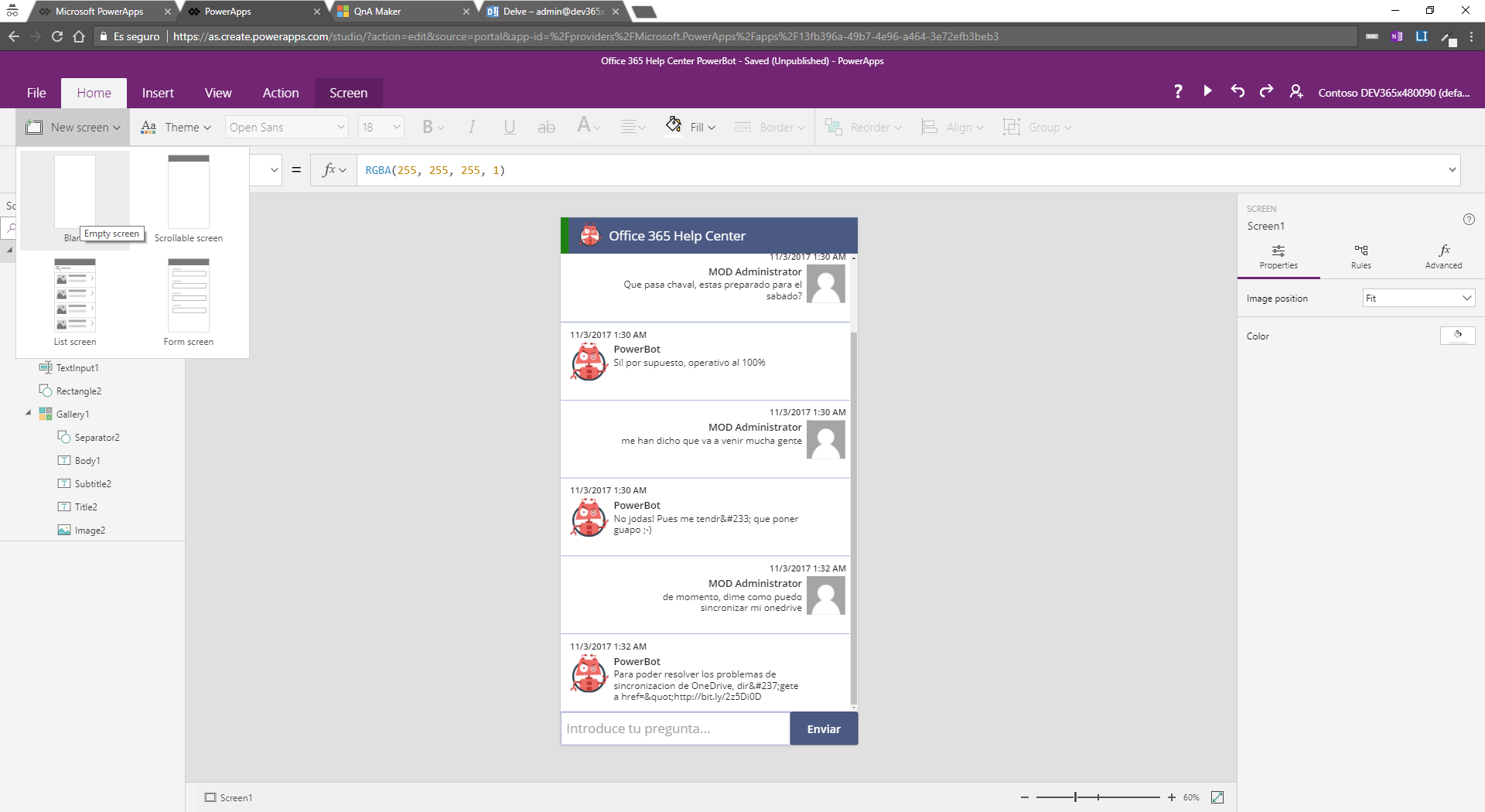
If(id=0,Align.Left,Align.Right)



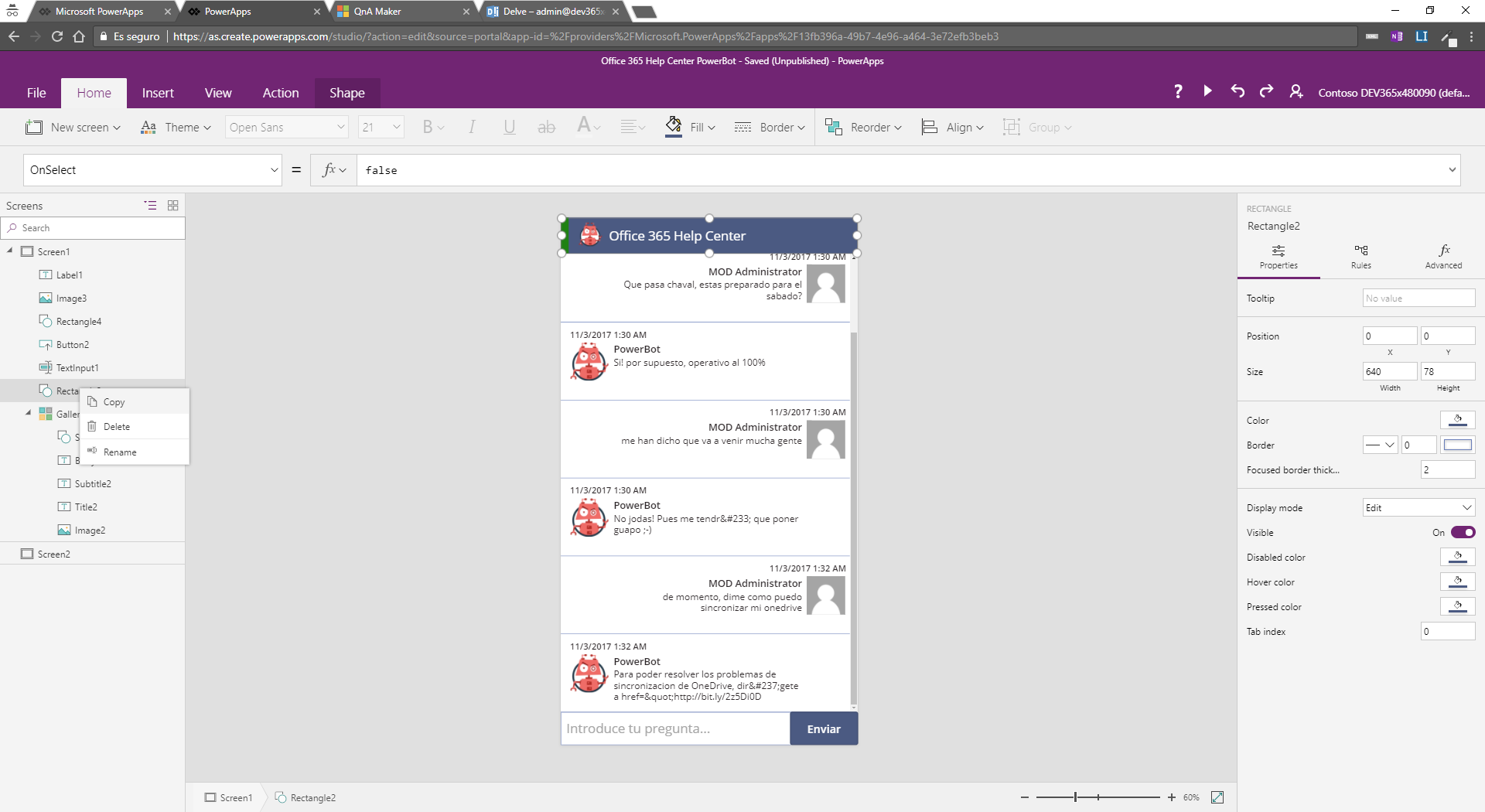
1. Lo mismo haremos con los controles Subtitle1 y Body1. La apariencia final de nuestra aplicación debería ser la siguiente:



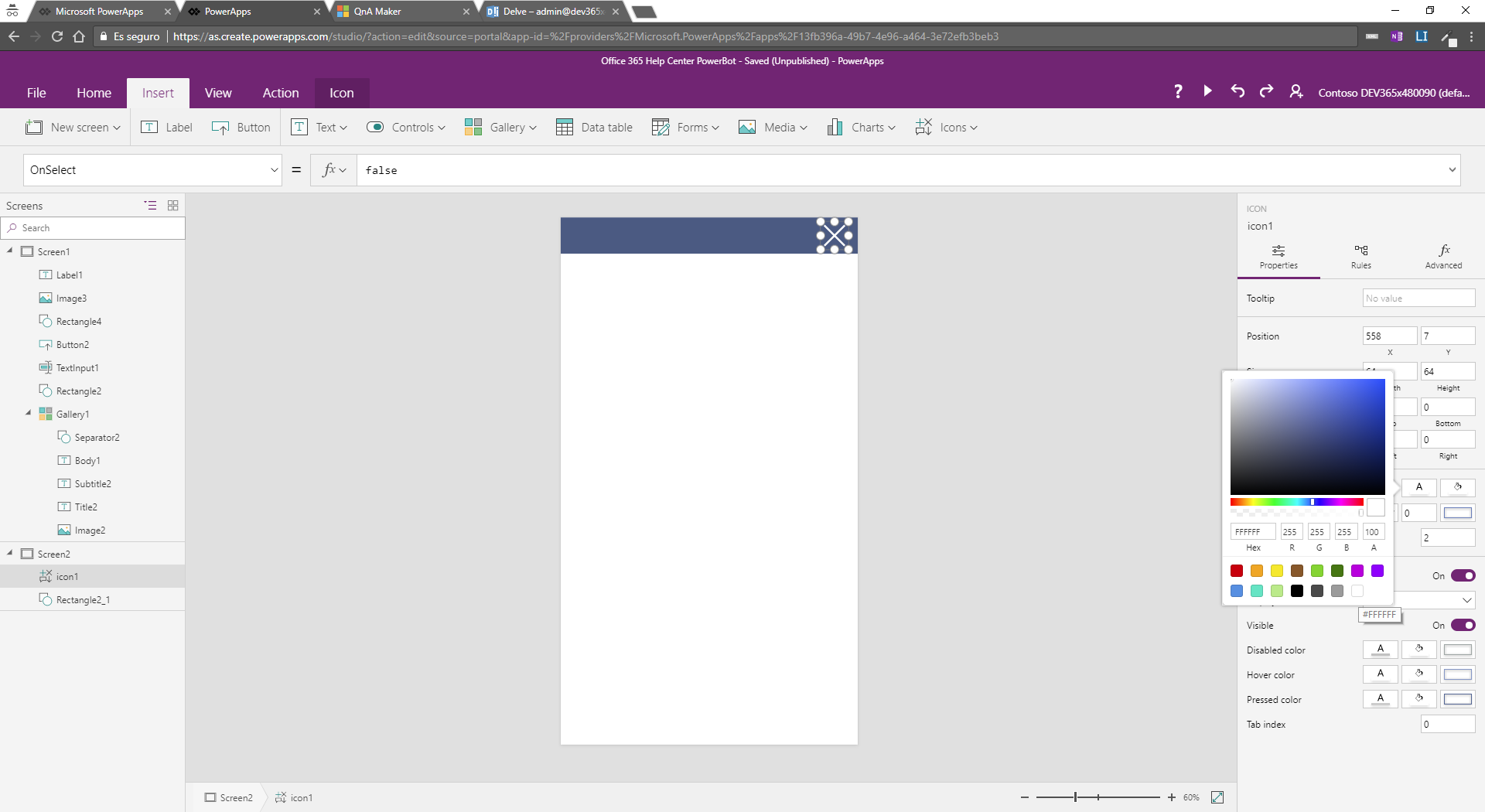
1. A continuación, insertaremos una nueva pantalla en blanco, para ello seleccionamos en la sección Home del menú, New Screen



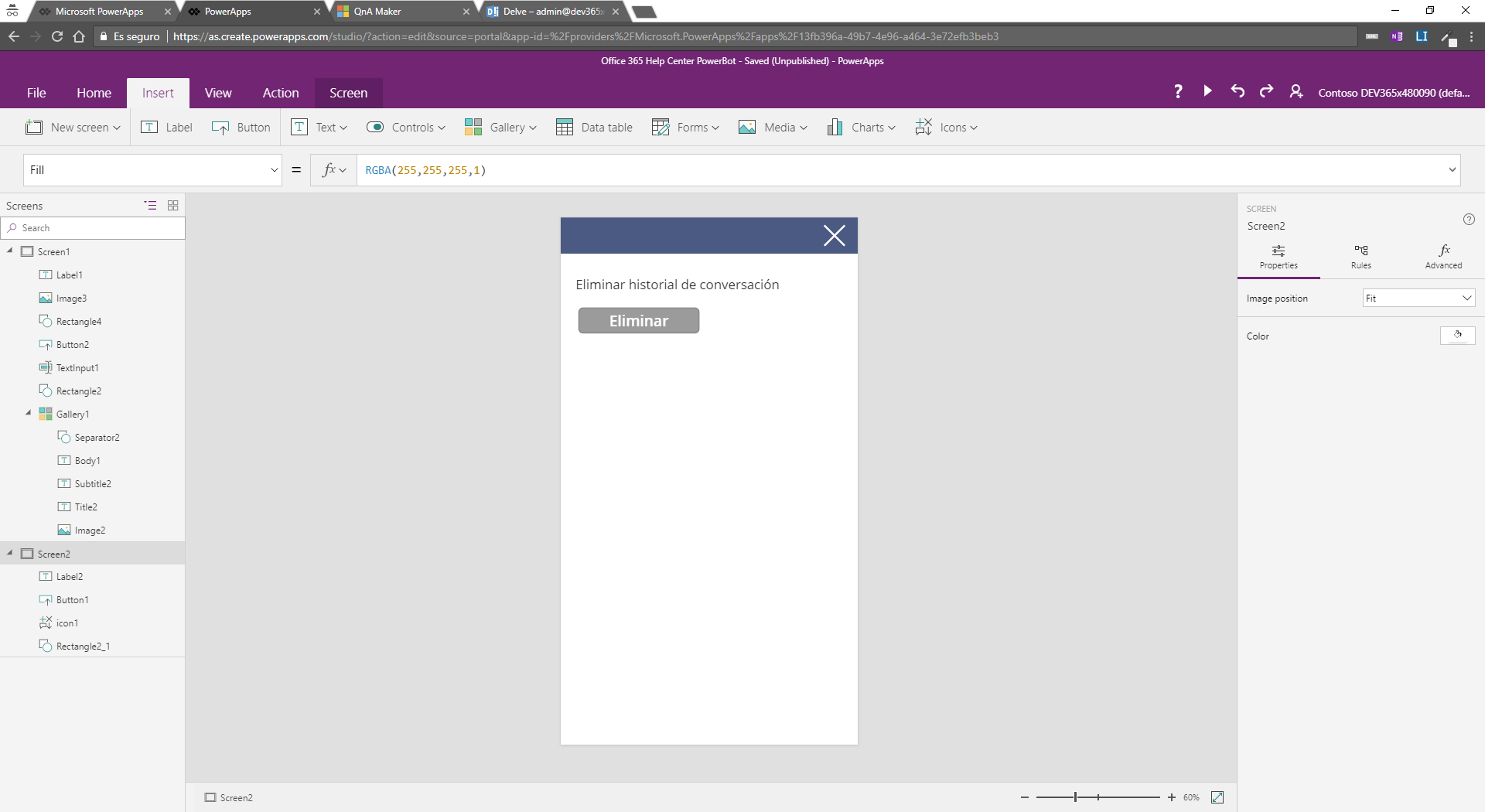
Copiaremos el control Rectangle1 de Screen1 y lo pegaremos en Screen2.



Lo colocaremos en la parte superior de la pantalla como en Screen1 e insertaremos un icono de una X, cambiándole el color a blanco



1. Agregaremos dos controles, un label y un Button. Les daremos formato para que adquieran este aspecto y cambiaremos los textos para que se muestren como los de la imagen



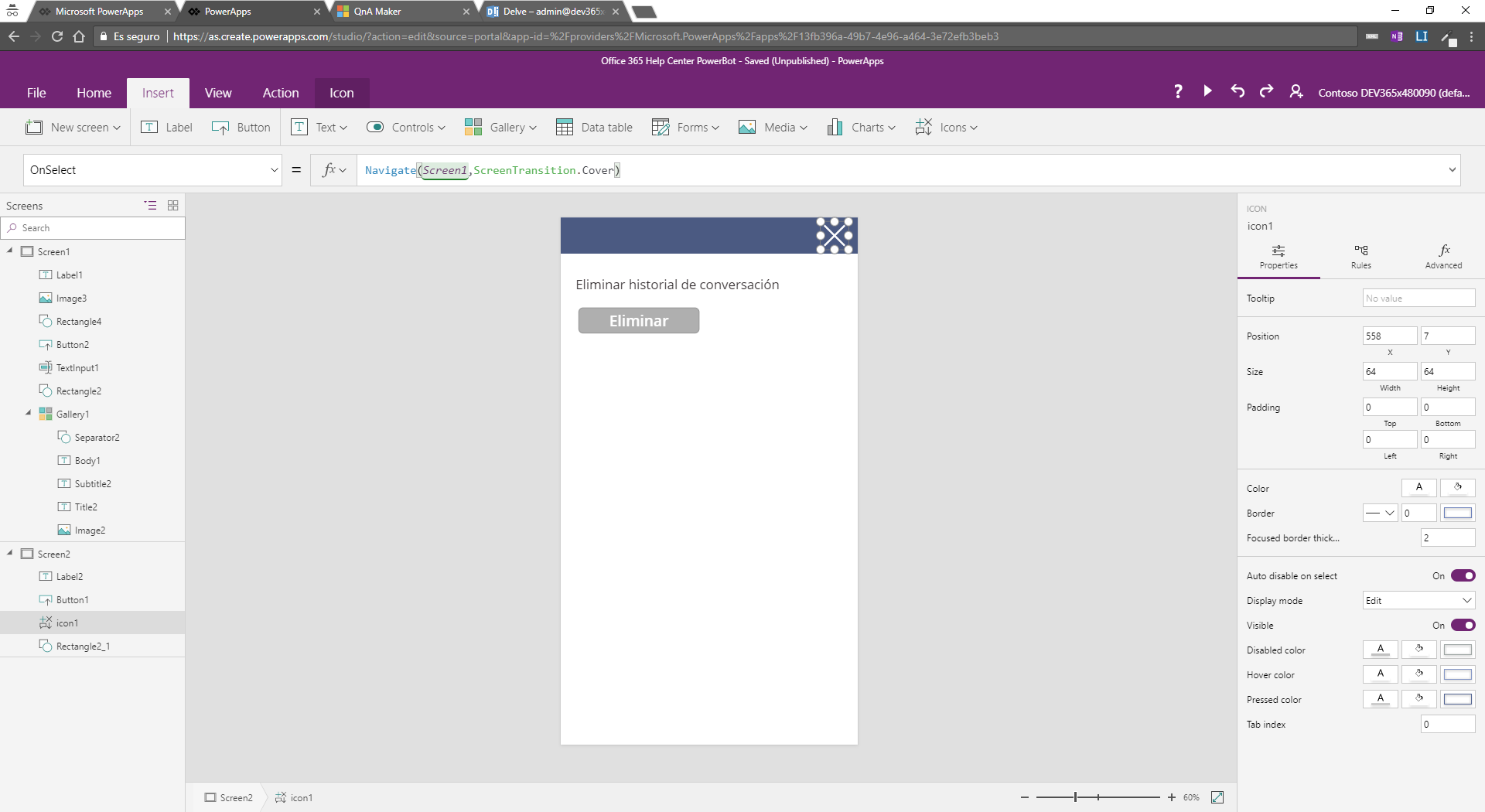
En la propiedad OnSelect del botón de Eliminar escribiremos el siguiente código

Clear(ChatColl)

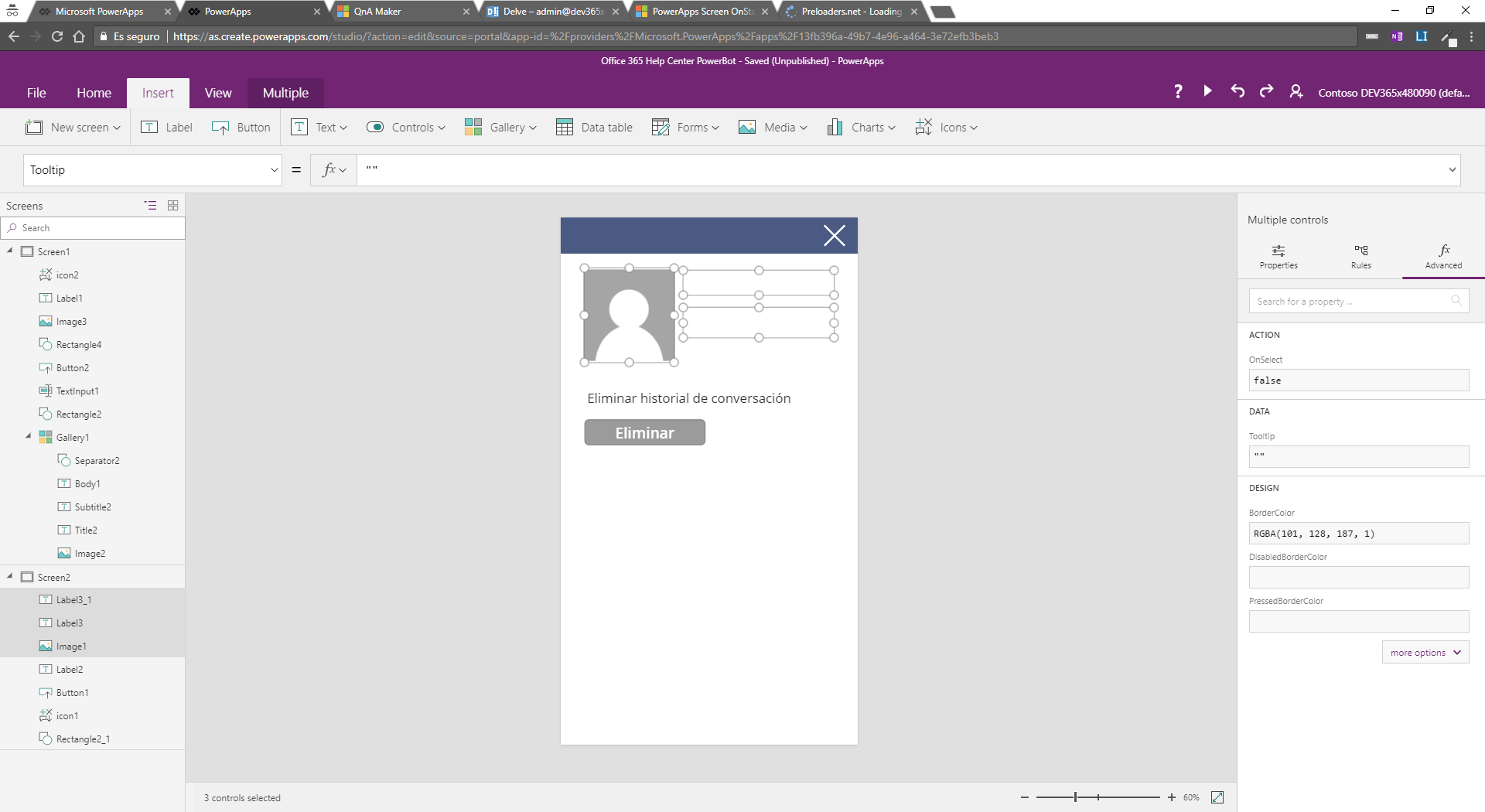
Esto eliminara el historial de conversaciones almacenado en la colección de PowerApps.

Por último, en la propiedad OnSelect del icono X que hemos agregado, escribir el siguiente código

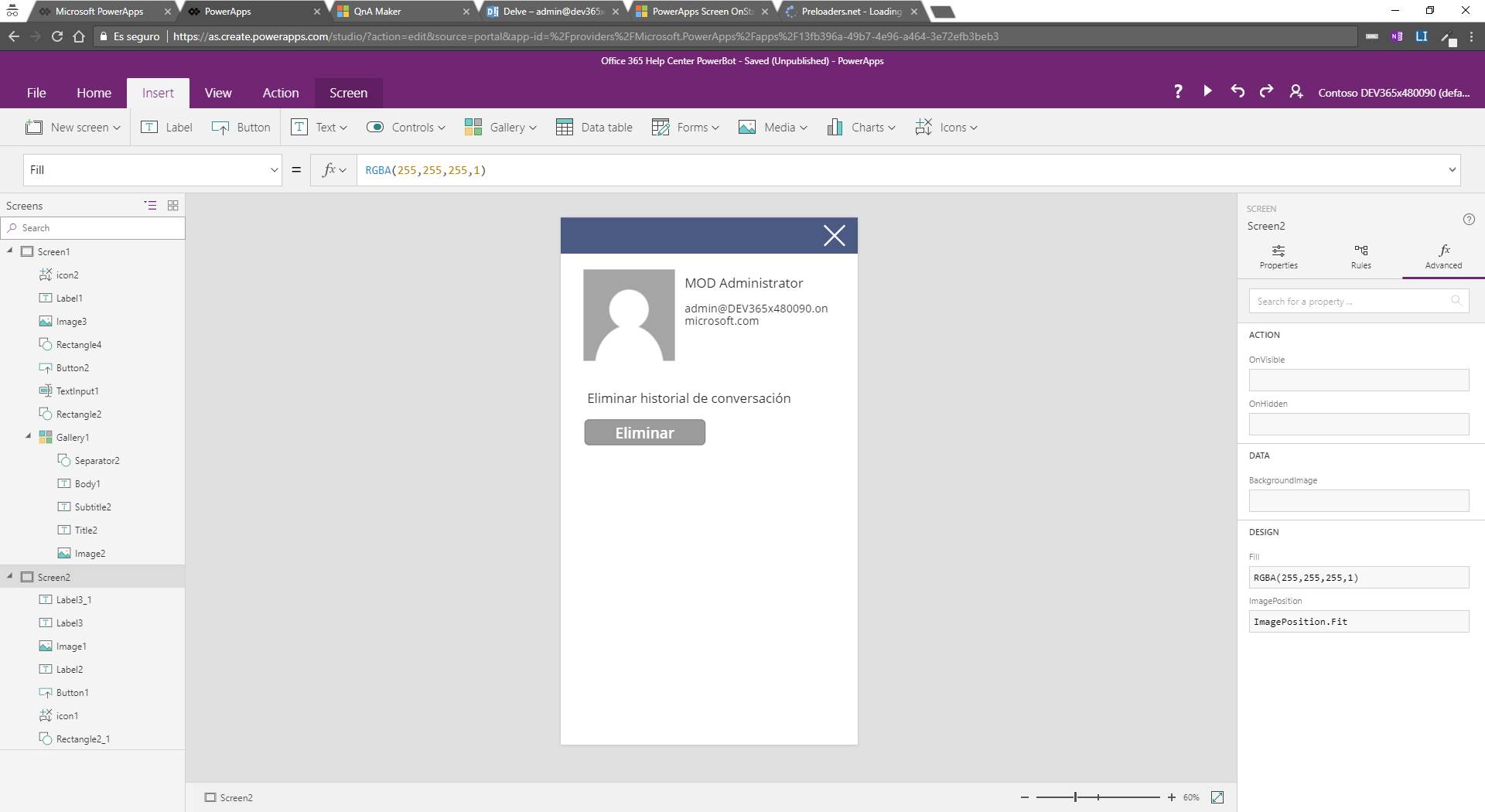
Navigate(Screen1,ScreenTransition.Cover)



Encima de el texto de Eliminar historia… agregaremos estos controles, una imagen y dos label



1. En la propiedad Image de la imagen, pondremos “User().Image” en la propiedad text de los dos label pondremos “User().FullName” y “User().Email”

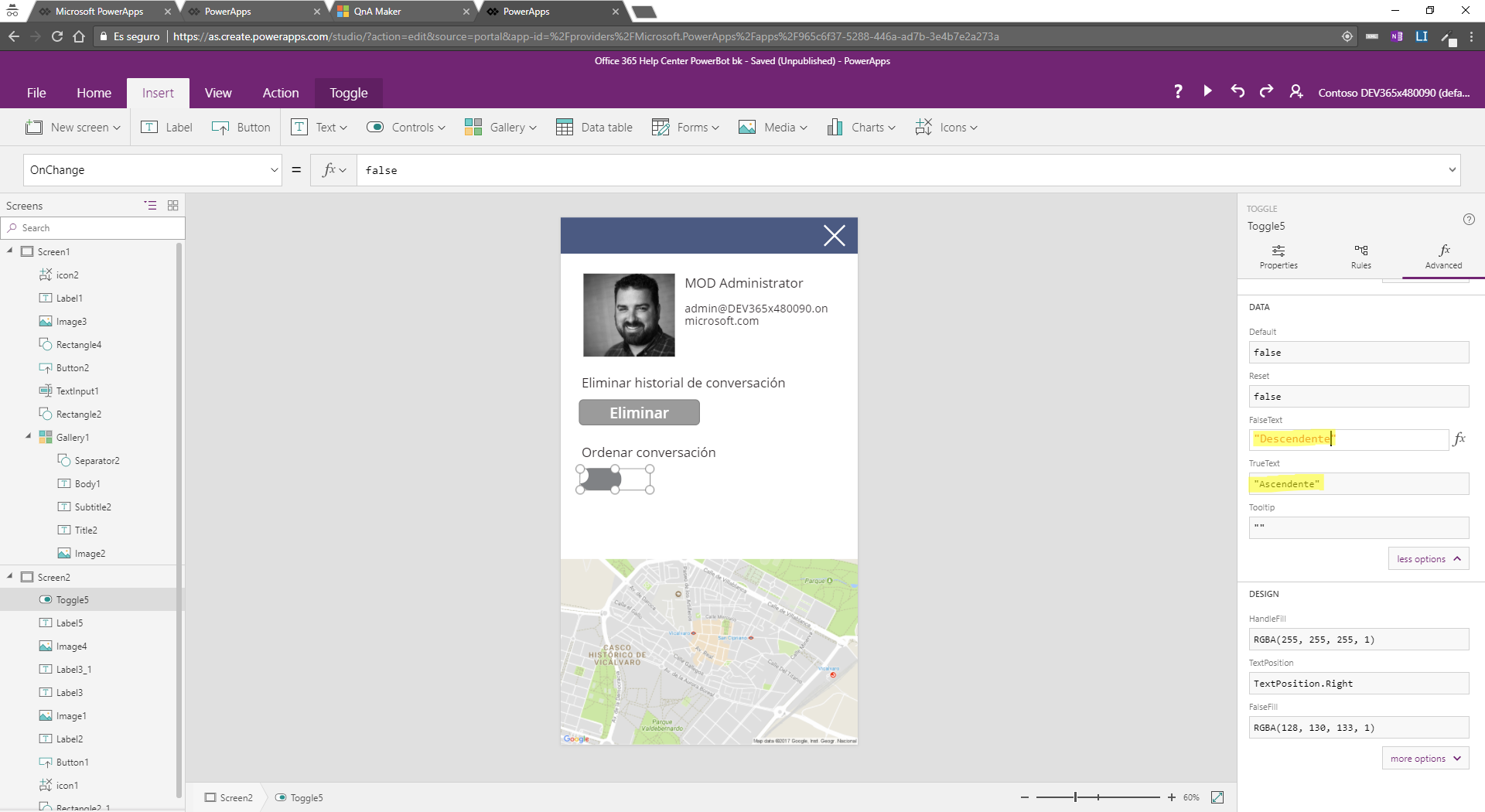


1. Vamos a agregar una imagen en la parte inferior de la pantalla y en la propiedad image, agregaremos el siguiente código.

"https://maps.googleapis.com/maps/api/staticmap?center="& Location.Latitude & ","& Location.Longitude & "&zoom=15&size=640x400&path=weight:3%7Ccolor:blue%7Cenc:{coaHnetiVjM??\_SkM??~R&key=AIzaSyC2bb0CcBU\_X6zmarwvGhzdy63CtAntwks"

Esto nos mostrara un mapa con la localización actual que tiene nuestro dispositivo. Es un ejemplo de como podemos utilizar la api de Google maps para integrarla en nuestros desarrollo de PowerApps.

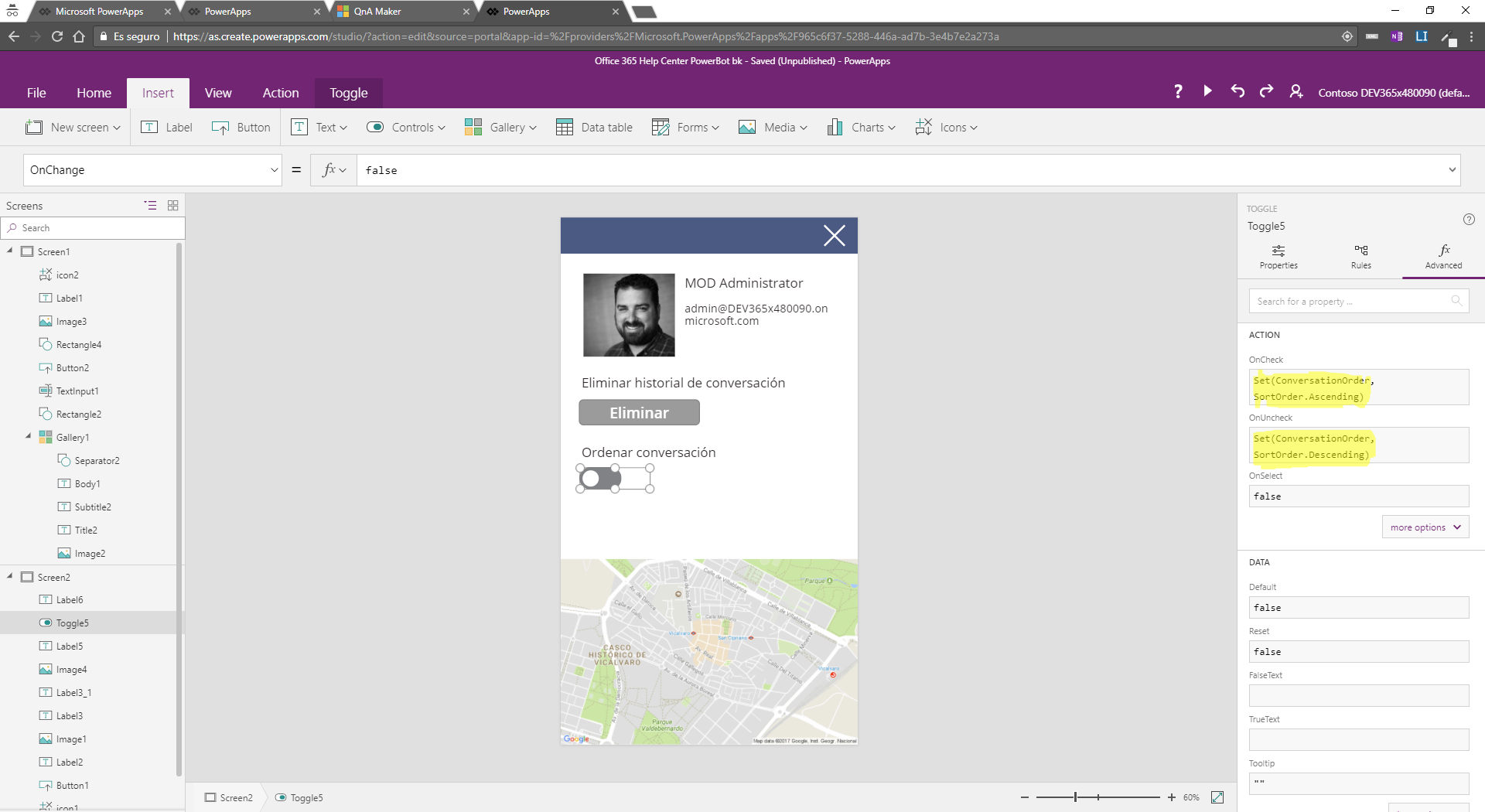
1. Vamos a agregar un control label y renombraremos la propiedad text por el texto “Ordenar conversación”. A continuación agregaremos un control Toogle y eliminaremos los valores de las propiedades FalseText y TrueText.



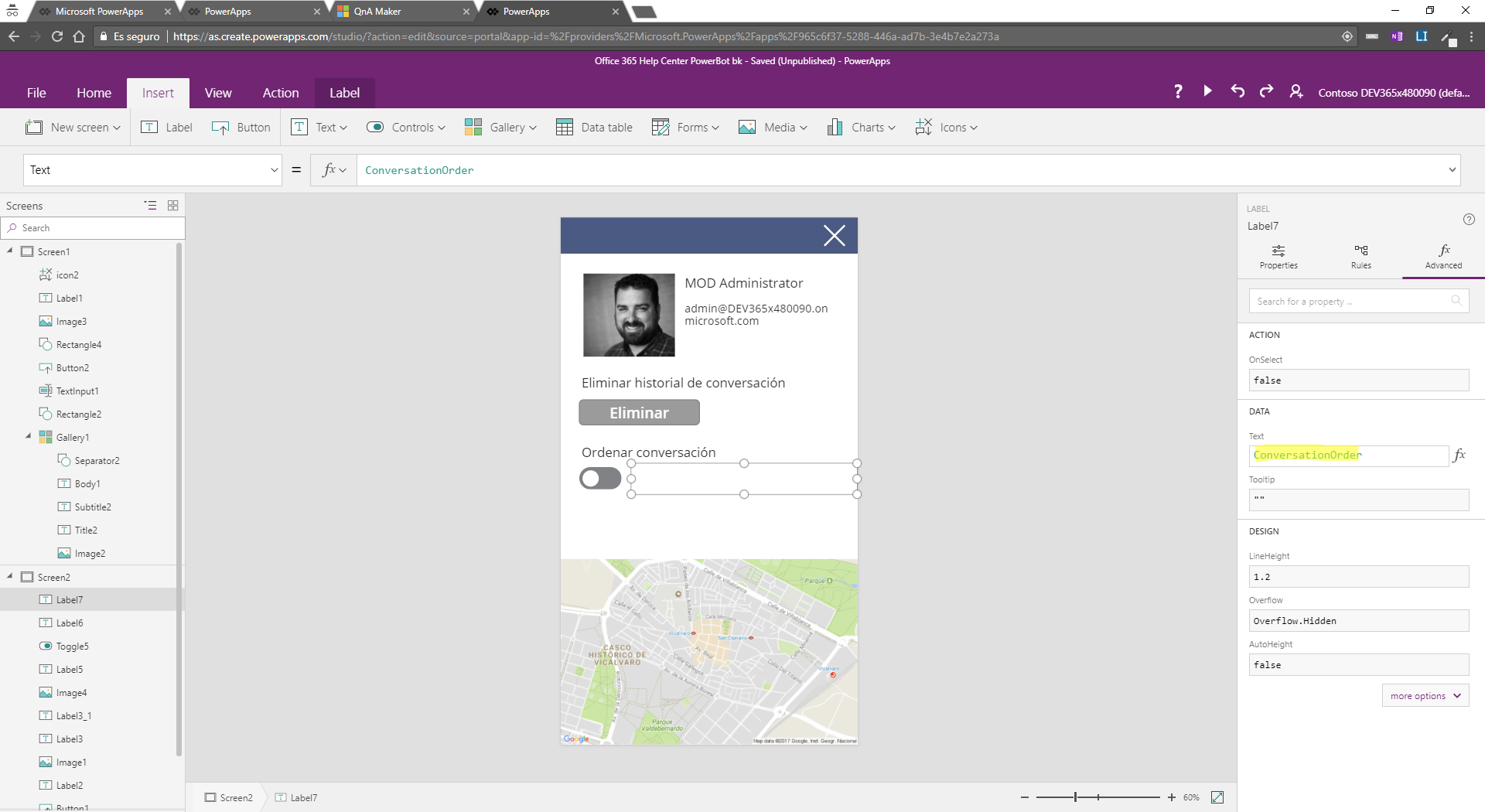
1. En las propiedades Oncheck y Uncheck agregamos las siguientes funciones:

OnCheck: Set(ConversationOrder, SortOrder.Ascending)

UnCheck: Set(ConversationOrder, SortOrder.Descending)

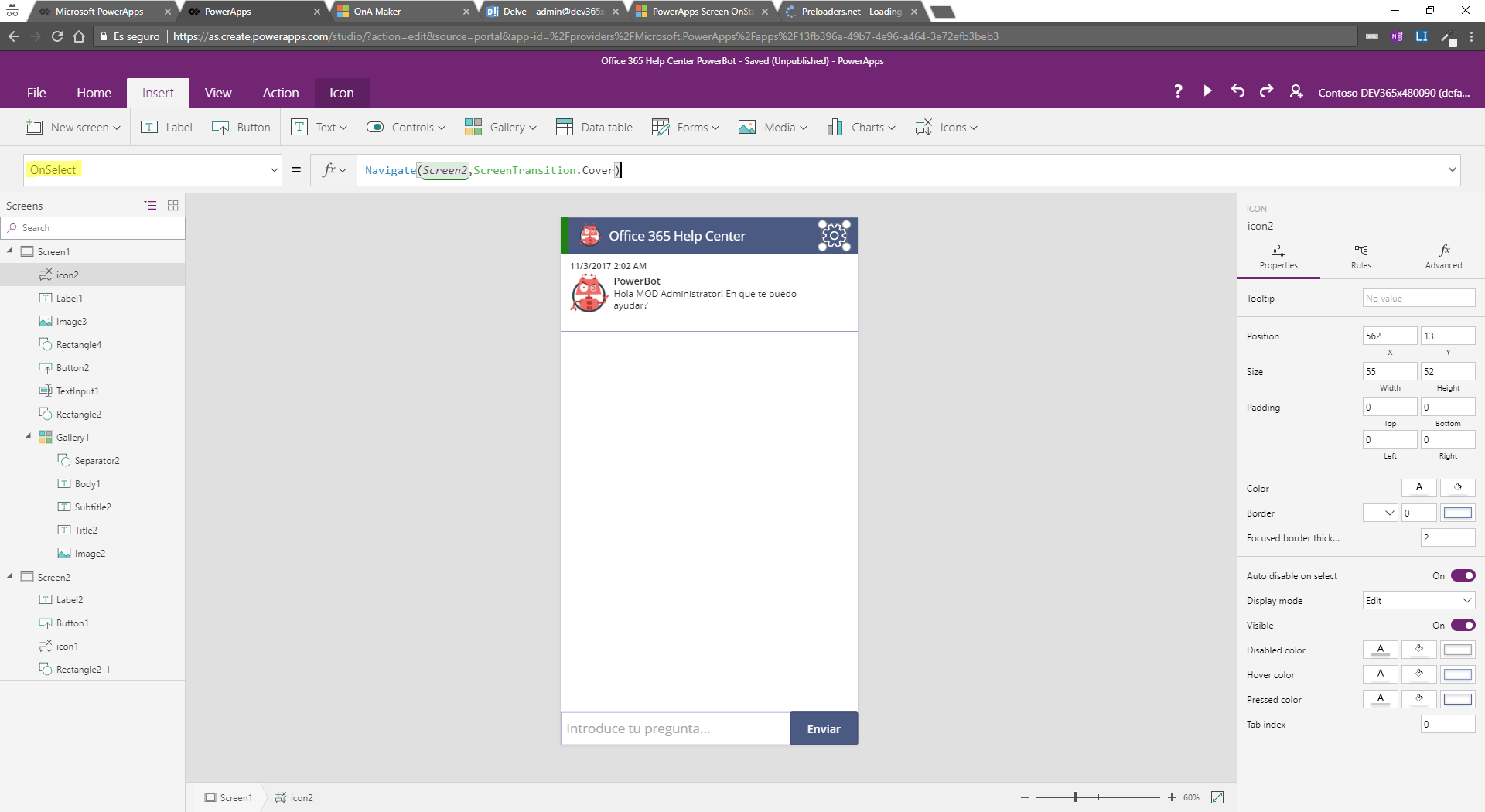


1. Agregamos un label y lo colocamos al lado del botón Toggle. En la propiedad text del elemento añadimos el nombre de la variable “ConversationOrder”



1. Volvemos a la pantalla anterior e insertamos un icono como el de la imagen, lo colocamos del mismo modo que la imagen de abajo y agregamos el siguiente código a su propiedad OnSelect.

Navigate(Screen2,ScreenTransition.Cover)



1. Finalmente, sustituimos el texto en la propiedad ítems de la galería por el siguiente código:

Sort(ChatColl,DateTime, ConversationOrder)

Esto hará que podamos elegir que tipo de ordenación queremos que tenga la conversación de nuestro PowerBot.