

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

“Presentación de datos OData  
en Power BI” Universidad de Almería

Curso 2016/2017

**Alumno/a:**

Antonio López Hernández

**Director/es:**

Manuel Torres Gil





# Índice del documento

Capítulo 0 .....	5
Capítulo 1 .....	6
1.1 ¿Qué es Bussiness Intelligence? .....	6
1.2 Utilidad de Bussiness Intelligence .....	6
1.3 Hacer uso de Bussiness Intelligence .....	7
1.4 Distintas herramientas para hacer uso de Bussiness Intelligence .....	8
Resumen .....	9
Capítulo 2 .....	10
2.1 ¿Qué es PowerBI? .....	10
2.2 Funcionamiento de Power BI .....	11
2.3 Productos de Power BI .....	11
2.3.1 Power BI Cloud. ....	11
2.3.2 Power BI Desktop. ....	13
2.3.3 Power BI Mobile. ....	15
2.4 Fuentes de Power BI .....	16
Resumen .....	17
Capítulo 3 .....	18
3.1 ¿Qué es OData? .....	18
3.2 Cómo surge OData .....	19
Resumen .....	22
Capítulo 4 .....	23
4.1 Fuentes externas OData .....	23
4.2 Consumo fuentes OData externas .....	23
Resumen .....	30
Capítulo 5 .....	31
5.1 Introducción al ejemplo. ....	31
5.3 Creación del proyecto en Visual Studio .....	37
5.4 Conexión del proyecto con la base de datos. ....	38
4.5 Instalación del framework Restier desde NuGet .....	42
5.6 Modificaciones pertinentes y ejecución del servicio. ....	44
5.7 Consumo de la fuente mediante Power BI .....	47
5.8 Conclusiones finales. ....	49
Resumen .....	59

<b>Capítulo 6</b> .....	60
<b>6.1 Conclusión sobre el proyecto.</b> .....	60
<b>6.2 Cronograma.</b> .....	61
<b>6.3 Trabajo futuro.</b> .....	62
<b>6.3.1 Desarrollo de framework propio de OData.</b> .....	63
<b>6.3.2 Creación de un proyecto real para Power BI.</b> .....	63
<b>Bibliografía</b> .....	64

# Capítulo 0

## Introducción

---

En este proyecto se pretende demostrar el uso y la gran utilidad de Power BI para la inteligencia de negocio. Power BI, como ya veremos a lo largo de este documento, admite distintas fuentes de datos para poder conseguir información sobre nuestra entidad. Este proyecto tratará de una empresa ficticia a la que someteremos al paso por la herramienta Power BI. Aunque esta herramienta admite distintas fuentes de datos, como puede ser un archivo Access o un archivo Excel, para este proyecto hemos querido centrarnos en la fuente de datos OData. Esto es debido a que, al igual que Power BI, está desarrollada por Microsoft. Aunque OData ofrece una gran fuente, veremos que no existen muchas más a disposición del usuario. Además, crear nuestra fuente de datos propia ofrece una gran funcionalidad y facilita el traspaso de datos.

La idea del proyecto es que cualquier persona sea capaz de crear su fuente de datos OData. Para este caso vamos contar con una base de datos y la vamos a importar a SQL Server. Desde SQL Server crearemos un programa en Visual Studio que sea capaz de crear nuestro propio servidor API REST. Después atacaremos a nuestro servidor con la herramienta Power BI y conseguiremos rescatar la información que deseemos de nuestra base de datos para enviarla a Power BI. De este modo recabaremos suficiente información para crear una serie de tablas e informes que nos servirán para tomar decisiones para la empresa.

En este proyecto se realizará una introducción a los conceptos básicos necesarios en los que se basan los ejemplos desarrollados en él. Esto lo conseguiremos gracias a una serie de conceptos que se relatan en los primeros capítulos del documento con una guía y unos ejemplos para terminar de comprenderlo.

Se presentan dos proyectos en los cuales hemos realizado una serie de ejemplos de cómo afectarían para el desarrollo de una toma de decisiones. Primeramente obtendremos datos de una fuente externa en formato OData y veremos cómo podemos interactuar con las fuentes API REST en formato OData. En segundo lugar crearemos nuestra propia fuente de datos OData y la atacaremos con la herramienta Power BI. En ese momento realizaremos estudios de inteligencia de negocio simulando una empresa ficticia y como esta herramienta puede ayudarnos en la toma de decisiones.

# Capítulo 1

## Bussiness Intelligence

---

### Contenido

- 1.1 ¿Qué es Bussiness Intelligence?
- 1.2 Utilidad de Bussiness Intelligence.
- 1.3 Hacer uso de Bussiness Intelligence.
- 1.4 Distintas herramientas para hacer uso de Bussiness Intelligence

### **1.1 ¿Qué es Bussiness Intelligence?**

Bussiness Intelligence, a partir de ahora, inteligencia de negocio que es su significado en castellano, es el conjunto de estrategias, aplicaciones, herramientas, tecnología o arquitecturas técnicas que son utilizadas para conseguir información útil para un negocio gracias a la transformación de datos. Esto quiere decir que gracias a una serie de datos proporcionados por una entidad podremos transformarlos en información útil para la misma.

### **1.2 Utilidad de Bussiness Intelligence**

Está enfocada principalmente a empresas. Gracias a este concepto podemos conseguir una información útil para nuestra entidad, ya sea para mejorar o corregir algún defecto de esta. Con la información adquirida al aplicar esta inteligencia de negocio podremos mejorar la toma de decisiones en la empresa, ya sean de gran importancia o detalles mínimos. Cuanto mejor uso hagamos de esta inteligencia de negocio mejores resultados obtendremos de las decisiones tomadas.

*"La capacidad de comprender las interrelaciones de los hechos presentados de tal forma que consiga orientar la acción hacia una meta deseada" definición de inteligencia.*

*Hans Peter Luhn, investigador de IBM, 1958*

## 1.3 Hacer uso de Bussiness Intelligence

Distintas entidades tienen una definición de Bussiness Intelligence creado para sus propias aplicaciones. Dependiendo del programa que estemos usando podemos aplicar de una manera u otra la inteligencia de negocio.

***"Las mejores decisiones comienzan con un mejor conocimiento del negocio"***

*"Obtenga información más detallada de sus datos. Utilice las eficaces funcionalidades de Business Intelligence (BI) de SQL Server 2016, Azure Analysis Services y Power BI para convertir sus datos complejos en información del negocio y compartir esta información con toda la organización."*

*Microsoft SQL Server 2016*

Con esto podemos demostrar que la inteligencia de negocio es una ciencia que, como ya dijimos anteriormente, hace uso, como en este ejemplo, de herramientas para ser aplicada.

Por otro lado gracias al hacer uso de la inteligencia de negocio conseguiremos grandes beneficios para nuestra empresa. Destacamos tres grandes beneficios:

1. Datos reales. Nadie nos vende nada. Observamos nuestra empresa con datos reales de nuestras propias bases de datos. Nadie los ha manipulado y por tanto sabremos a ciencia cierta qué está ocurriendo.
2. Información muy valiosa de nuestros clientes. Con unos rápidos ejemplos podremos comprobar si nuestros clientes están o no satisfechos entendiendo las gráficas que podemos llegar a generar, sabiendo desde donde se están produciendo más compras o si estamos llegando a usuarios de distintas regiones.
3. Mejor construcción interna. Con una buena herramienta, si estamos aplicando inteligencia de negocio, observaremos si realmente estamos almacenando toda la información deseada de nuestros clientes o si deberíamos guardar más y mejor la información.

*El **Business Intelligence** es, sin lugar a dudas, una de las palabras más populares en el actual panorama de los negocios. Los nuevos modelos de BI son más fáciles de utilizar, tienen interfaces muy simples y se pueden entender aunque no se tengan conocimientos tecnológicos.*

*Saima Solutions, Consultora de Bussiness Intelligence.*

## 1.4 Distintas herramientas para hacer uso de Bussiness Intelligence.

Podemos encontrar distintas herramientas para hacer uso de Bussiness Intelligence, de las cuales podemos destacar las siguientes.

- Qlik: Plataforma enfocada al análisis y visualización de datos. Posee aplicaciones interactivas que permiten el acceso a los datos de cara al usuario. Según el informe generado por la empresa Gartner esta herramienta es la más usada por los usuarios para la visualización de datos.



Figura 1.4.1 Imagen de Qlik. Fuente:

<http://global.qlik.com/~media/Images/Products/slideshow/11-large.ashx?h=268&w=413>

- Tableau: Esta herramienta está enfocada a la visualización interactiva de los datos. Posee una buena cantidad de fuentes a las que se puede conectar, es la herramienta más visual.

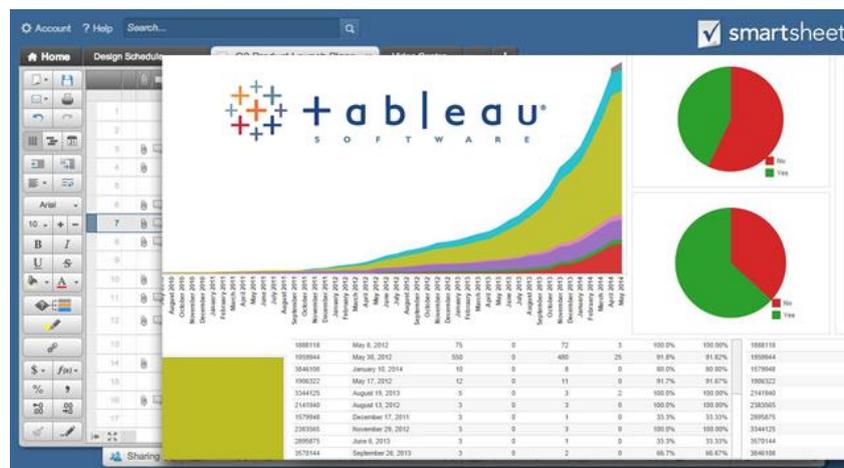


Figura 1.4.2 Imagen de Tableau. Fuente:

<https://d2myx53yhj7u4b.cloudfront.net/sites/default/files/haymaker/screenshot-Tableau-for-Smartsheet.png>

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

- Power BI. Es la elegida para este proyecto, presenta la mayor cantidad de fuentes de datos de conexión, es una herramienta muy potente y está desarrollada por Microsoft.

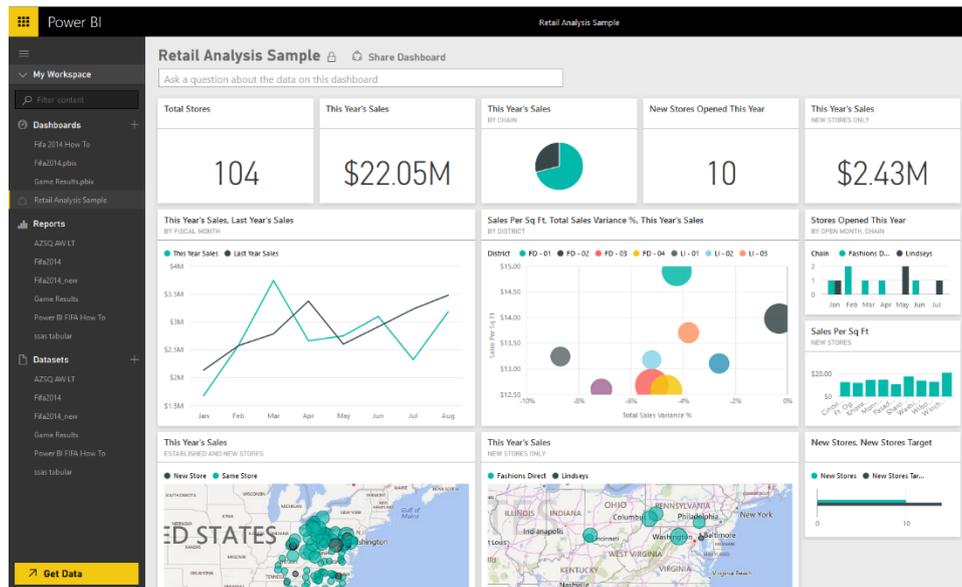


Figura 1.4.3 Imagen de Power BI. Fuente: <http://www.radacad.com/wp-content/uploads/2015/08/43.png>

Podemos encontrar varias herramientas según el ámbito de trabajo donde nos encontremos. Para este proyecto se ha decidido realizar toda la documentación en un entorno de Microsoft. Por ello las herramientas que usaremos son creadas por la entidad de Microsoft. En concreto estamos hablando de SQL Server, Visual Studio, PowerBI y una fuente de datos OData. De todos ellos hablaremos a continuación.

Dependiendo del entorno en el que nos encontremos debemos de elegir unas determinadas herramientas para conseguir una buena inteligencia de negocio y con ella tomar unas buenas decisiones para nuestra empresa.

La persona responsable de hacer uso de las herramientas oportunas para aplicar inteligencia de negocio es denominada "Customer Intelligence Specialist".

*El trabajo que desarrolla un profesional del Business Intelligence consiste en obtener conocimiento de la información para tomar decisiones en base a los datos.*

*Deloitte, Consultora de Business Intelligence.*

## Resumen

En este primer capítulo se ha hablado brevemente de inteligencia de negocio y se ha expuesto nuestro caso en el que usaremos distintas herramientas de la empresa Microsoft para llevar a cabo un estudio sobre inteligencia de negocio en dos empresas simuladas.

Se exponen las grandes ventajas que supone hacer uso de la inteligencia de negocio en nuestras empresas para obtener información sobre nuestros propios datos y analizarlos para poder tomar mejores decisiones.

# Capítulo 2

## Power BI

---

### Contenido

- 2.1 ¿Qué es Power BI?
- 2.2 Funcionamiento de Power BI.
- 2.3 Productos de Power BI.
  - 2.3.1 Power BI Cloud
  - 2.3.2 Power BI Desktop
  - 2.3.3 Power BI Mobile
- 2.4 Fuentes para Power BI.

## 2.1 ¿Qué es PowerBI?

PowerBI es un conjunto de herramientas, desarrollado por Microsoft, que es utilizado para obtener los datos de una empresa sin importar su fuente de datos y analizarlos para obtener informes y paneles de información de la empresa.

Power BI permite conectarse a distintos orígenes de datos y una vez conectado, simplificarlos para generar informes y paneles de información hacia la empresa.

Power BI tiene la característica de poder compartir esta información con el resto de la empresa gracias a su capacidad de adaptarse a distintas plataformas, como veremos más adelante, y puede ser escalado por toda la empresa. Power BI no es único para un ecosistema de Microsoft. Como ya veremos más adelante, asimila fuentes de datos de distintas entidades.

*Microsoft es reconocido como líder en el cuadrante mágico de Gartner para las plataformas de inteligencia empresarial y análisis.*

*Gartner, empresa consultora*

*Power BI gana el premio Editors' Choice en el resumen de la reseña sobre las herramientas de BI con características de autoservicio PCMag.com.*

*PCMAG.*

## 2.2 Funcionamiento de Power BI.

Conectaremos PowerBI a una fuente de información de nuestra empresa, que puede ser un servidor, una base de datos o prácticamente cualquier fuente de información gracias a la potencia que tiene PowerBI y su gran versatilidad.

Una vez conectado a esa fuente podremos tratar esos datos y transformarlos para darle forma en paneles de información para el cliente que ha solicitado su inteligencia de negocio.

Ya que tenemos la información recogida podremos analizarla detenidamente y usarla para tener una buena toma de decisiones

## 2.3 Productos de Power BI.

Vamos a ver una serie de productos que ofrece Power BI y que se han utilizado para este proyecto. Power BI cuenta con más productos según las características que necesitemos. A continuación se muestra un breve resumen de cada uno de los productos y más adelante se detallan.

- Power BI Cloud. Sistema en la nube donde vamos a almacenar nuestros informes y paneles de información, desde aquí la información será repartida al resto de productos.
- Power BI Desktop. Aplicación de escritorio donde desarrollaremos los paneles de información y analizaremos los datos. Una vez realizado esto podremos subir los resultados a la nube.
- Power BI Mobile. Aplicación móvil donde podremos observar los datos que tenemos almacenados en el Cloud pero no podremos realizar modificaciones.

### 2.3.1 Power BI Cloud.

Power BI Cloud es el sistema de negocio que se establece en la nube y al que podremos acceder desde distintos dispositivos. Además servirá como punto de acceso al resto de productos de Power BI.

Desde la figura **2.3.1** podemos observar el entorno de Power BI Cloud. En ella encontramos un panel de navegación a la izquierda, que nos permite movernos por los elementos que tenemos en la nube. Seleccionando áreas de trabajo vemos los proyectos de los cuales tenemos un documento en Desktop y además hemos realizado una publicación, por ello aparecen como proyectos en la nube.



Figura 2.3.1.1 Power BI Cloud

En primer lugar nos muestra los proyectos que tenemos, si accedemos a la pestaña de informes veremos los informes que hemos desarrollado en Power BI Desktop y podremos analizarlos.

En la figura **2.3.1.2** se muestran los informes que tenemos actualmente en la nube. Veamos detenidamente el informe Northwind que se muestra en la figura **2.3.1.3**.

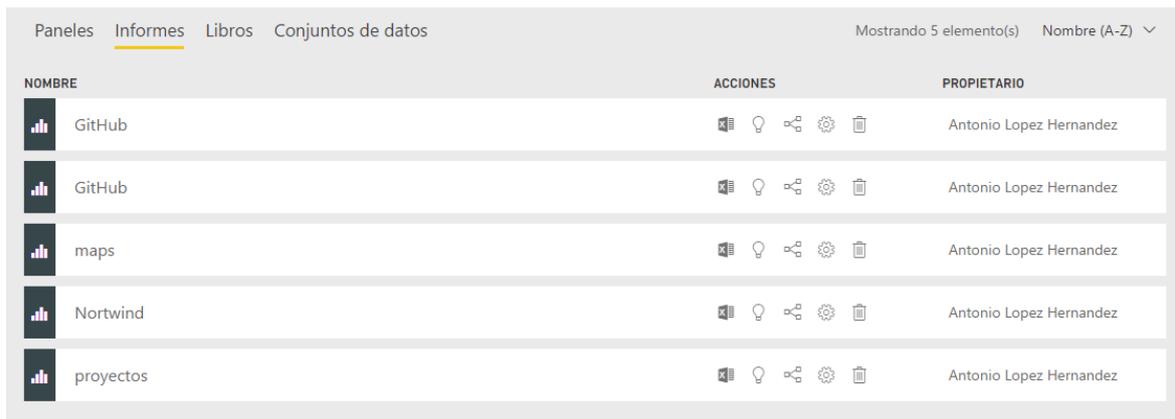


Figura 2.3.1.2 Pestaña de informes

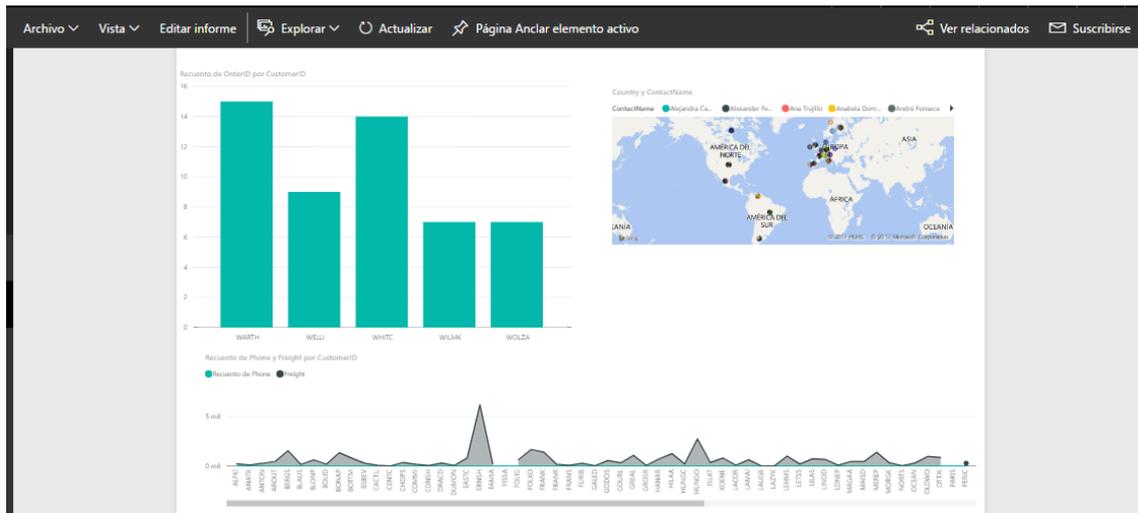


Figura 2.3.1.3 Ejemplo de Informe

Si accedemos a uno de estos informes veremos el panel de información en el que podremos navegar. Tenemos la posibilidad de interactuar con ella, por ejemplo ampliando el mapa o moviéndonos por el gráfico inferior.

## 2.3.2 Power BI Desktop.

Power BI Desktop es la aplicación para escritorio de Windows que nos permitirá conectarnos a las fuentes de información y tratar sus datos. Además nos posibilitará crear los informes y paneles de información para los usuarios.

Power BI Desktop nos permitirá realizar una publicación que subirá el informe a Power BI Cloud el cual se compartirá con el resto de usuarios.

Para este proyecto se hará uso de esta aplicación de escritorio para conectarse a la fuente de datos en API REST que emite los datos en formato OData.

En la figura 2.3.2.1 observamos un ejemplo de la aplicación de Power BI Desktop funcionando en un equipo Windows.

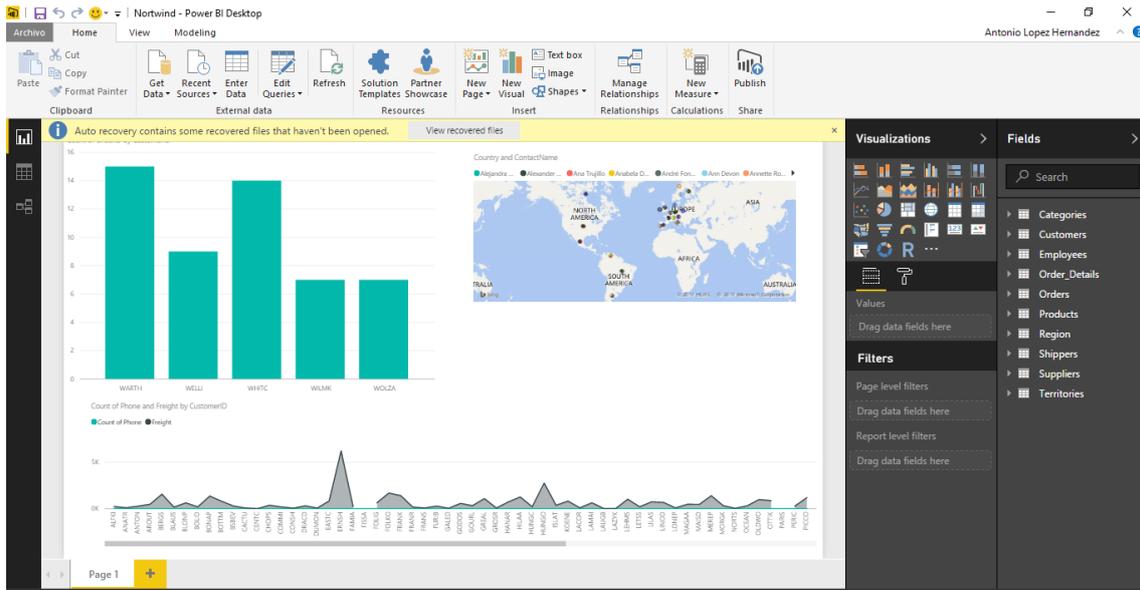


Figura 2.3.2.1 Power BI Desktop

El entorno nos es familiar dado que es un programa de Microsoft. Encontramos un panel de herramientas en la parte superior, un panel de navegación a la izquierda y a la derecha tenemos la característica principal de Power BI, las tablas que hemos obtenido de una base de datos y los distintos métodos que hay para exponer la información. En los ejemplos que se redactan en este proyecto se explican algunos de esos métodos.

En la figura 2.3.2.2 encontramos un ejemplo de una opción que permite Power BI Desktop, que es mostrar las relaciones que existen entre las distintas tablas que hemos importando de la base de datos.

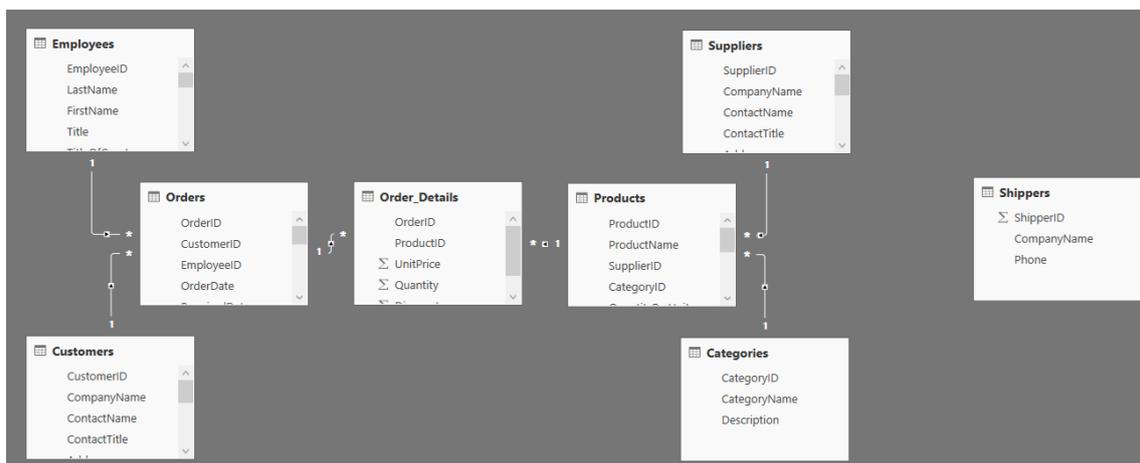


Figura 2.3.2.2 Relaciones de tablas

### 2.3.3 Power BI Mobile.

Power BI Mobile es la APP para Android que nos permite conectarnos a Power BI Cloud para poder acceder a los informes y paneles de información desde cualquier lugar gracias a los dispositivos móviles. En las figuras 2.3.3.1 y 2.3.3.2 encontramos los paneles y los informes.

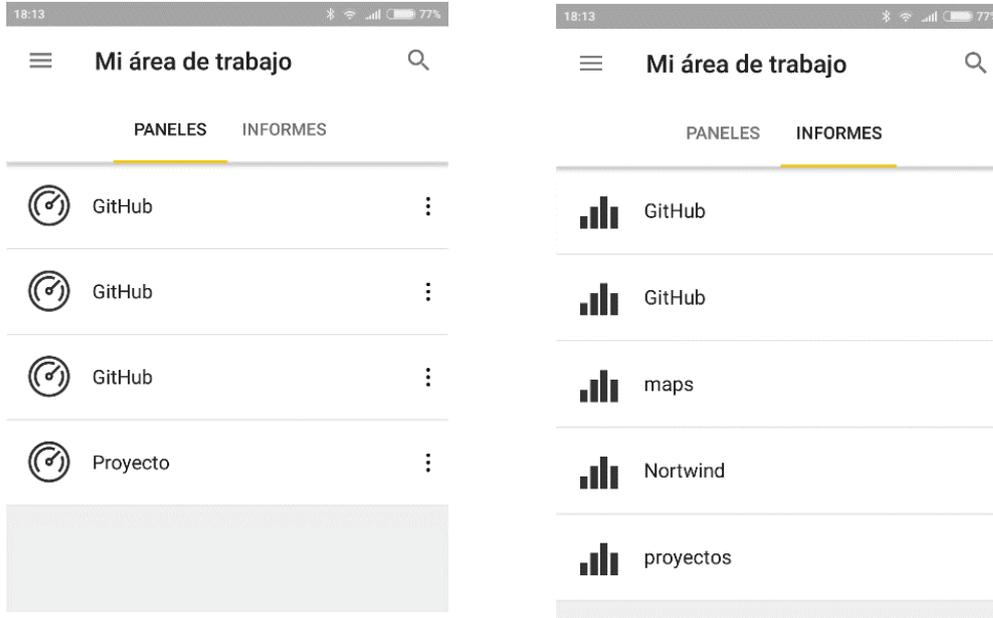


Figura 2.3.3.1 y 2.3.3.2 Power BI Mobile

En la siguiente figura 2.3.3.3 podemos ver el informe Northwind como lo veíamos en el resto de aplicaciones.

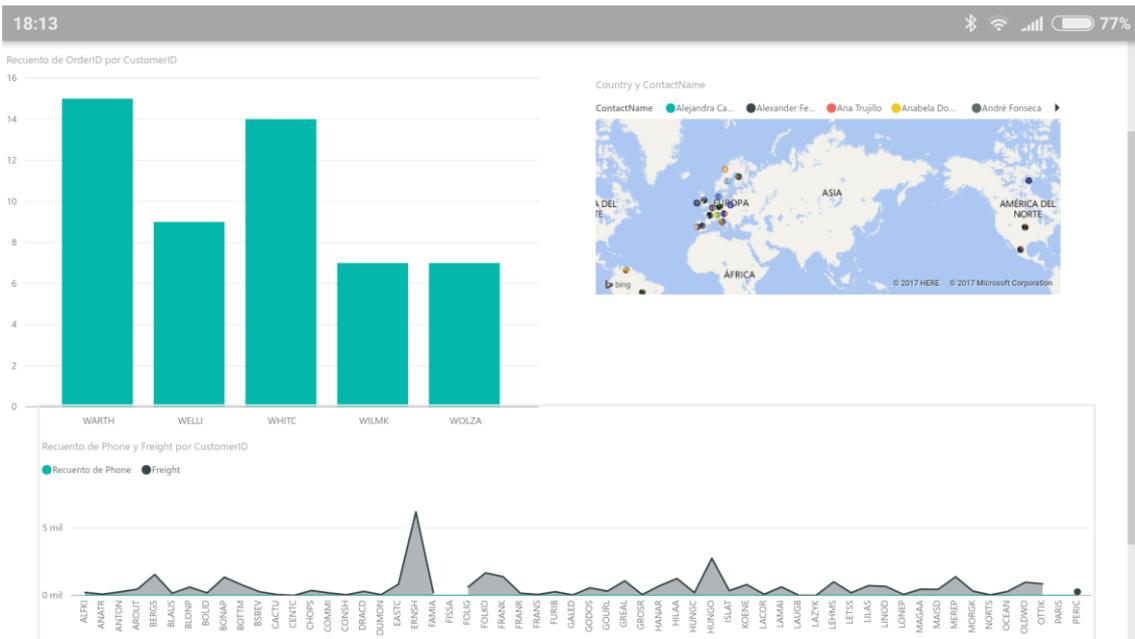


Figura 2.3.3.3 Informe Northwind en Power BI Mobile

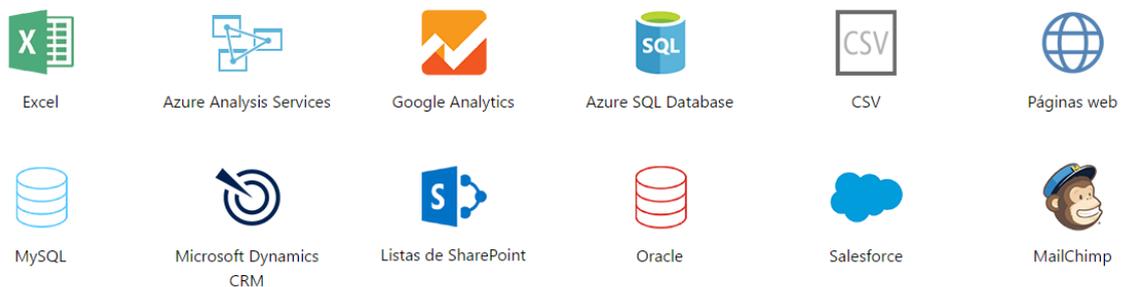
## 2.4 Fuentes de Power BI.

PowerBI tiene la capacidad de conectarse a una inmensa cantidad de fuentes de datos, actualmente se encuentra en expansión y por ello mensualmente aumentan el número de orígenes de datos de donde podremos obtener información.

Posiblemente esta sea la característica más grande de Power BI, no tanto sus herramientas sino su potencia a la hora de conectar con distintas fuentes de datos. Es muy importante que una herramienta pueda adaptarse a distintas fuentes de datos, dado que incluso en una misma empresa podemos encontrar los datos en dos fuentes de datos distintos, no es algo lógico ni recomendado, pero se puede dar el caso. Al hacer uso de una herramienta tan potente como Power BI que termine la conexión a distintas fuentes de datos podríamos incluso juntar toda esa información y crear mejores informes.

En la Figura 2.4.1 podemos observar algunas de las muchas fuentes de datos que Power BI nos ofrece y a la que podremos tener acceso. Entre ellas destacar:

- **MySQL** es una de las mayores fuentes de base de datos junto con **Oracle** y **Azure** que representan las bases de datos de gran tamaño más utilizadas.
- **Excel** los documentos de este tipo se pueden encontrar en multitud de pequeñas y medianas empresas, y poder conectarnos a sus hojas de cálculo nos va a dar un gran soporte.
- **MailChimp**, para la gestión de cuentas de correo y demás información puede llegar a ser muy útil.



*Figura 2.4.1 Ejemplo de orígenes de datos de Power BI*

Como podemos ver en la imagen Power BI no es único de un ecosistema de Microsoft, sino que admite fuentes de datos de diferentes entidades, aunque por otro lado lo que sí es único es que la aplicación de Power BI Desktop y Mobile son para Windows y Android respectivamente.

De todas las distintas fuentes de datos de donde podemos obtener información se ha decidido para este proyecto tener una base de datos en SQL Server y los datos serán obtenidos en PowerBI mediante la consulta a una API REST que devuelve los datos en formato OData.

La elección de fuentes en formato OData se ha decidido debido a la buena manera de exponer los datos mediante un estándar y además desarrollado por Microsoft. OData, como veremos en el siguiente capítulo, proporciona una serie de mejoras para la exposición de datos, no solo para la inteligencia de negocio, sino incluso para el público dado que OData cuenta con una ISO, de la cual ofrecemos información en el siguiente capítulo.

## Resumen

En este capítulo se ha comentado la herramienta Power BI, se han nombrado tres de los productos de Power BI, Desktop, Cloud y Mobile. Hemos podido ver algunas imágenes de todos esos productos y su funcionamiento, aunque se recomienda que se utilicen para comprobar su potencial. También se habla de las fuentes de datos de Power BI, donde se ha decidido seleccionar OData como fuente de datos.

# Capítulo 3

## OData

---

### Contenido

- 3.1 ¿Qué es OData?
- 3.2 Cómo surge OData.
- 3.3 Estructura para OData.
- 3.4 Framework para OData.
- 3.5 RESTier para OData.

### 3.1 ¿Qué es OData?

*Open Data Protocol, (aka **OData**), es una iniciativa impulsada por Microsoft para la exposición de datos como Servicio basándose en estándares de Internet y es la evolución de lo que era anteriormente conocido como ADO.NET Data Services*

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

OData es un tipo de formato para obtener información bien estructurada a través de API REST. Cuando accedemos a una fuente API REST podemos recibir la información en distintos formatos, como puede ser JSON. OData es un tipo de formato desarrollado por Microsoft y aceptado como estándar por OASIS, un grupo de compañías entre las cuales también participa Microsoft.

La ISO que proporciona OData es denominada ISO/IEC 20802. Esta ISO es reciente, se la han otorgado en el año 2017. Podemos obtener toda la información y descargarnos la documentación de la ISO desde este mismo enlace:

[http://standards.iso.org/ittf/PubliclyAvailableStandards/c069208\\_ISO\\_IEC\\_20802-1\\_2016.zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c069208_ISO_IEC_20802-1_2016.zip)

## 3.2 Cómo surge OData.

OData nace para ayudar a una mejor construcción y estructuración de las API REST. Sin tener que preocuparnos de la estructura de la fuente, OData nos devuelve una información a través de las API REST estandarizada para una gran cantidad de herramientas que puedan consumir de ella. Está basado en AtomPub (Un protocolo simple basado en HTTP para crear o actualizar recursos en una web), JSON y HTTP.

Originalmente usaba datos estructurados a través de Atom o JSON pero ha ido evolucionando en estos años para conseguir una nueva versión, la más reciente que muestra los datos en un formato propio, OData V4. Esta última versión mantiene las operaciones que sus antecesores tenían con el GET, POST, PUT y DELETE, funciones parecidas a las que usamos en otras API REST. La modificación se basa en las cabeceras para poder realizar estas funciones.

Desde que se creó OData ha llegado ya a su versión 4. Por tanto para este proyecto se pretende usar este estándar en su última versión. Los cambios los podemos consultar en su página oficial, como pequeño comentario originalmente devolvía los datos estructurados en xml y actualmente posee su propio esquema. Dado que la herramienta para obtener la fuente OData va a ser PowerBI, en su última versión, esta también requiere que los datos que le pasemos sean en formato OData V4.

## 3.3 Estructura para OData.

Se va a presentar la estructura que posee OData y un breve ejemplo de unas consultas con GET, POST y DELETE, que deben lanzarse con la herramienta Postman, un conjunto de herramientas para los desarrolladores de API, para interactuar con la base de datos a través de la API REST en formato OData.

Como hablábamos anteriormente vamos a hacer uso de OData V4, que actualmente tiene una forma similar a la mostrada en la figura 3.3.1.

```
HTTP/1.1 200 OK
Content-Length: 1007
Content-Type: application/json; odata.metadata=minimal
OData-Version: 4.0
{
  '@odata.context': 'http://services.odata.org/V4/(S(a2k31bgwiyejn2j2iibvq4p))/TripPinServiceRW/$metadata#People',
  '@odata.nextLink': 'http://services.odata.org/V4/(S(a2k31bgwiyejn2j2iibvq4p))/TripPinServiceRW/People?%24skiptoken=8',
  'value': [
    {
      '@odata.id': 'http://services.odata.org/V4/(S(a2k31bgwiyejn2j2iibvq4p))/TripPinServiceRW/People('russellw
hyte')',
      '@odata.etag': 'W/'08D1D5BD423E5158'',
      '@odata.editLink': 'http://services.odata.org/V4/(S(a2k31bgwiyejn2j2iibvq4p))/TripPinServiceRW/People('ru
collubate')'
```

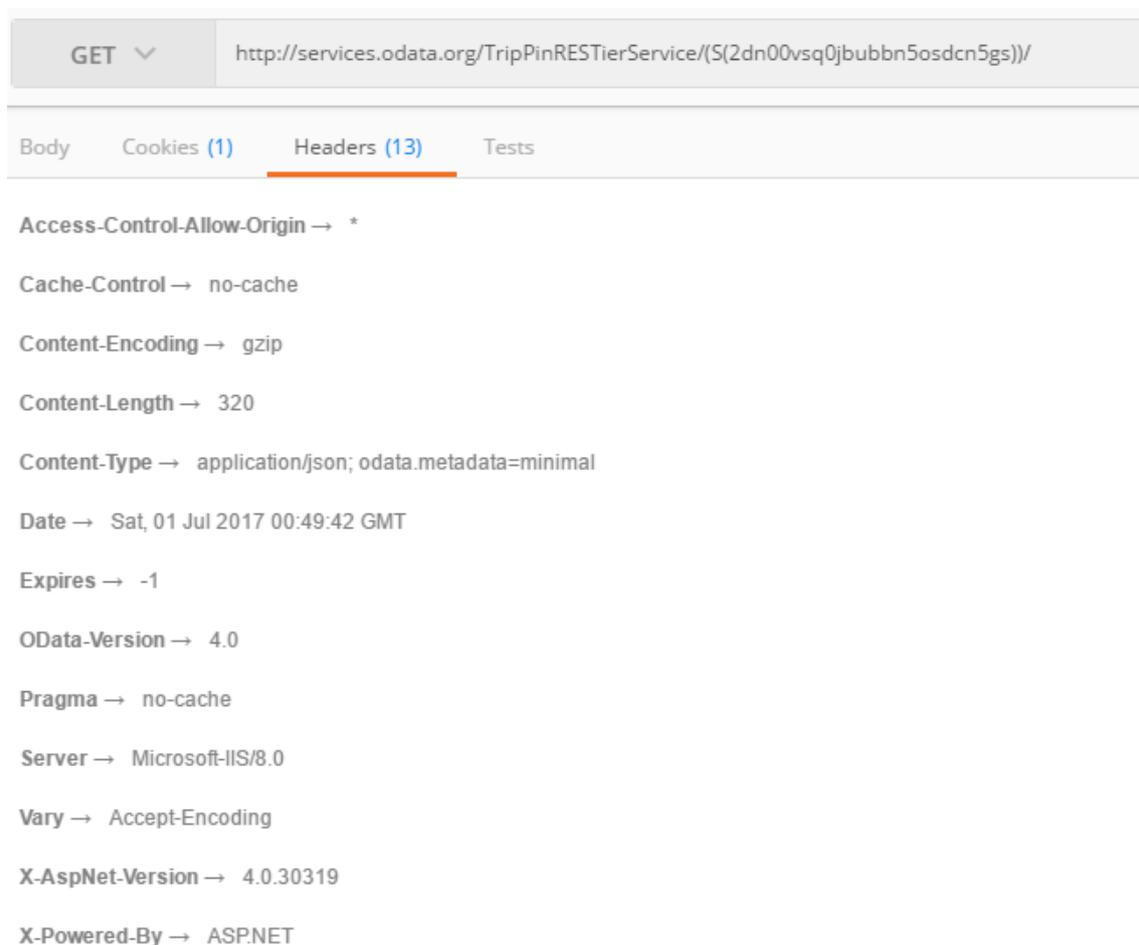
Figura 3.3.1 Presentación de una fuente OData.

Esta figura y muestra de presentación de código en OData esta obtenida de la página oficial de OData, la cual también nos proporciona un servicio OData.

La información relevante viene a continuación de la etiqueta value, donde comienza indicando la estructura de los datos que vamos a recibir. Precediéndola encontramos unas etiquetas que indican donde encontrar la información del Metadata.

Para verlo un poco mejor vamos a lanzar una consulta con la herramienta Postman y vamos a utilizar una fuente de datos que nos proporciona la misma página de OData, esto es un pequeño ejemplo, el cual encontramos completo como desarrollarlo en el **capítulo 4**. <http://services.odata.org/V4/TripPinServiceRW>

Podemos observar en la figura **3.3.2** que en la herramienta Postman debemos de introducir unas cabeceras para poder hacer uso de API REST cuando tratamos con datos OData, esto se mostrará con mayor detalle en el **capítulo 4**.



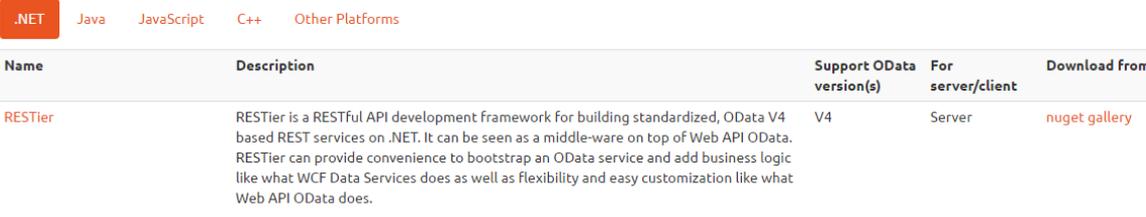
*Figura 3.3.2 Header para OData en Postman*

## 3.4 Framework para OData.

OData no desarrolla ningún software propio para pasar nuestros datos a formato OData, pero nos da una gran cantidad de frameworks desarrollados por terceros que si lo consiguen.

Estos framework están divididos en distintas pestañas dependiendo del entorno donde se hayan desarrollado, como podemos observar en la figura **3.4.1**:

- .Net, aplicaciones destinadas principalmente a ser librerías para Visual Studio.
- Java, la mayoría son aplicaciones php.
- JavaScript, la mayor parte son aplicaciones php para web.
- C++, es un solo proyecto en github para una aplicación cliente.
- Other Plataforms, como por ejemplo Python.



Name	Description	Support OData version(s)	For server/client	Download from
RESTier	RESTier is a RESTful API development framework for building standardized, OData V4 based REST services on .NET. It can be seen as a middle-ware on top of Web API OData. RESTier can provide convenience to bootstrap an OData service and add business logic like what WCF Data Services does as well as flexibility and easy customization like what Web API OData does.	V4	Server	<a href="#">nuget gallery</a>

*Figura 3.4.1 Ejemplo de la plataforma*

Se han probado varios de los framework que ofrece OData, con este pequeño resultado:

- Java, Apache Olingo, aunque la presentación y la documentación eran excelentes no funcionaba correctamente a la hora de ponerlo a funcionar.
- Java, SDL OData Frameworks, la documentación era escasa, y siguiendo los pasos que indican no se consigue nada.
- Java, OData4j, aunque la documentación podría ser mejor ayuda lo suficiente para hacer funcionar el proyecto, no nos interesó porque daba resultados para versiones de OData de la 1 a la 3.
- C++ y Other Plataforms se descartaron dado que no ofrecían una versión única a modo de servidor.
- .Net, RESTier, con mucha documentación, explicación y guías paso a paso además de una sencilla instalación.

Entre los distintos frameworks que OData nos proporciona, para este proyecto, se ha seleccionado el framework denominado Restier, mostrado en la figura **3.4.1**.

La decisión se ha tomado comenzando por seleccionar de entre todas las categorías una que tuviera que ver con el ámbito de Microsoft, por tanto nos decantamos por los paquetes .NET.

A continuación se seleccionaron aquellos frameworks que dieran unos datos en formato OData V4 dado que los datos van a ser enviados a Power BI el cual solo admite los datos en formato V4.

Por último se seleccionaron aquellos frameworks que fueran un servidor, para poder producir nosotros nuestros datos en formato OData.

Como instancia final nos quedaba RESTier, que además se encuentra en la galería de NuGet, por tanto su instalación era sencilla como ya veremos en el siguiente capítulo.

## 3.5 RESTier para OData.

Encontramos toda la documentación necesaria para RESTier en su página oficial, <http://odata.github.io/RESTier/>. En ella nos ofrece una guía paso a paso para la instalación de su framework, la cual no es necesario consultar dado que nosotros ofrecemos una guía similar y detallada para la instalación del mismo.

En el capítulo **3.3** estábamos haciendo uso del programa Postman para realizar una consulta a la base de datos que proporciona OData, se ha intentado realizar consultas de POST, PUT y DELETE pero no se obtiene resultado alguno, siempre proporcionan el mismo error.

No es posible acceder a las clases de RESTier dado que es privado y lo único que recibiremos en nuestro proyecto será una asociación en el using.

```
using System.Web.Http;
using Microsoft.Restier.Providers.EntityFramework;
using Microsoft.Restier.Publishers.OData;
using Microsoft.Restier.Publishers.OData.Batch;
namespace Northwnd
{
```

*Código 3.5.1 Imagen del código en Visual Studio*

Por otro lado para comprender el funcionamiento recomiendo la clase ofrecida por Java que si nos permite descargar el código abierto y que en esencia trabajan en lo mismo, mapear los objetos de la base de datos como variables para tratarlos, transformarlos y tras ello realizar inserciones en la base de datos.

## Resumen

En este tercer capítulo se nombra OData, la fuente de datos que hemos seleccionado. Se han estudiado los distintos frameworks que OData ofrece, se ha decidido hacer uso del framework Restier por funcionar como servidor, mostrar los datos en OData V4 y una instalación sencilla en proyectos de Visual Studio. Se ha mostrado brevemente el esquema que sigue OData y cómo podemos realizar consultas REST en formato OData gracias a Postman.

# Capítulo 4

## Fuente OData Externa

---

### Contenido

- 4.1 Fuentes externas OData.
- 4.2 Consumo fuentes OData externas.

### 4.1 Fuentes externas OData.

No se han encontrado muchas fuentes de consumo público en formato OData. Esto es debido a que son utilizadas para el negocio y no son públicas ni accesibles para todo el mundo.

Por suerte OData proporciona unas fuentes propias y en distintos formatos OData. Esta fuente es bastante útil ya que proporciona un servidor dedicado para cada usuario que lo solicita, por tanto podremos tratar la base de datos como si fuera de nuestra propiedad, eliminando, añadiendo y modificando lo que creamos oportuno.

Sería interesante que más empresas mostraran parte de su información con el exterior, pero en la mayoría de casos esa información está clasificada y es confidencial, por tanto nos va a costar encontrar una fuente externa. Nuestra solución ha sido, como veremos en el **capítulo 5**, crear nuestra propia fuente de datos en formato OData, mostrando solo las tablas que queremos mostrar al público y reservándonos las que podemos considerar críticas.

En el siguiente punto se va a mostrar como consumir estas fuentes externas y cualquier otra, y en el siguiente capítulo crearemos y consumiremos nuestra propia fuente de datos OData.

### 4.2 Consumo fuentes OData externas.

Vamos a introducir las fuentes OData que son accesibles actualmente y cómo podemos hacer uso de ellas, además desarrollaremos un proyecto en Postman donde haremos uso de la fuente de datos proporcionada por OData con una guía paso a paso para entender cómo funciona la API REST cuando tratamos con datos en formato OData.

La conexión a una fuente existente de OData es bastante sencilla, primero debemos saber cuál es el enlace principal. Como ejemplo usaremos la base de datos que ofrece el propio servidor OData para la versión V4. Para poder hacer uso de ella accederemos a la página oficial de OData y accederemos al panel de navegación, en él iremos a la sección de desarrolladores y nos introduciremos en el apartado referencias de servicio, como se muestra en la figura 4.1.1. Una vez aquí accederemos al apartado de OData V4.

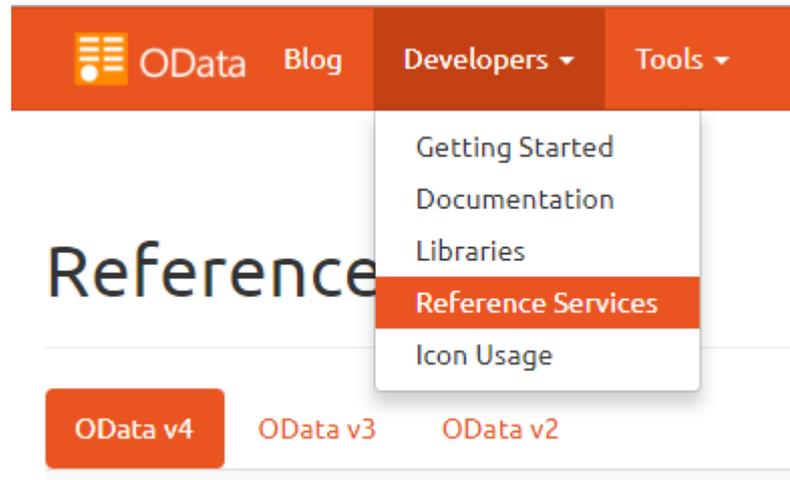


Figura 4.2.1 Referencias de Servicio

Nos encontramos ante una página como la que se muestra en la figura 4.2.2.

## Reference Services

<a href="#">OData v4</a> <a href="#">OData v3</a> <a href="#">OData v2</a>	
Name	Description
<a href="#">TripPin(read/write)</a>	This OData V4 sample service is built with <a href="#">Restier</a> which is a turn-key located at <a href="#">this link</a> .

## Basic Usages

[▶ Run in Postman](#)

For usage of advanced scenarios, please refer to [Trippin Advanced Usages](#).

## Requesting Data

### Requesting EntitySet

```
GET http://services.odata.org/TripPinRESTierService/People
```

Figura 4.2.2 Entorno de OData que ofrece fuentes

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

Desde aquí tendremos acceso a dos aspectos que nos ofrece la misma base de datos, por un lado si accedemos a la opción Run in Postman se nos ofrece descargar para Postman una serie de comandos que podemos utilizar para aprender a hacer consultas en formato OData, como ya se explicó en el **capítulo 3**. Si en la columna de Name accedemos al nombre en Rojo de **TripPin(read/write)**. Accediendo a este último nos devolverá un enlace en el que contaremos con la base de datos TripPin en un servidor dedicado para nosotros, donde podremos realizar pruebas y se verá modificado (**Figura 4.2.3**).



*Figura 4.2.3 Nuestro servidor dedicado OData*

En este caso nuestro enlace es:

[http://services.odata.org/TripPinRESTierService/\(S\(0tmew1igbqpwnjpbpanri41j\)\)/](http://services.odata.org/TripPinRESTierService/(S(0tmew1igbqpwnjpbpanri41j))/)

En Rojo vemos remarcado cual es el servidor dedicado que tenemos para pruebas. Una vez que tenemos este enlace podremos tratar la base de datos como queramos, eliminar, insertar o modificar contenido de la misma. Esta información es volátil. Cuando cerremos la conexión, el servidor será eliminado y no habremos almacenado nada de información. A continuación se expone un ejemplo de cómo tratar con esta fuente proporcionada por OData.

Vamos a realizar una serie de pruebas con la herramienta Postman hacia el servidor ofrecido por OData para poder ver la estructura y funcionamiento de OData. Vamos a comenzar por lanzar una consulta GET a la página y ver qué contenido nos devuelve el servidor. Tras ello vamos a insertar contenido en la base de datos con la opción POST y vamos a eliminar el contenido con la opción DELETE como veremos a continuación.

Previos a comenzar se muestra en la figura **4.2.4** un pequeño esquema de la base de datos a la que vamos a realizar las consultas, este esquema se ha conseguido gracias a la generación de esquemas de Power BI Desktop como vimos en el capítulo 2. Como podemos observar en la figura no hay relación entre las tablas pero podemos ver los nombres y contenidos de las mismas.

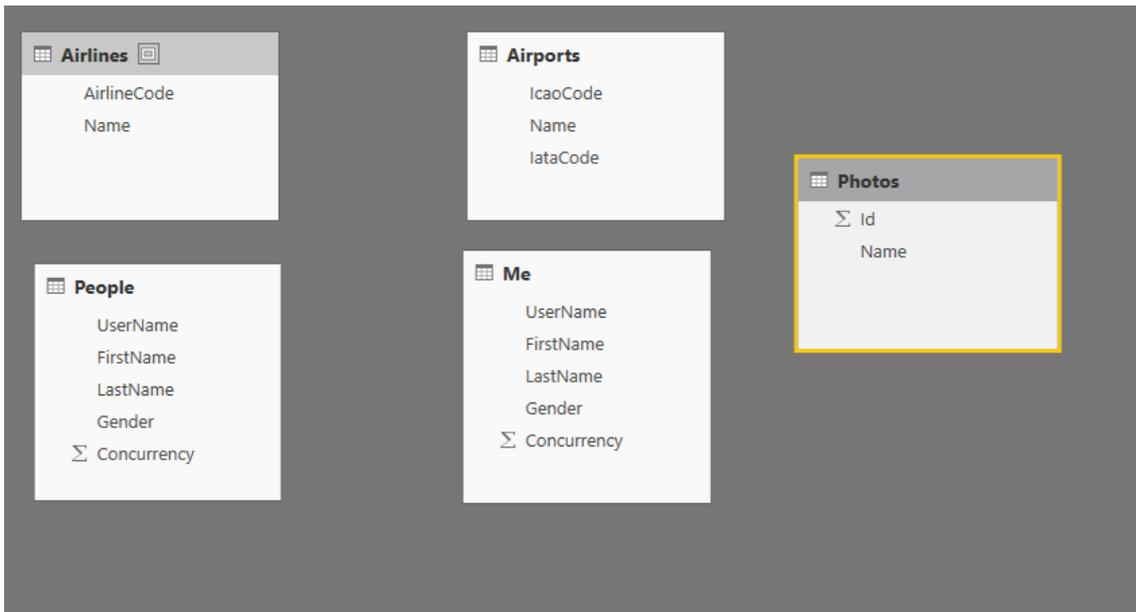


Figura 4.2.4 Muestra de la base de datos

Vamos a lanzar una consulta GET al servidor principal ofrecido por OData como se muestra en la figura 4.2.5, este nos devuelve el nombre de las tablas principales de la base de datos a la que estamos accediendo.

```

1  {
2    "@odata.context": "http://services.odata.org/TripPinRESTierService/(S(2dn00vsq0jbubbn5osdcn5gs))/$metadata",
3    "value": [
4      {
5        "name": "People",
6        "kind": "EntitySet",
7        "url": "People"
8      },
9      {
10       "name": "Airlines",
11       "kind": "EntitySet",
12       "url": "Airlines"
13     },
14     {
15       "name": "Airports",
16       "kind": "EntitySet",
17       "url": "Airports"
18     },
19     {
20       "name": "NewComePeople",
21       "kind": "EntitySet",
22       "url": "NewComePeople"
23     },
24     {
25       "name": "Me",
26       "kind": "Singleton",
27       "url": "Me"
28     }
29   ]
30 }

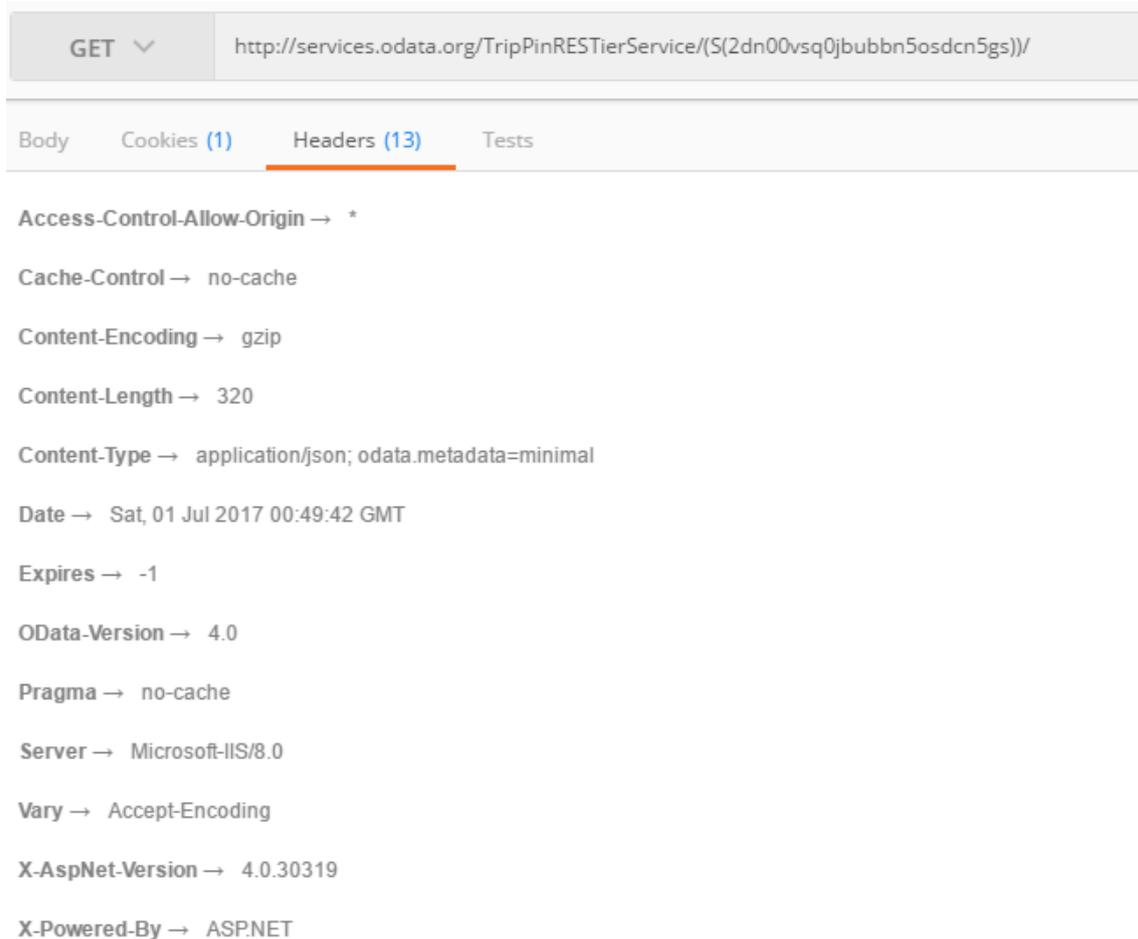
```

Figura 4.2.5 Muestra de OData con Postman

Como podemos observar nos devuelve una etiqueta `odata.context` que buscará dentro de la misma fuente donde hemos realizado la consulta una devolución de metadata. Los nombres devueltos en la figura 4.2.5 coinciden con el esquema que mostraba en la figura 4.2.4.

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

Podemos observar también que además de la consulta que nos devuelve nos acompañan una serie de valores en el Header mostrados en la figura 4.2.6 y muestran algunos de los datos que necesitaremos para poder realizar consultas a la API REST de OData.



*Figura 4.2.6 Header de la consulta*

A continuación vamos a realizar una ejecución GET de la tabla People para ver su contenido como se muestra en la figura 4.2.7, podemos observar en el buscador que no se encuentra ningún usuario con el nombre Antonio, esto lo hacemos para comprobar que cuando hagamos la función POST el usuario será encontrado.

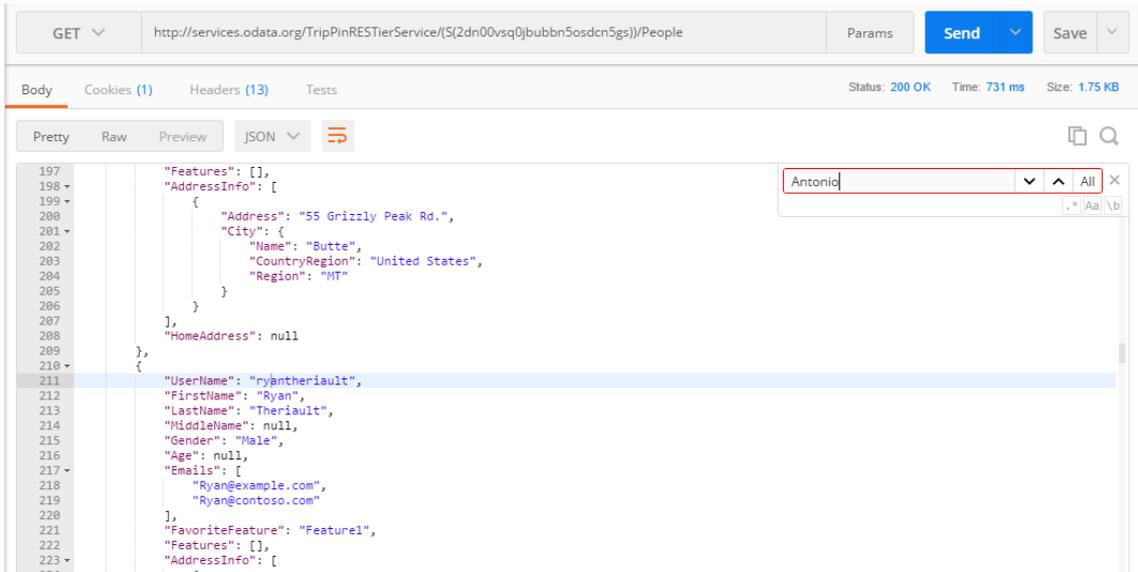


Figura 4.2.7 GET de la tabla People

A continuación vamos a realizar una función POST para insertar una persona en esta tabla, así mostraremos como debe de hacerse en Postman para fuentes OData.

Necesitaremos añadir la siguiente línea al Header (figura 4.2.8) y para añadir una nueva persona añadiremos en el campo Body el código que se muestra en la figura 4.2.9.

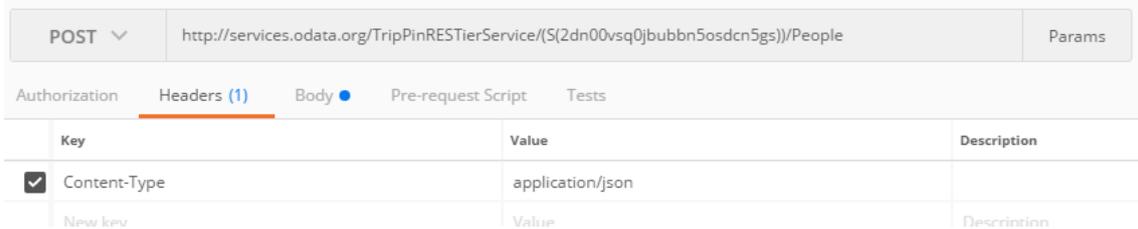


Figura 4.2.8 Header para realizar un POST en Postman

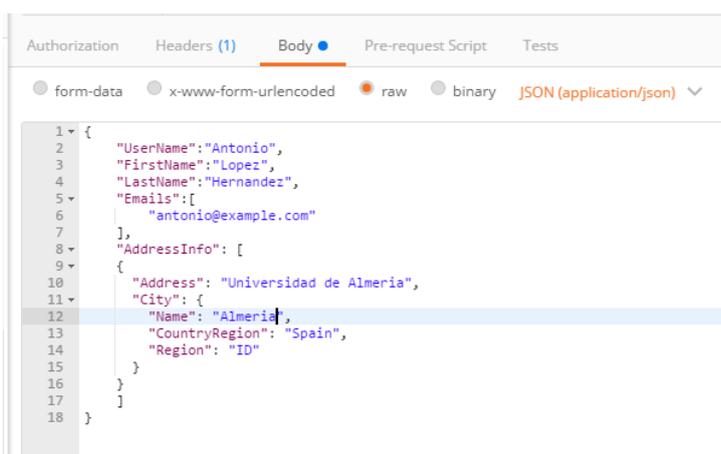
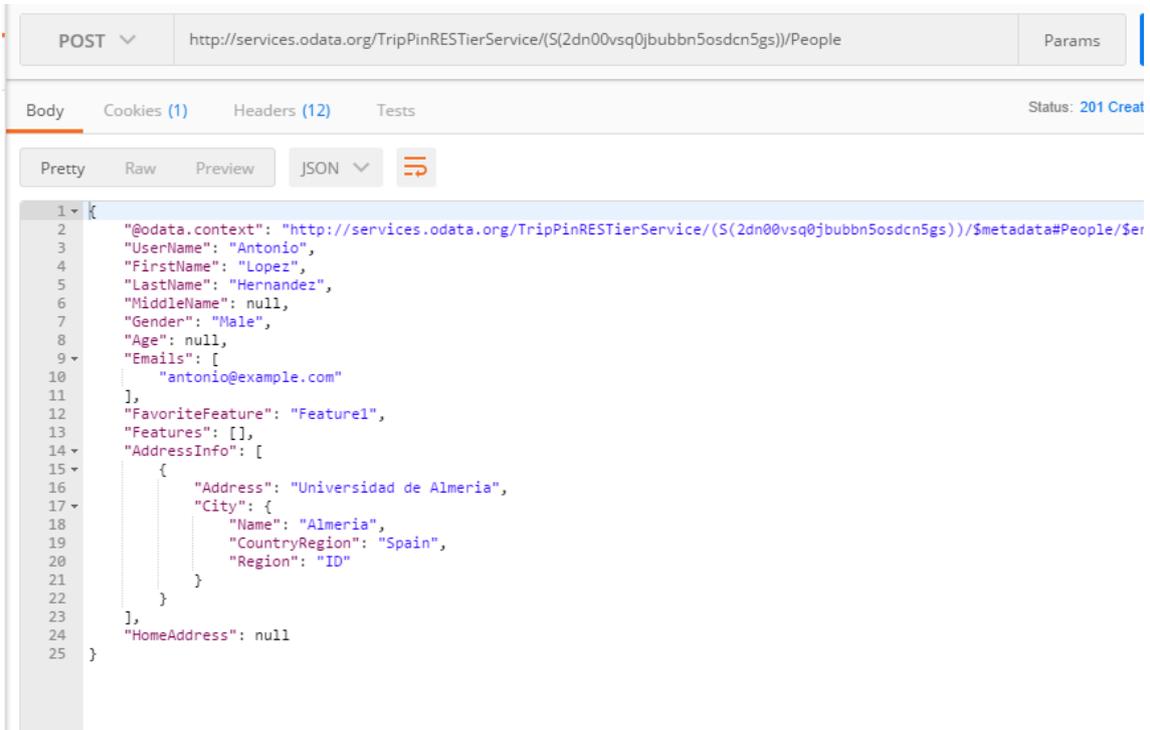


Figura 4.2.9 Body para función POST

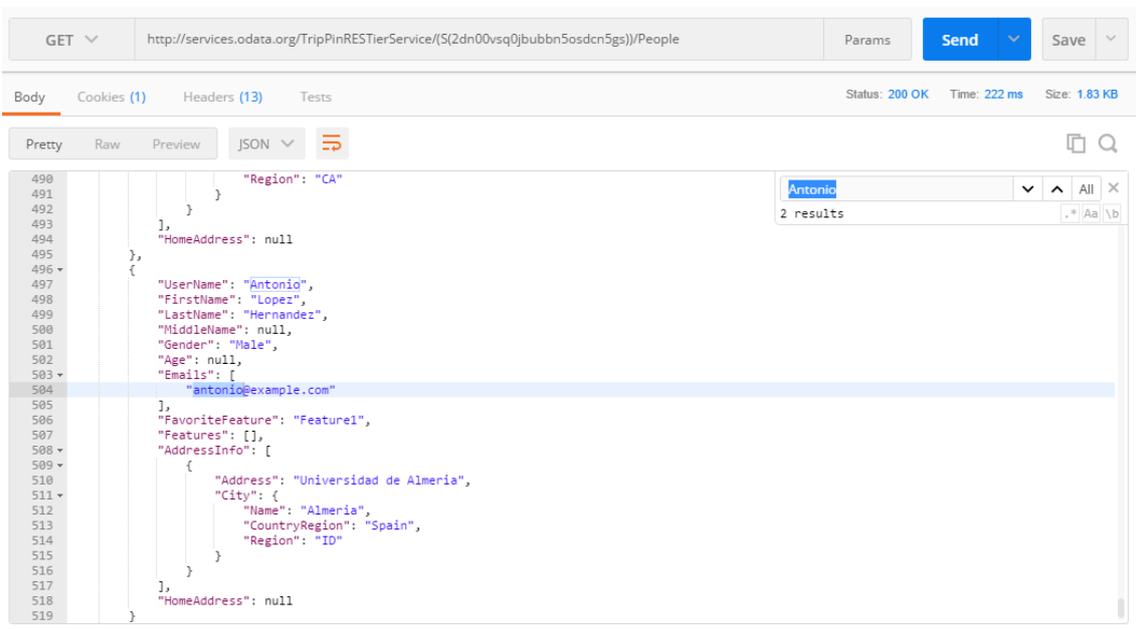
## PRESENTACIÓN DE DATOS ODATA EN POWER BI

Tras realizar la ejecución nos devolverá un código de éxito como el que se muestra en la figura 4.2.10 ahora habremos insertado un usuario a modo de prueba en el sistema, y podemos comprobar que ha sido insertado realizando una opción de GET como se muestra en la figura 4.2.11.



```
POST http://services.odata.org/TripPinRESTierService/(S(2dn00vsq0jbubbn5osdcn5gs))/People Params
Body Cookies (1) Headers (12) Tests Status: 201 Created
Pretty Raw Preview JSON
1 {
2   "@odata.context": "http://services.odata.org/TripPinRESTierService/(S(2dn00vsq0jbubbn5osdcn5gs))/$metadata#People/$entity",
3   "UserName": "Antonio",
4   "FirstName": "Lopez",
5   "LastName": "Hernandez",
6   "MiddleName": null,
7   "Gender": "Male",
8   "Age": null,
9   "Emails": [
10    "antonio@example.com"
11  ],
12  "FavoriteFeature": "Feature1",
13  "Features": [],
14  "AddressInfo": [
15    {
16      "Address": "Universidad de Almeria",
17      "City": {
18        "Name": "Almeria",
19        "CountryRegion": "Spain",
20        "Region": "ID"
21      }
22    }
23  ],
24  "HomeAddress": null
25 }
```

Figura 4.2.10 Resultado del comando POST



```
GET http://services.odata.org/TripPinRESTierService/(S(2dn00vsq0jbubbn5osdcn5gs))/People Params Send Save
Body Cookies (1) Headers (13) Tests Status: 200 OK Time: 222 ms Size: 1.83 KB
Pretty Raw Preview JSON
490   "Region": "CA"
491   }
492   },
493   "HomeAddress": null
494   },
495   {
496     "UserName": "Antonio",
497     "FirstName": "Lopez",
498     "LastName": "Hernandez",
499     "MiddleName": null,
500     "Gender": "Male",
501     "Age": null,
502     "Emails": [
503       "antonio@example.com"
504     ],
505     "FavoriteFeature": "Feature1",
506     "Features": [],
507     "AddressInfo": [
508       {
509         "Address": "Universidad de Almeria",
510         "City": {
511           "Name": "Almeria",
512           "CountryRegion": "Spain",
513           "Region": "ID"
514         }
515       }
516     ],
517     "HomeAddress": null
518   }
519 }
```

Figura 4.2.11 Resultado de la ejecución de POST mostrado con GET

Hemos realizado una inserción en la base de datos gracias a la API REST en formato OData a una fuente que no nos pertenece, a continuación vamos a eliminar los cambios realizados y volveremos a comprobar que ha tenido éxito lanzando otra consulta GET.

Para realizar una opción de DELETE tendremos que seleccionarla e indicar la ID que queremos eliminar en el mismo texto, como se muestra en la figura 4.2.12. A continuación vamos a comprobar que se ha eliminado correctamente con otro GET (4.2.13).

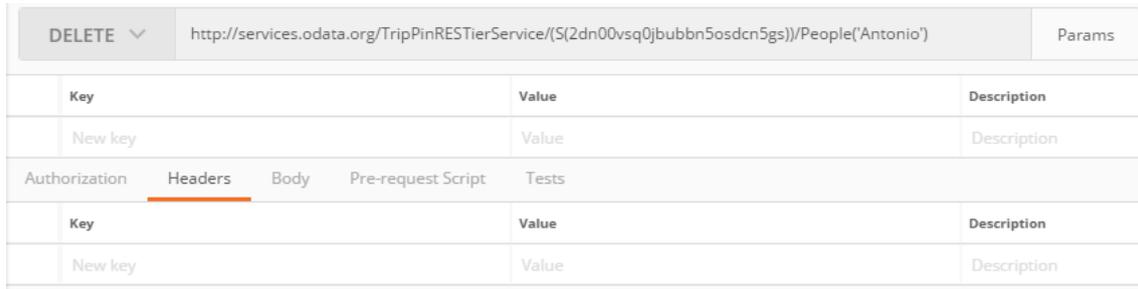


Figura 4.2.12 Ejecución del comando DELETE



Figura 4.2.13 Comprobación de la ejecución DELETE

De esta manera hemos comprobado cómo funciona OData desde la aplicación Postman.

Tras realizar esta serie de operaciones que como hemos mostrado son sencillas y rápidas y estamos modificando una base de datos, parece interesante, una vez que sabemos consumir la fuente, crear la nuestra propia y así poder tratarlos. No solo para nuestro consumo propio de API REST, además podremos dar una buena imagen de empresa si tenemos nuestro propio servidor que oferta API REST y devuelve los datos en un estándar como es OData. Esto puede ser muy útil para adquirir algunas ISO para nuestra empresa o para mostrar de una forma más eficiente los datos ante posibles inspecciones. Además, crear nuestra propia fuente puede sernos de gran utilidad.

## Resumen

En este capítulo se ha accedido a una fuente OData externa y se ha tratado con ella. Se expone cómo deben realizarse las consultas frente a fuentes OData mediante la herramienta Postman y se han mostrado unos ejemplos paso a paso.

# Capítulo 5

Northwnd

---

## Contenido

- 5.1 Introducción al ejemplo.
- 5.2 Agregación de la Base de Datos a SQL Server 2014.
- 5.3 Creación proyecto Visual Studio.
- 5.4 Conexión del proyecto con la base de datos.
- 5.5 Instalación del framework Restier desde NuGet.
- 5.6 Modificaciones pertinentes y ejecución del servicio.
- 5.7 Consumo de la fuente mediante Power BI.
- 5.8 Conclusiones finales.

## 5.1 Introducción al ejemplo.

Para mejorar la comprensión del proyecto y para poder comprobar el funcionamiento del mismo a continuación se expone una guía paso a paso para crear un proyecto a modo de ejemplo que nos servirá para explicar cómo podemos crear y tratar el servicio. No se hace especial interacción ni examinación del framework, la guía paso a paso está adaptada para personas que no tengan mucha experiencia en el ámbito de la informática, siguiéndolas conseguirán conectar su base de datos a un proyecto Visual Studio que emitirá los datos en formato OData y una vez creada la API REST será enviada a Power BI. Lea las instrucciones detenidamente sin saltar pasos, lea el contenido adicional escrito para las modificaciones pertinentes para la adaptación del proyecto a su caso.

En este caso vamos a suponer el ejemplo de querer exponer parte de nuestra base de datos en un servidor REST, este mostrará los datos en formato OData, además accederemos a nuestro servidor a través de la herramienta Power BI para analizarla y buscar posibles mejoras en la empresa.

Los pasos a seguir serán los siguientes:

**5.2 Agregación de la Base de Datos a SQL Server 2014**, vamos a agregar nuestra base de datos a SQL Server para poder conectarlo con nuestro proyecto de Visual Studio.

**5.3 Creación proyecto Visual Studio**, crearemos un proyecto Visual Studio que funcionará como servidor para la API REST, por tanto tiene que tener conexión con la base de datos creada en el anterior punto.

**5.4 Conexión del proyecto con la base de datos**, realizaremos la conexión de la base de datos creada con el proyecto creado.

**5.5 Instalación del framework Restier desde NuGet**, vamos a instalar el framework Restier que transformará nuestros datos en formato OData y los expondrá como fuente.

**5.6 Modificaciones pertinentes y ejecución del servicio**, tenemos que modificar algunos archivos para que tenga conexión con la base de datos además de seleccionar donde actuará nuestra API Rest.

**5.7 Consumo de la fuente mediante Power BI**, dado que poseemos el proyecto en ejecución proporcionado una fuente OData vamos a acceder a ella mediante Power BI para recuperar las tablas creadas en SQL.

**5.8 Conclusiones finales**, se proponen una serie de ejemplos ahora que tenemos los datos de nuestra empresa conectados a Power BI.

## 5.2 Agregación de la Base de Datos a SQL Server 2014.

Vamos a comenzar copiando nuestra base de datos a SQL Server, esto lo vamos a hacer para poder acoplar la base de datos a un proyecto en Visual Studio que será el encargado de actuar como servidor y ofrecer la API REST con datos en formato OData al exterior.

En este apartado vamos a comenzar una guía paso a paso para poder montar la base de nuestro proyecto. Vamos a comenzar por descargarnos la base de datos, en este caso se trata de Northwnd, para no tener problemas de compatibilidad vamos a descargar la versión de backup.

Desde este enlace descargamos la base de datos Northwnd.

<http://download-codeplex.sec.s-msft.com/Download/Release?ProjectName=northwinddatabase&DownloadId=269237&FileTime=129576041042800000&Build=21050>

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

Descomprimos el archivo y obtendremos un archivo Northwnd.bak, abriremos la aplicación de SQL Server y agregaremos la base de datos. Como se muestra en la figura 5.2.1.

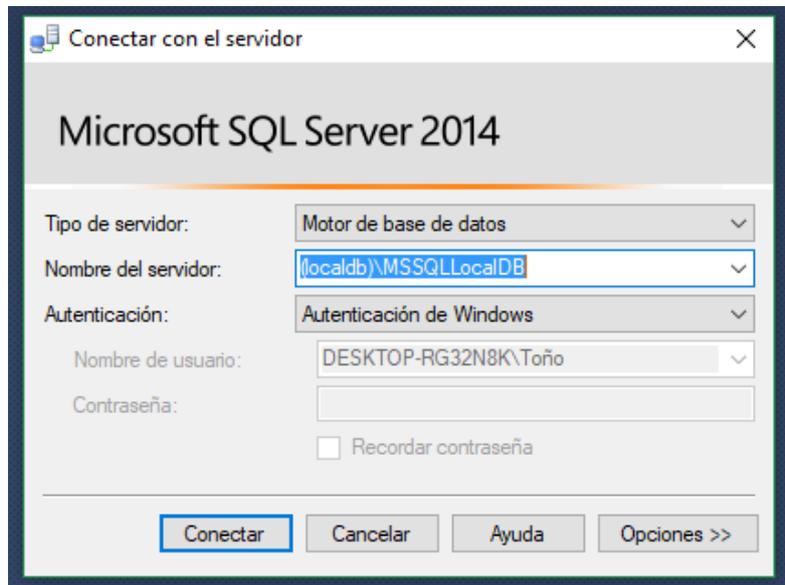


Figura 5.2.1 Conectando SQL Server 2014

En nuestro caso vamos a restaurar la base de datos dado que el archivo descargado es un backup, como se muestra en la figura 5.2.2.

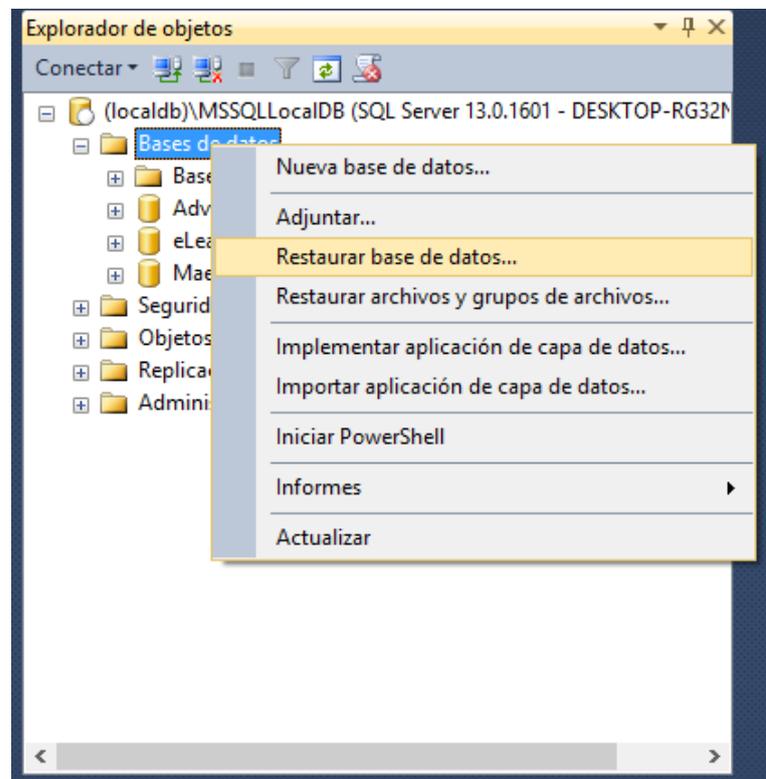
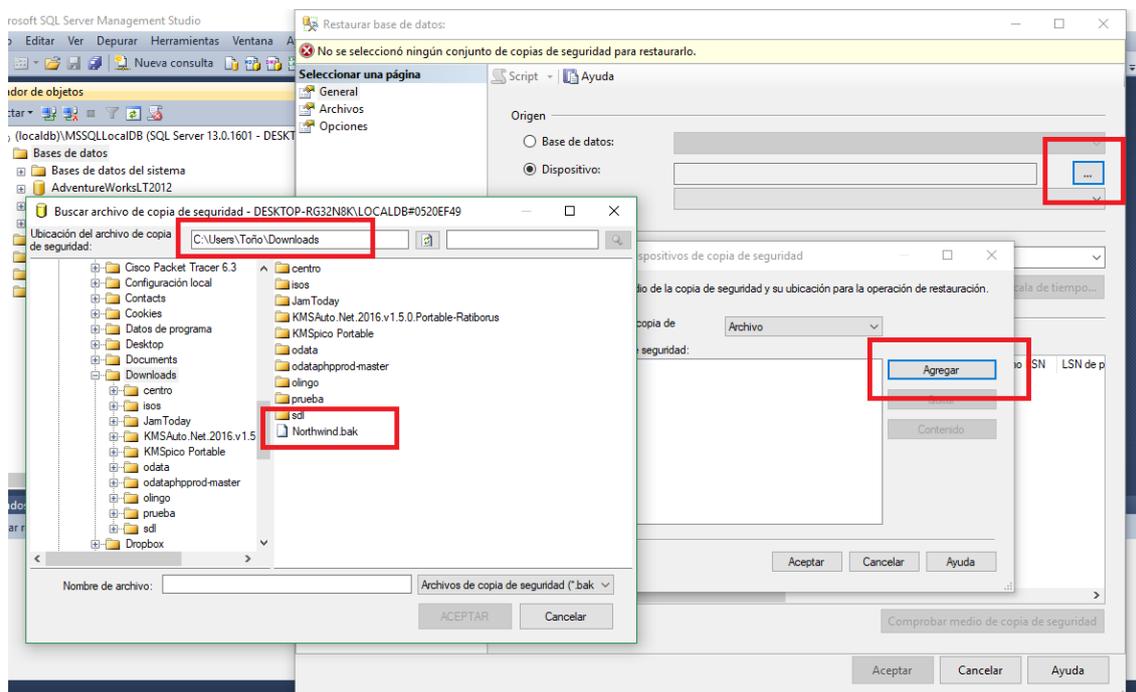


Figura 5.2.2 Restaurar Base de Datos

Si en lugar de restaurar la base de datos tenemos una creada deberemos de utilizar la opción de Adjuntar y seleccionar la dirección de la base de datos. Continuamos con los pasos que se muestran en la figura 5.2.3.

En la figura 5.2.3 llegaremos al menú de selección, elegimos Dispositivo y usamos el icono con los puntos suspensivos para buscar la base de datos, elegimos tipo de archivo y agregar, nos aparecerá un buscador para nuestro ordenador donde tendremos que ir a la dirección donde tengamos el .bak descargado y descomprimido, lo elegiremos, como se muestra en la figura 5.2.4. Seguimos los pasos como se muestra en las sucesivas imágenes.



5.2.3 Elegir la base de datos

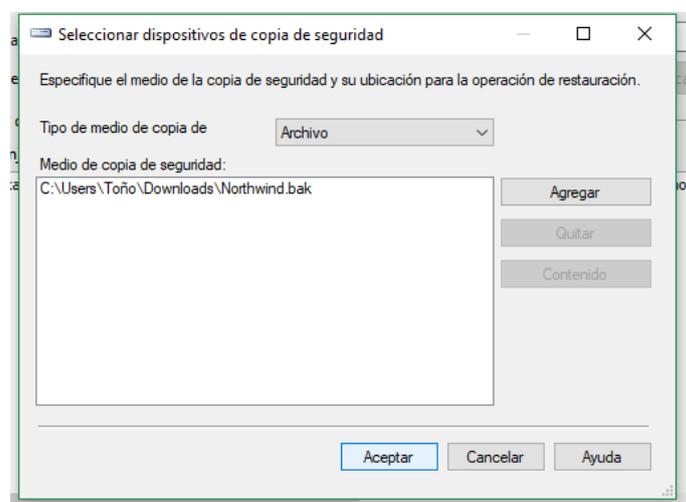


Figura 5.2.4 Elección del .bak

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

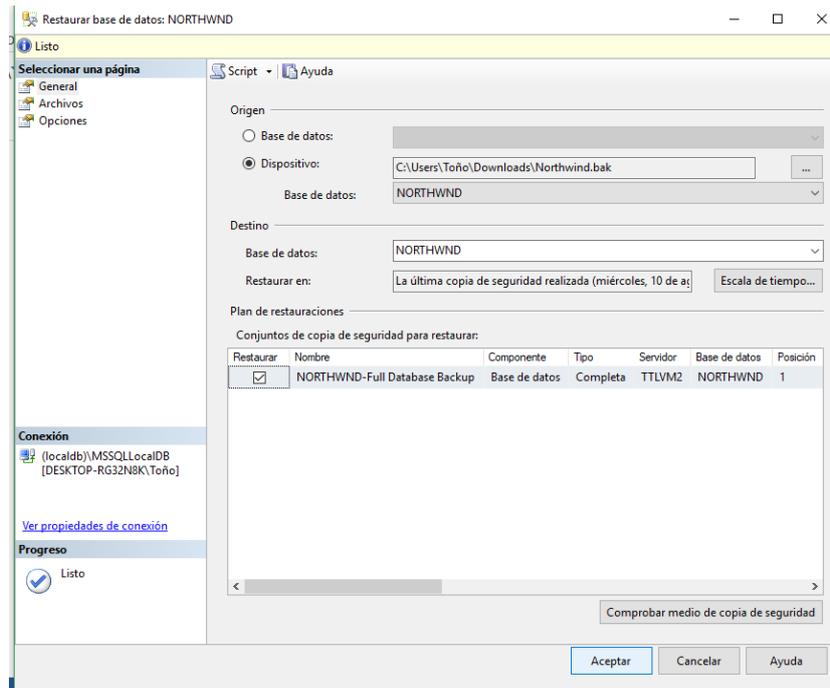


Figura 5.2.5 Elección de la base de datos

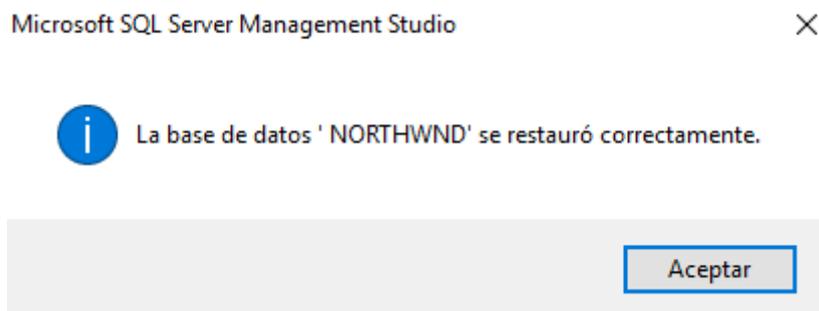


Figura 5.2.6 Éxito al restaurar base de datos

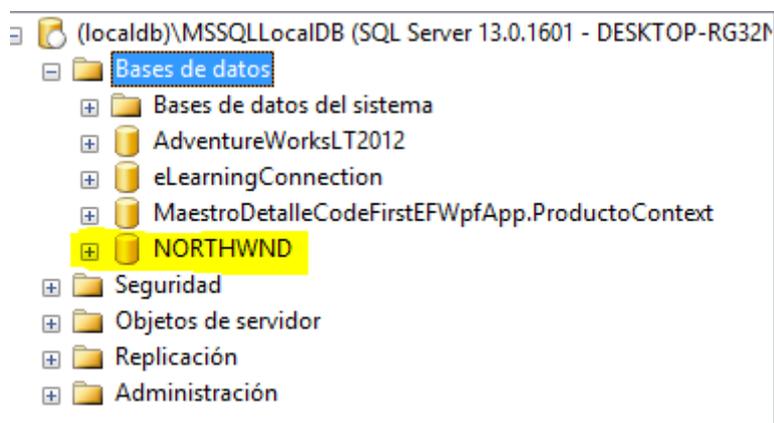


Figura 5.2.7 Comprobación de la nueva base de datos

Para comprobar si hemos instalado la base de datos correctamente, como vemos en la figura 5.2.8, podemos realizar una consulta simple para comprobar que nos ha devuelto información. Tras escribir la consulta, como vemos en la figura 5.2.9, en el recuadro, la ejecutaremos como vemos en la figura 5.2.10.

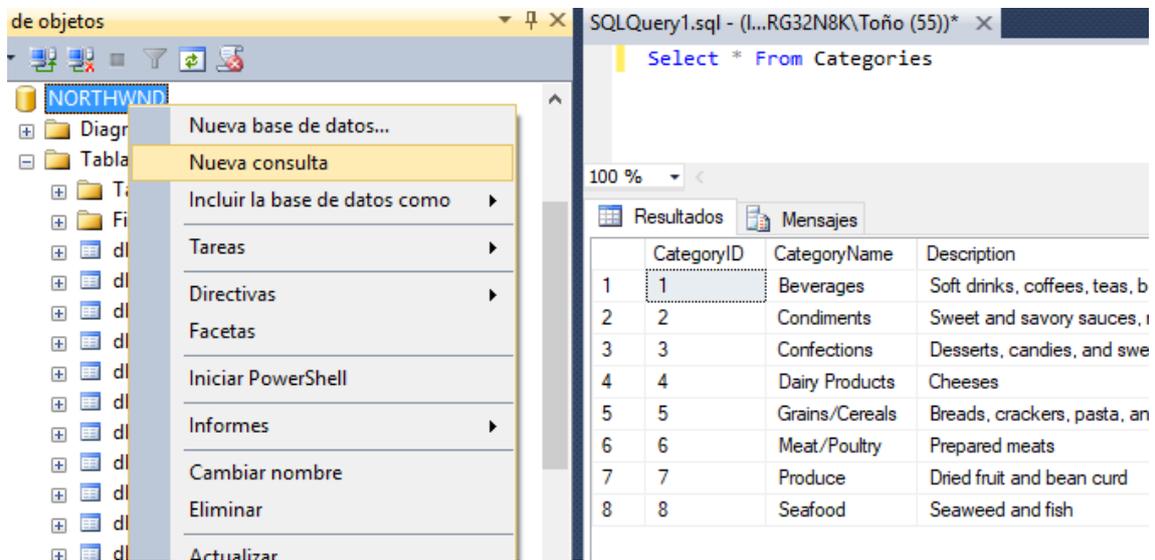


Figura 5.2.8 Comprobación de consultas



Figura 5.2.9 Consulta

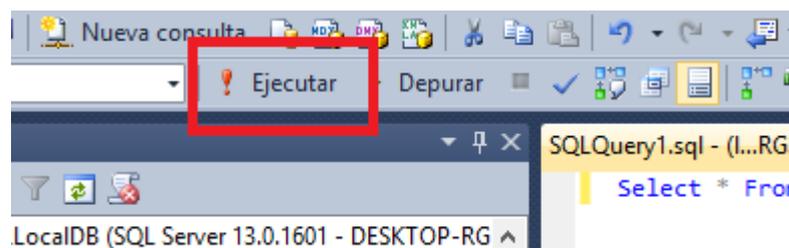


Figura 5.2.10 Botón de ejecución

Hemos obtenido la respuesta a la consulta, por tanto el funcionamiento es correcto y la tenemos correctamente instalada, podemos probar algunas consultas más si queremos una comprobación exhaustiva.

Antes de pasar al siguiente apartado recordemos varias cosas que vamos a tener que utilizar más adelante:

- Nombre de la base de datos: **NORTHWND**
- Nombre de la conexión: **(localdb)\MSSQLLocalDB**

## 5.3 Creación del proyecto en Visual Studio

El siguiente paso que vamos a realizar será crear un proyecto en Visual Studio, esto lo realizamos para poder crear nuestro propio servidor API REST, lo ejecutaremos y estaremos mostrando la información al exterior, a este proyecto le vamos a conectar la base de datos creada anteriormente y vamos a instalarle el framework necesario para poder ejecutar y crear sentencias en formato OData.

Vamos a crear un nuevo proyecto en Visual Studio, para este proyecto estamos haciendo uso de Visual Studio 2015.

Archivo -> Nuevo Proyecto

Instalado -> Plantillas -> Visual C# -> Web -> Aplicación web ASP.NET

Nombraremos nuestro proyecto como: Northwnd y lo ubicaremos.

Hay que tener en cuenta desmarcar la casilla de la derecha de Application Insights.

Como se muestra en la siguiente imagen **5.3.1**.

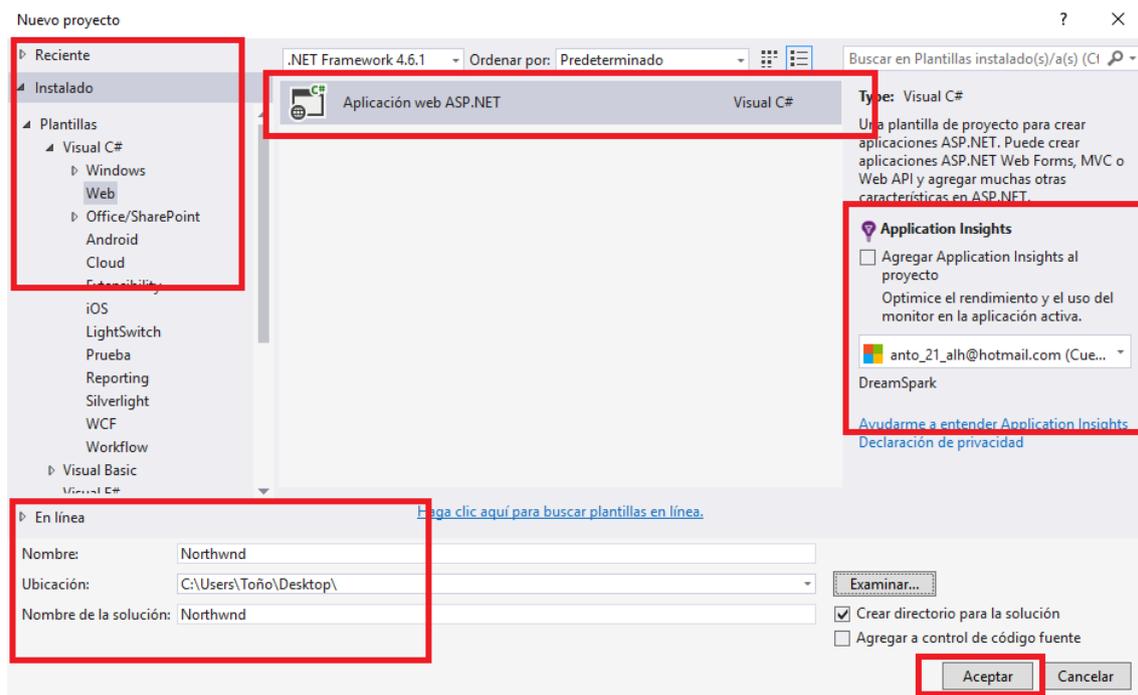


Figura 5.3.1 Creación del proyecto Visual Studio

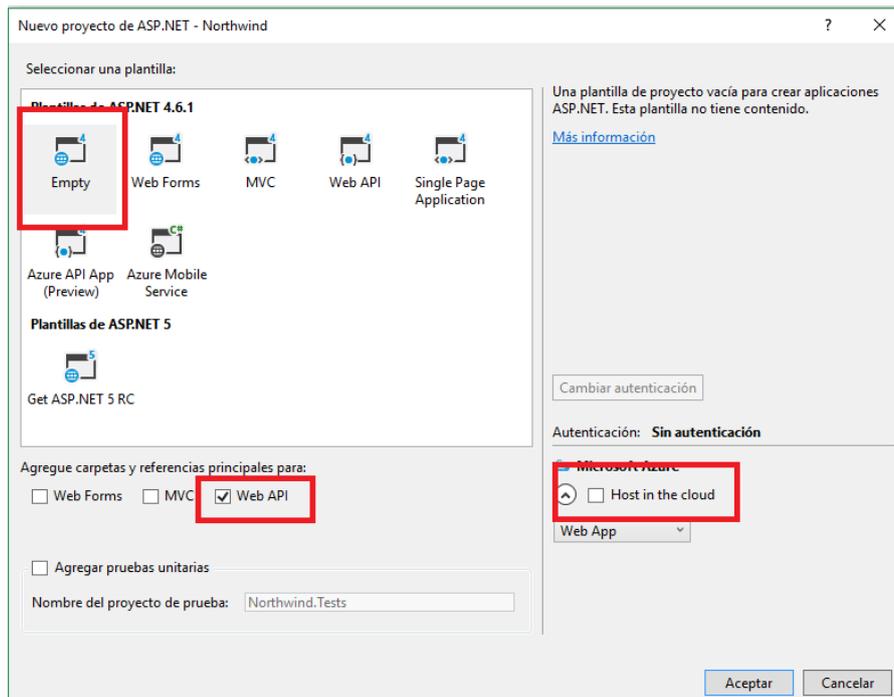


Figura 5.3.2 Elección del proyecto

## 5.4 Conexión del proyecto con la base de datos.

Con el proyecto de Visual Studio creado y SQL Server iniciado, como hicimos en el apartado 5.2 realizaremos la conexión entre estos dos proyectos.

Proyecto (Northwnd) Botón derecho -> Agregar -> Nuevo Elemento

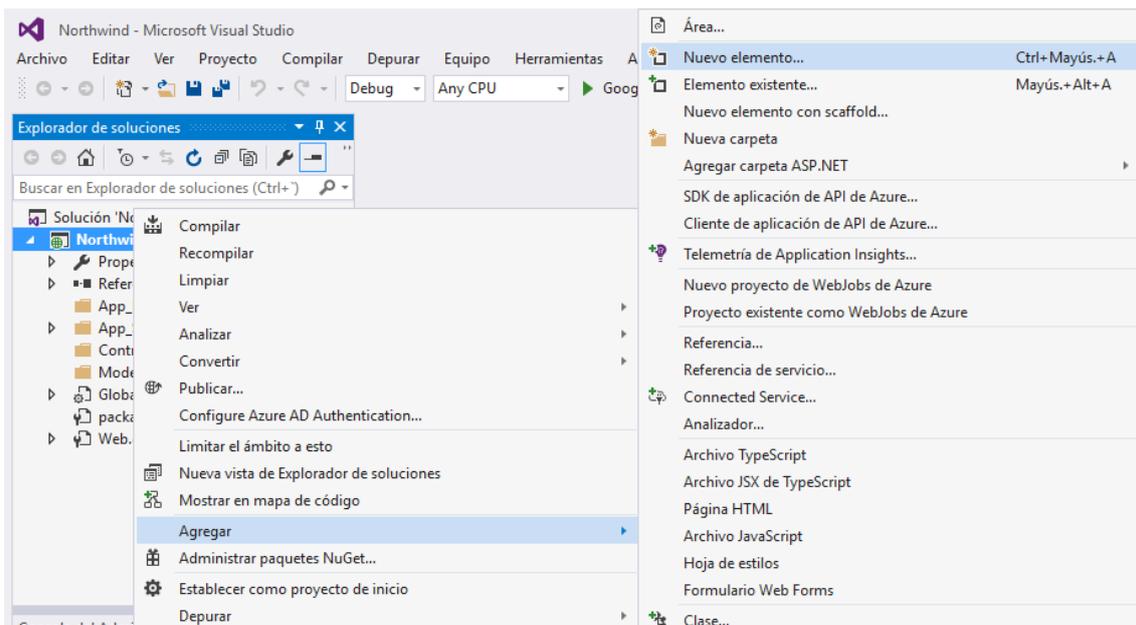


Figura 5.4.1 Agregación de la base de datos al proyecto.

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

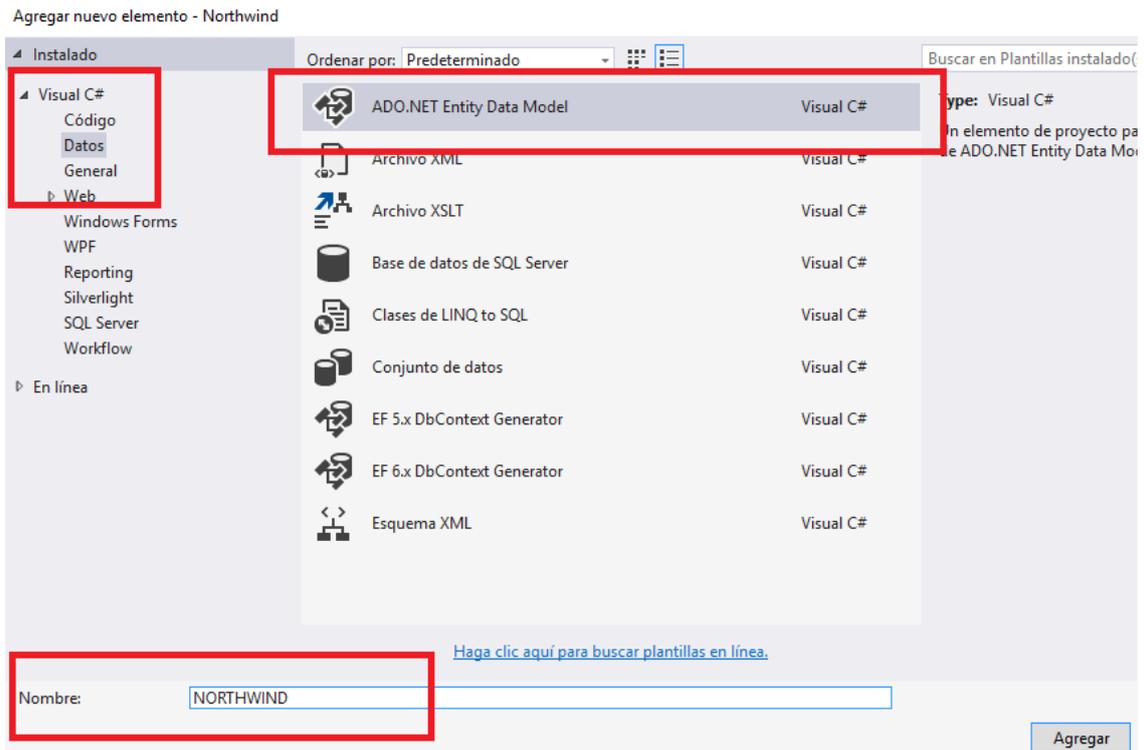


Figura 5.4.2 Elección del elemento



Figura 5.4.3 Elección del tipo de base de datos

A continuación seleccionamos una nueva conexión con la base de datos.

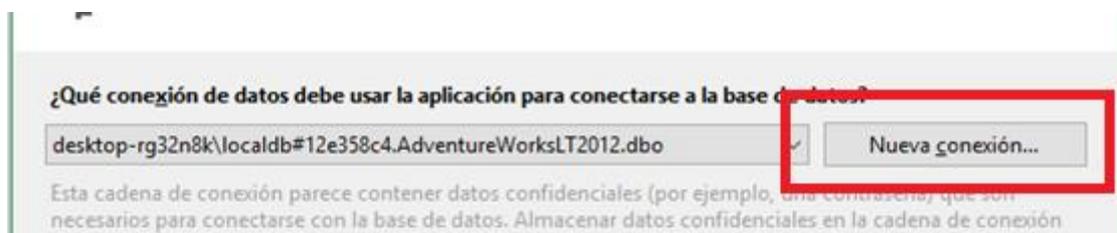


Figura 5.4.4 Elección de una nueva conexión

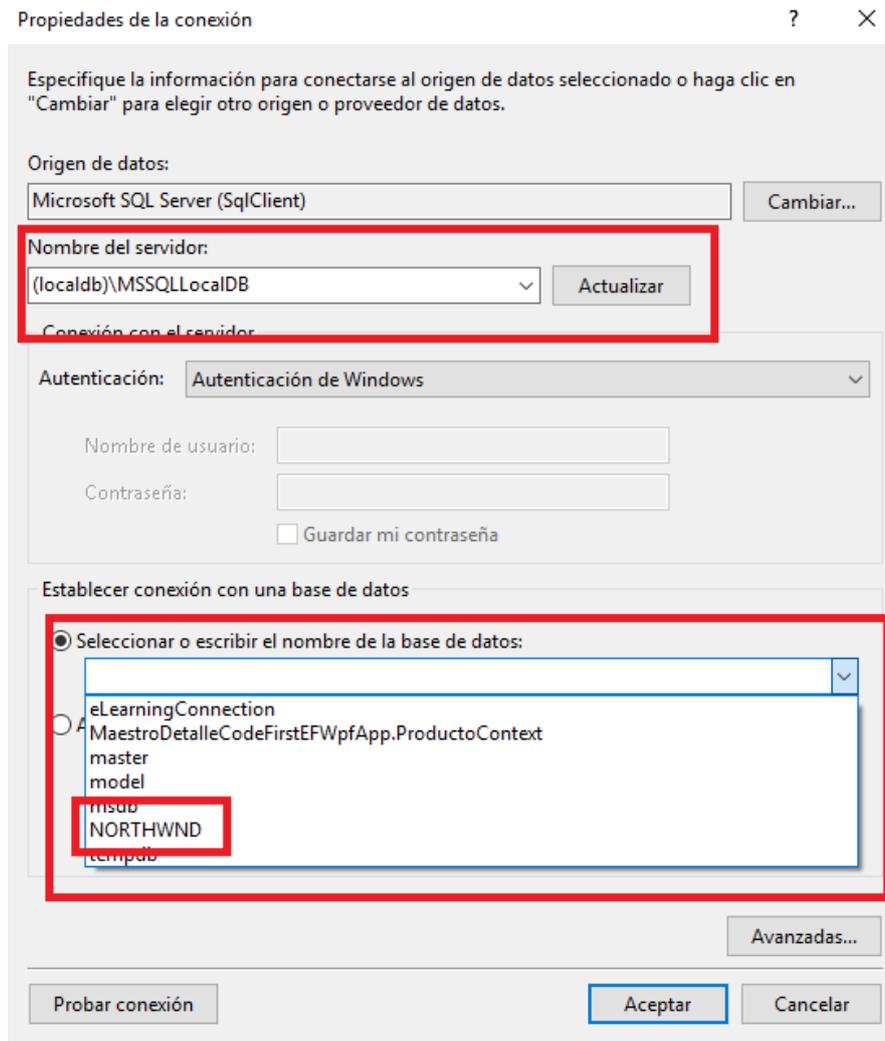


Figura 5.4.5 Estableciendo valores de conexión

Recordamos en este punto que la conexión es: **(localdb)\MSSQLLocalDB**

Y que la base de datos es: **NORTHWND**

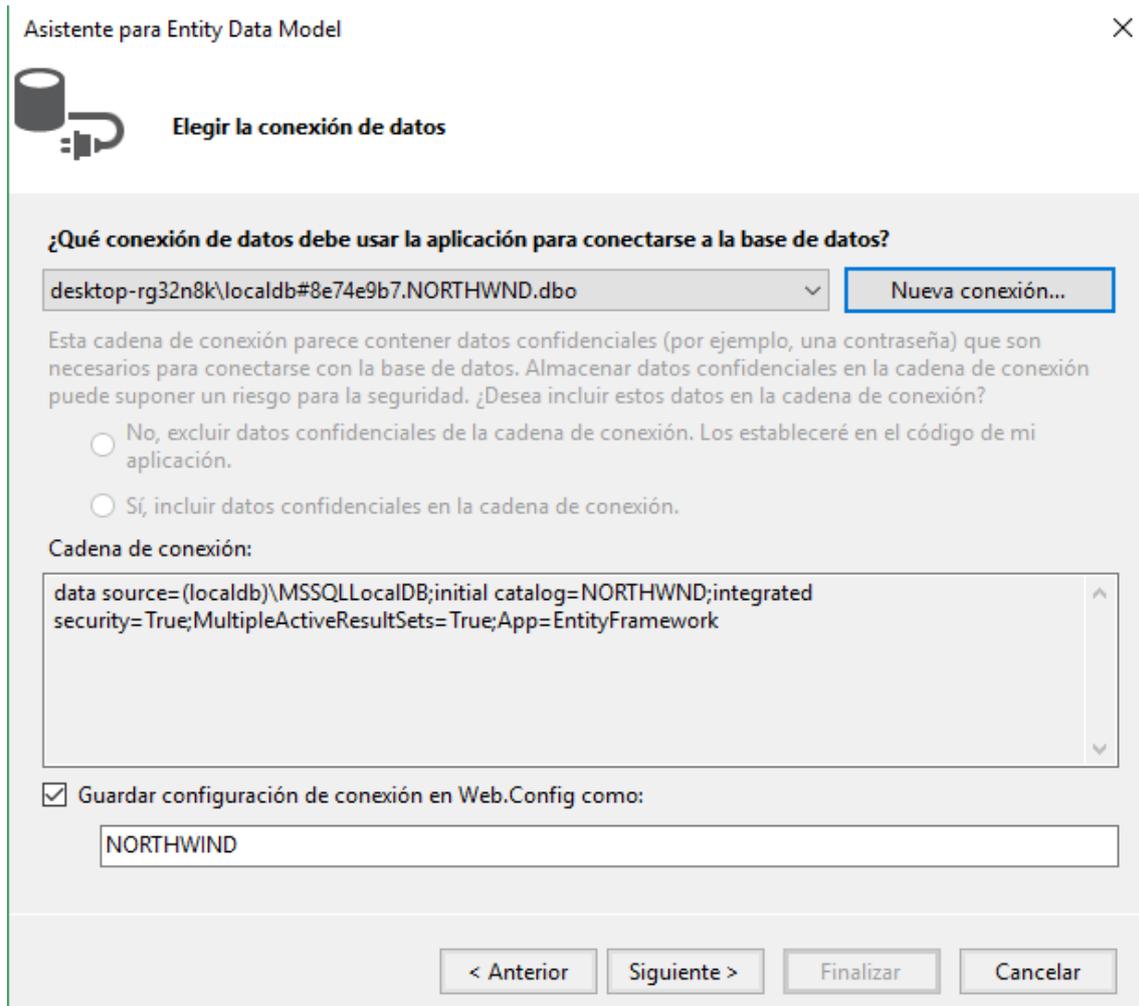


Figura 5.4.6 Nueva conexión

Tras pulsar el botón de Siguiete, el paso que sigue es importante, hemos establecido la conexión con la base de datos total, pero debemos de saber si vamos a querer darle toda la información a la persona que vaya a acceder a la API REST o no, para ello en el siguiente paso vamos a elegir qué tablas de la base de datos añadiremos al proyecto. Para este ejemplo nos bastaremos con las tablas: **Employees, Customers, Order Details, Orders**. Como vemos en la figura 5.4.7

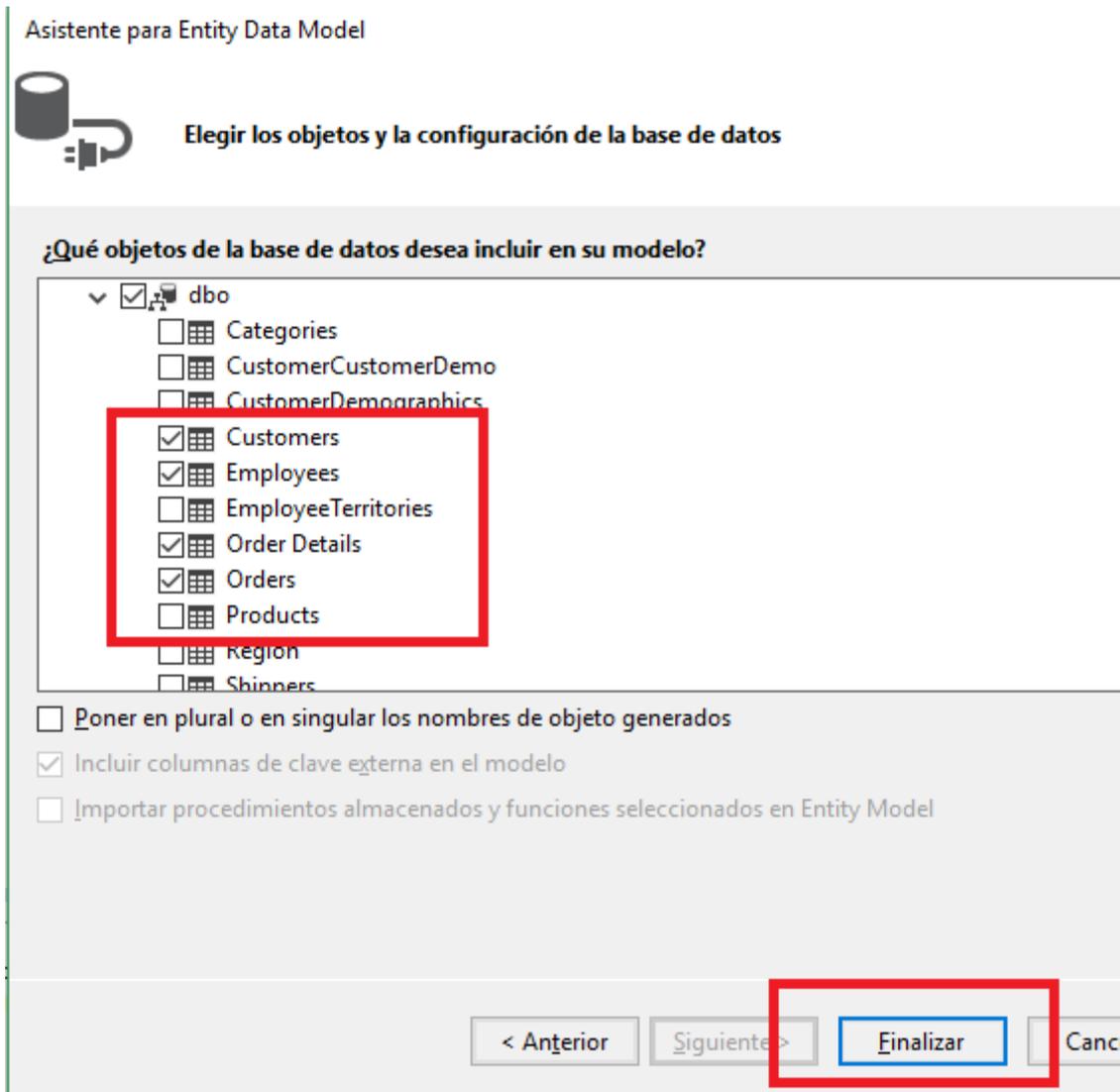


Figura 5.4.7 Selección de las tablas que queremos importar al proyecto

Es importante seleccionar correctamente las tablas que queremos proporcionar al proyecto, dado que a partir de este punto vamos a instalar el framework y tras lanzar la aplicación a modo de servidor cualquier persona tendrá acceso a las tablas que hayamos seleccionado a través de la Api rest.

## 5.5 Instalación del framework Restier desde NuGet

Este paso nos ayudará a instalar el framework RESTier del cual ya hablamos anteriormente, este se encargará de transformar los datos en formato OData y mostrarlos a través de la API Rest.

Proyecto (botón derecho) -> Administrar paquetes NuGet

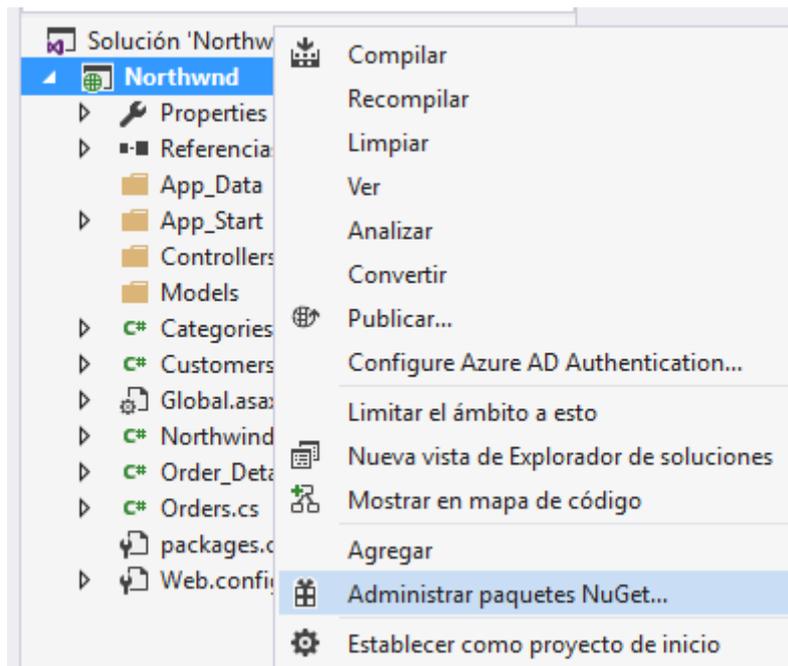


Figura 5.5.1 Administración de paquetes NuGet

En la pestaña que nos aparece tenemos que seleccionar **Examinar** para buscar nuevos paquetes, vamos a buscar el paquete **Microsoft.Restier**. Debemos tener activa la casilla de **Incluir versión preliminar**. Tenemos que seleccionar el primer paquete, que como vemos ya nos da información sobre su objetivo, instalamos ese paquete con la opción **Instalar** situada a la derecha del panel.

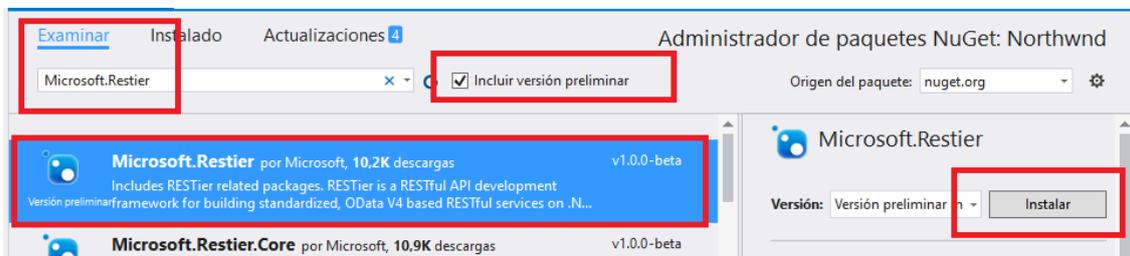
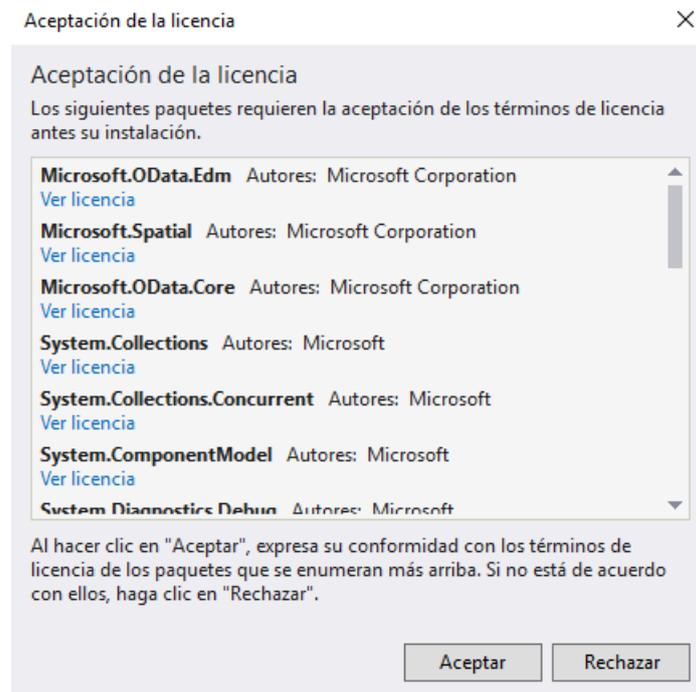


Figura 5.5.2 Selección del paquete Restier

Deberemos de aceptar el mensaje de licencias que nos aparecerá. El mensaje será similar al que encontramos en la figura 5.5.3.



*Figura 5.5.3 Aceptación de las licencias*

Tras aceptar las licencias comenzará la instalación del paquete, esto puede tardar bastante. Obtendremos un mensaje en el panel inferior con Finalizado si todo el proceso ha tenido éxito.

## 5.6 Modificaciones pertinentes y ejecución del servicio.

Aunque actualmente tenemos todo instalado debemos de modificar una serie de archivos para que el proyecto se enlace con la base de datos y sea capaz de ejecutar el programa en la dirección que nosotros le indiquemos, esto será utilizado para saber dónde mostraremos nuestra API Rest.

Llegados a este punto podemos cerrar SQL Server dado que las tablas ya las tenemos en el proyecto de Visual Studio. A continuación vamos a modificar el archivo WebApiConfig.cs

Proyecto Northwind -> App\_Start -> WebApiConfig.cs (doble click)

Una vez abierto este archivo sustituimos el contenido del archivo por el siguiente, mostrado en el código **5.6.1**.

```

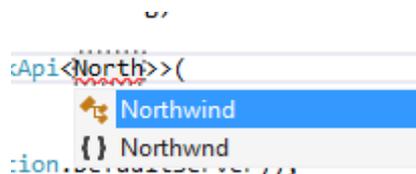
using System.Web.Http;
using Microsoft.Restier.Providers.EntityFramework;
using Microsoft.Restier.Publishers.OData;
using Microsoft.Restier.Publishers.OData.Batch;

namespace Northwnd
{
    public static class WebApiConfig
    {
        public async static void Register(HttpConfiguration config)
        {
            await config.MapRestierRoute<EntityFrameworkApi<NORTHWND>>(
                "NORTHWND",
                "api/NORTHWND",
                new RestierBatchHandler(GlobalConfiguration.DefaultServer));
        }
    }
}

```

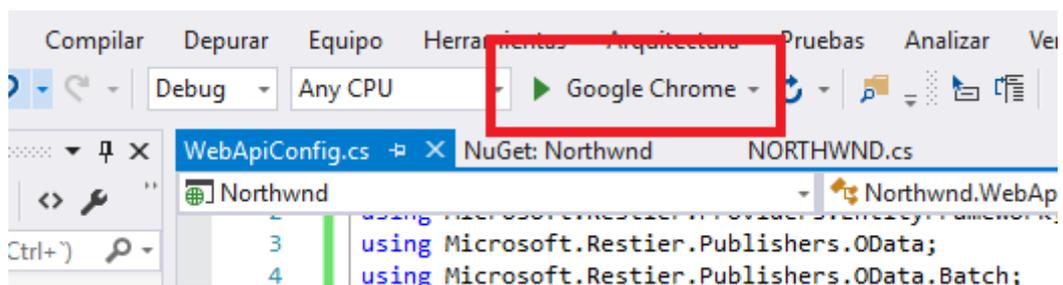
*Código 5.6.1 Modificación para el archivo Web*

El código básicamente hace uso de las librerías instaladas por el framework Restier. Se establece cuáles son los parámetros de la API REST. En el caso de que la palabra **NORTHWND** esté subrayada en rojo puede deberse a que hayamos puesto el nombre del modelo o de la clase de otra forma, en cualquier caso solo debemos borrar la palabra **NORTHWND** y comenzar a escribir **North**, el corrector nos sugerirá cual es el nombre que buscamos como vemos en la figura 5.6.1



*Figura 5.6.1 Selección de la clase*

Las letras en rojo indican el nombre de la base de datos, y el siguiente nombre en rojo indica donde deberemos de acceder para obtener la primera consulta REST API. Vamos a ejecutar el proyecto si no tenemos ningún fallo de compilación.



*Figura 5.6.2 Ejecución del proyecto*

Tras su ejecución se nos abrirá el buscador por defecto, en mi caso Google Chrome, en él aparecerá la URL: localhost:<puerto al azar>

Si dejamos que termine de cargar nos dará un fallo de dirección puesto que esa no es la dirección que hemos establecido para usar la API REST, por tanto modificaremos la dirección añadiendo a la URL el nombre que dimos en el **código 5.6.1**. Para mi caso:

**localhost:<puerto asignado>/api/NORTHWND**

Como vemos en la figura 5.6.3

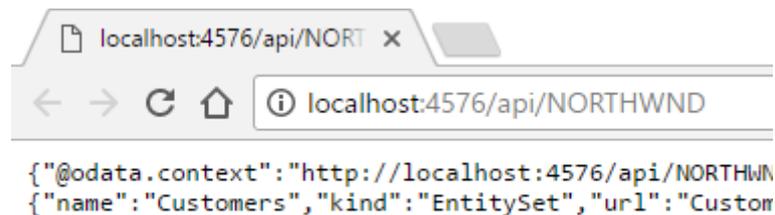


Figura 5.6.3 comprobación de la URL

Comprobamos que se ha creado la API REST y vemos que nos está devolviendo un tipo de dato en formato OData. Es curioso cómo podemos navegar fácilmente desde Chrome por esta API Rest, podemos observar en la información devuelta que nos está dando un **“name”** : **“Customers”** y más palabras clave que tienen que ver con las tablas que hemos elegido para nuestro proyecto, si queremos seguir navegando por la API vamos a modificar la URL, como por ejemplo:

**localhost:<puerto>/api/NORTHWND/Customers**

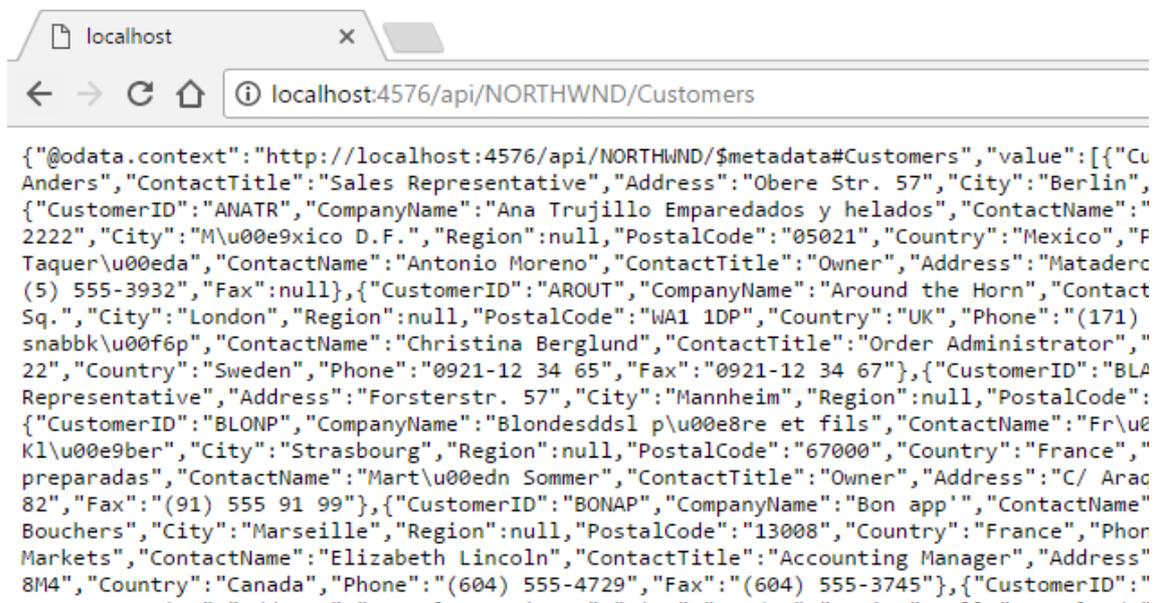


Figura 5.6.4

Lo que hemos hecho en realidad es lanzar una consulta a la base de datos donde le estamos pidiendo que nos devuelva toda la tabla Customers.

## 5.7 Consumo de la fuente mediante Power BI.

Ya que tenemos todo creado actualmente estamos ofreciendo un servicio API Rest desde el proyecto que creamos anteriormente, en este momento vamos a utilizar la herramienta Power BI para obtener información y poder comenzar a tratarla y analizarla.

Para este paso tenemos que tener abierto y funcionando el proyecto de Visual Studio y haber comprobado que estábamos recibiendo datos desde Google Chrome.

Vamos a comenzar creando un nuevo proyecto en Power BI con una guía paso a paso dado que estamos algo familiarizados con el entorno como vimos en el **Capítulo 2**.

Iniciamos Power BI Desktop, la aplicación puede ser descargada desde la página oficial pero para ello debemos de tener una cuenta de correo de una entidad asociada a Power BI, en nuestro caso la universidad tiene esta asociación y hemos podido crear la cuenta y descargar el programa gracias al correo de inlumine.

PowerBI inicia un proyecto vacío sin nombre, vamos a comenzar por añadirle la fuente desde la que partimos como vemos en la figura **5.7.1**

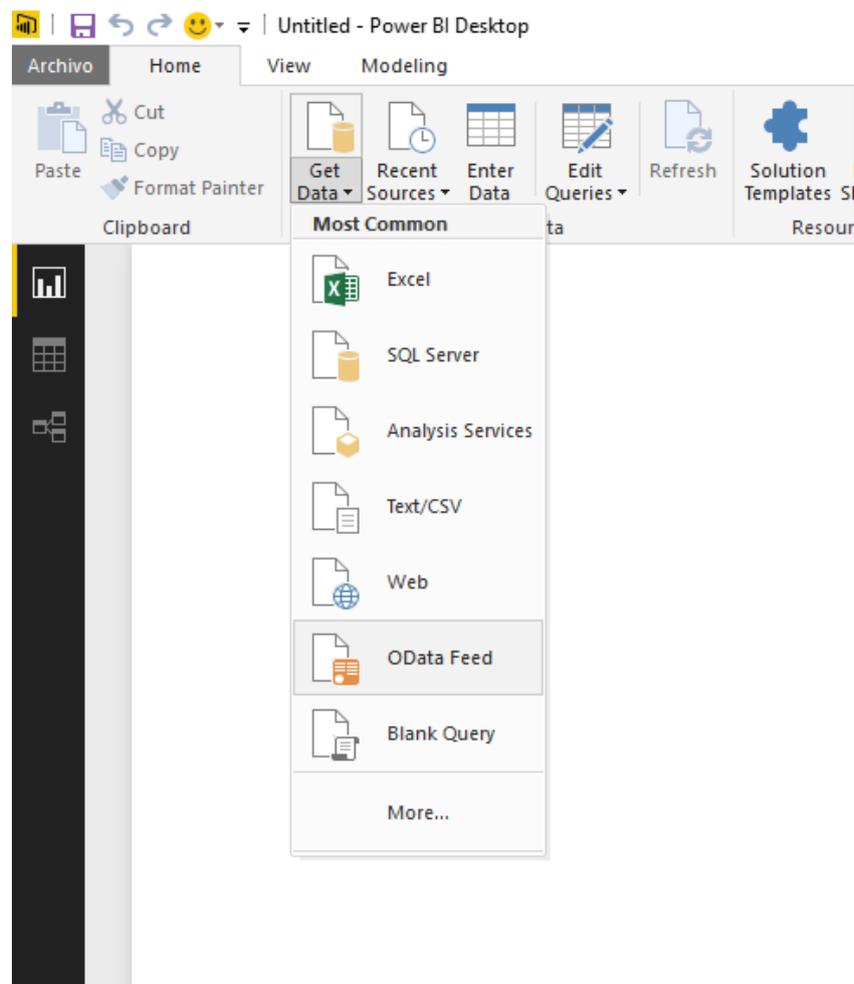


Figura 5.7.1 Añadir la fuente OData al proyecto Power BI

La ventana que aparece nos va a solicitar una URL, esta no es otra que la que hemos estado probando en Google Chrome, la principal, como vemos en la figura 5.7.2.

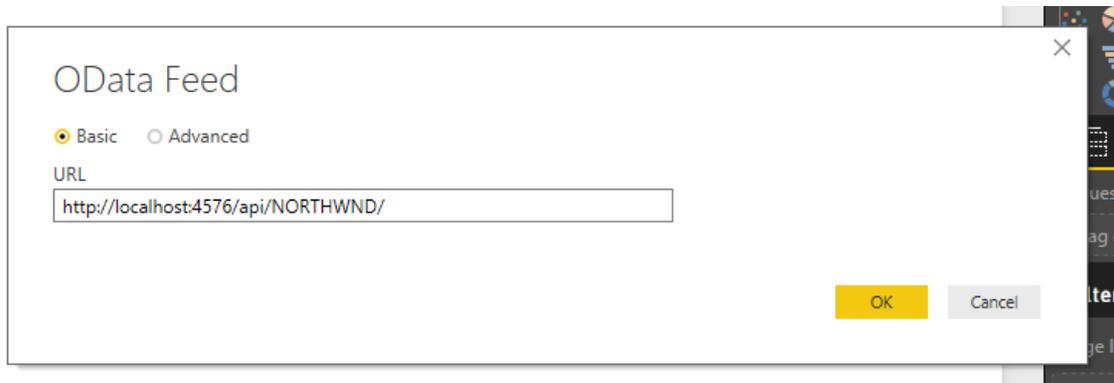


Figura 5.7.2 Selección de la URL

Tras seleccionar la URL nos va a pedir que seleccionemos las tablas que queremos añadir al proyecto. En realidad, Power BI lo que ha hecho es obtener la información de la primera consulta a la URL que le hemos dado, ha sacado los nombres de las tablas y los ha añadido a la URL original, creando así una nueva URL a la que lanzar una consulta de manera recursiva.

Vamos a seleccionar todas las tablas debido a que hicimos un corte de información cuando creamos el proyecto en Visual Studio.

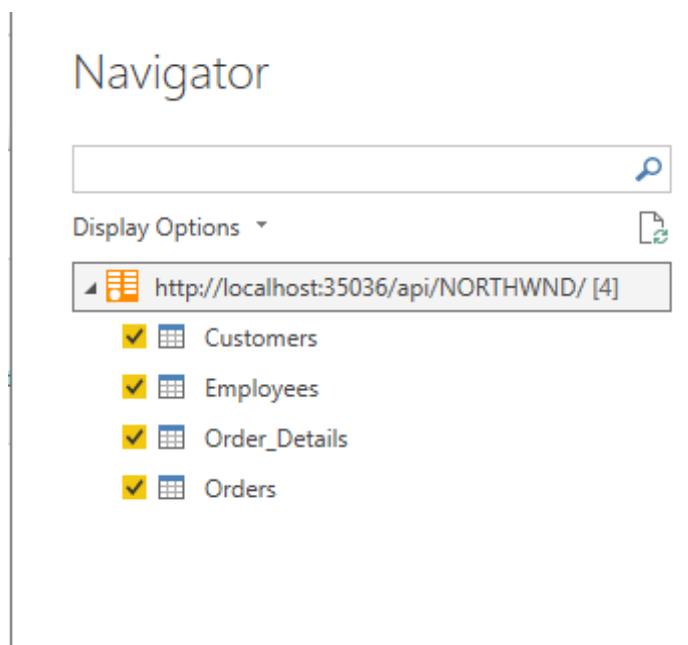


Figura 5.7.3 Selección de las tablas para Power BI

Una vez que Power BI haya terminado de cargar e importar las tablas a su proyecto, podemos parar el proceso de Visual Studio y cerrar el programa para liberar memoria en el sistema.

## 5.8 Conclusiones finales.

Ahora tenemos todos los datos guardados dentro de Power BI, desde este momento podemos hacer estudios de inteligencia de negocio, como ya hablábamos en el **capítulo 1**, a partir de ahora podremos reunir información que nos será útil a la hora de hacer una toma de decisiones.

Dado que hemos importado la base de datos Northwnd y hemos seleccionado las tablas de clientes y empleados hemos decidido realizar un estudio. Para ello vamos a comenzar por ver cómo están relacionadas las tablas que hemos importado, vamos a utilizar una opción de Power BI que muestra las relaciones con las tablas importadas, tercera pestaña del panel de navegación de la izquierda, como se muestra en la figura **5.8.1**.

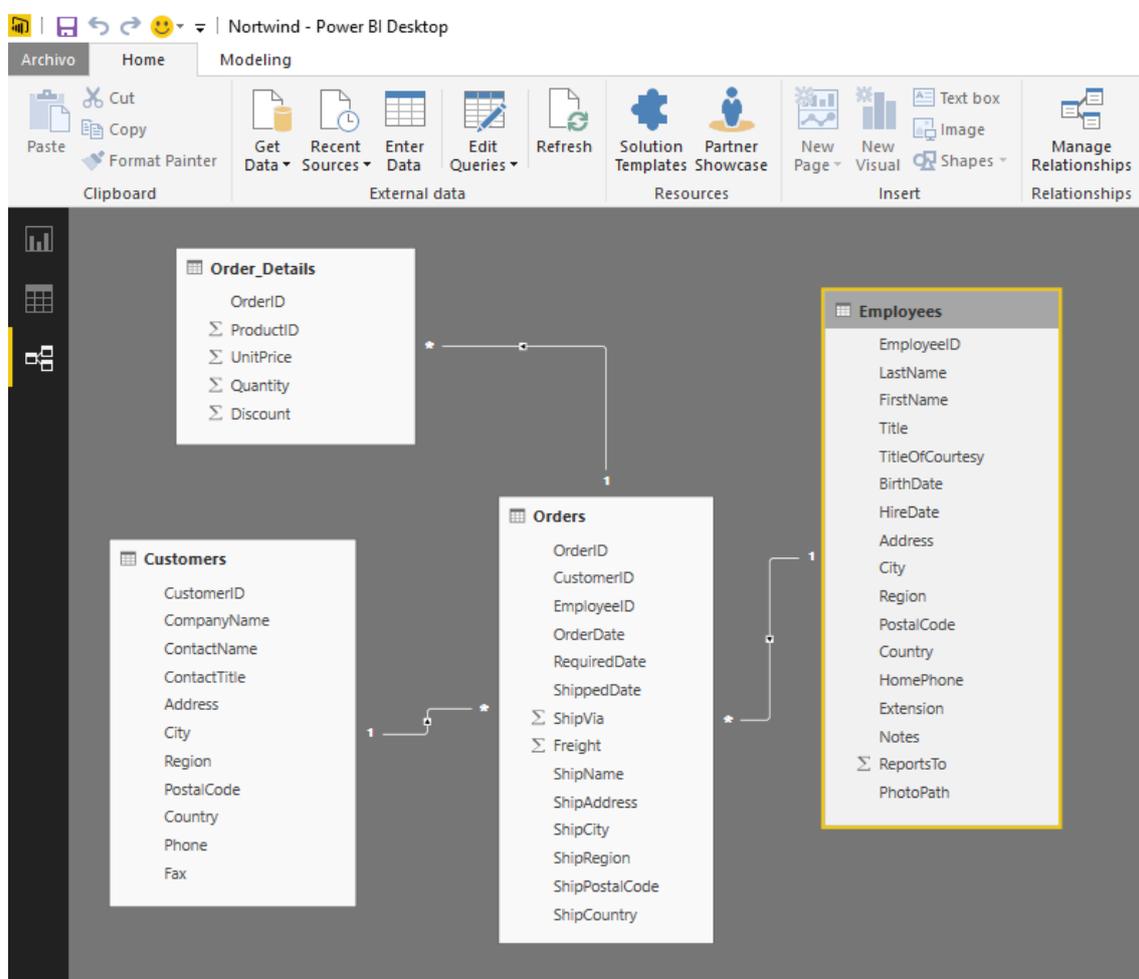


Figura 5.8.1 Relaciones entre las tablas

A continuación en la segunda pestaña del mismo panel podemos observar las tablas que hemos importados como se muestra en la figura 5.8.2.

The screenshot shows the Power BI interface with a data table and the Fields pane. The table displays customer information, and the Fields pane shows the available fields for the 'Customers' table.

CustomerID	CompanyName	ContactName	ContactTitle
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner
AROUT	Around the Horn	Thomas Hardy	Sales Representative
BERGS	Berglunds snabbköp	Christina Berglund	Owner
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative
BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Representative
BOLID	Bóldo Comidas preparadas	Martín Sommer	Owner
BONAP	Bon app'	Laurence Leblan	Owner
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Account Representative
BSBEV	B's Beverages	Victoria Ashworth	Sales Representative
CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Representative
CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Representative
CHOPS	Chop-suey Chinese	Yang Wang	Owner
COMMI	Comércio Mineiro	Pedro Afonso	Sales Representative
CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative
DRACD	Drachenblut Delikatessen	Sven Ottlieb	Owner
DUMON	Du monde entier	Janine Labrune	Owner
EASTC	Eastern Connection	Ann Devon	Sales Representative
ERNSH	Ernst Handel	Roland Mendel	Sales Representative

The Fields pane on the right shows the following fields for the 'Customers' table:

- Address
- City
- CompanyName
- ContactName
- ContactTitle
- Country
- CustomerID
- Fax
- Phone
- PostalCode
- Region

Figura 5.8.2 Muestra de una de las tablas

Supongamos que queremos invertir en la fidelidad de nuestros compradores, para ello queremos que las personas que más pedidos hagan y que más dinero estén haciéndonos ganar se vean recompensados con bonos de fidelidad. Para ello vamos a realizar las siguientes configuraciones en Power BI y nos vamos a ayudar con las siguientes gráficas.

Para ello vamos a usar las herramientas que ofrece Power BI en la primera pestaña, vamos a hacer uso de los datos que aportan las tablas para poder desarrollar graficas e informes como se muestra en la figura 5.8.3, una visión general que detallamos a continuación. Encontramos, a la izquierda una muestra de todos los usuarios que tiene el sistema con su ID de usuario. La gráfica superior muestra los usuarios mostrados por número de compras, abajo izquierda los usuarios que han realizado más de 15 compras y a la derecha los usuarios que más dinero han gastado.

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

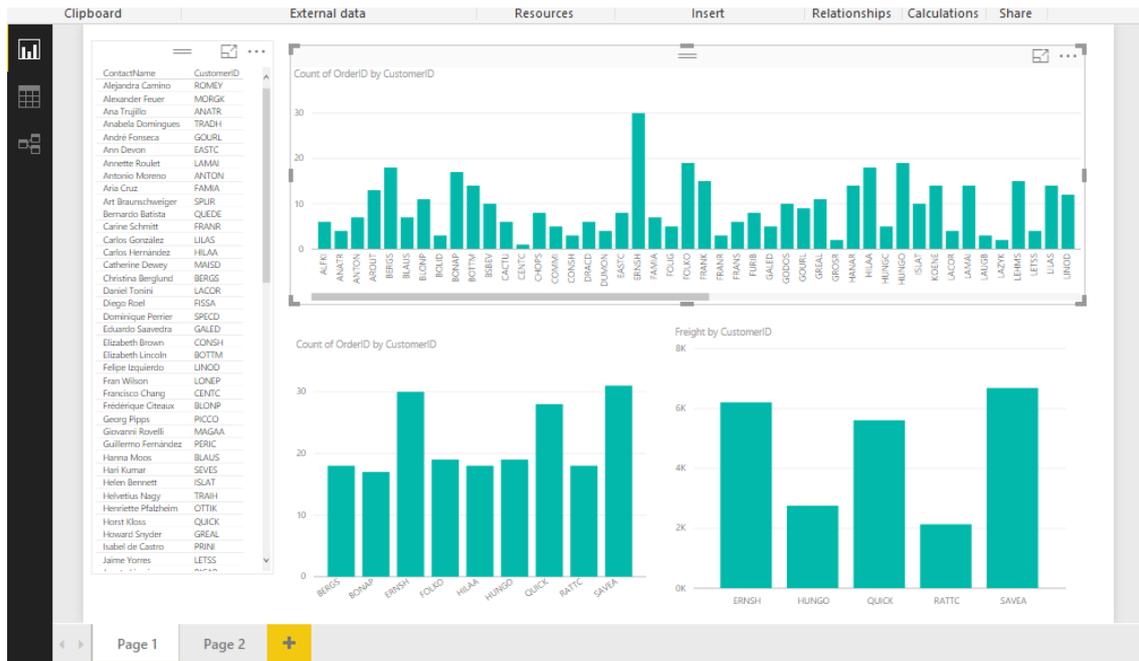


Figura 5.8.3 Visión general de Power BI Desktop

En la figura 5.8.4 encontramos a todos los usuarios registrados en el sistema y el número de pedidos que ha realizado cada uno. En la figura 5.8.5 vemos como se ha hecho para conseguir este resultado.

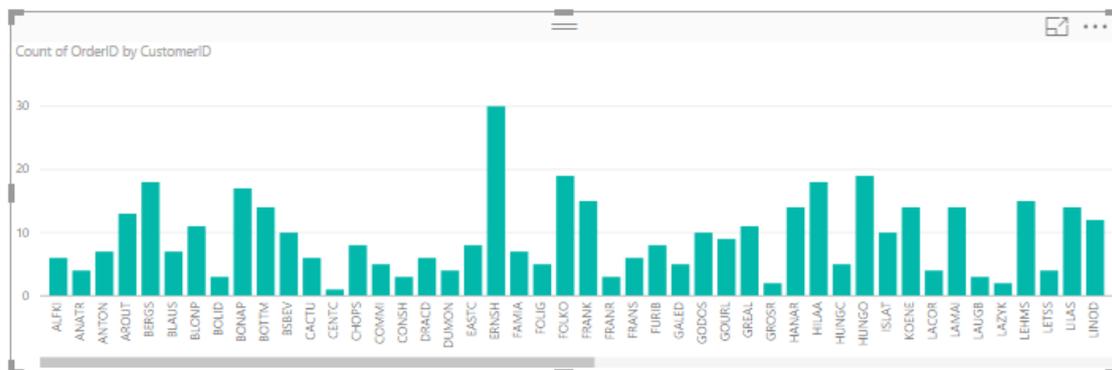


Figura 5.8.4 Total de pedidos por usuario

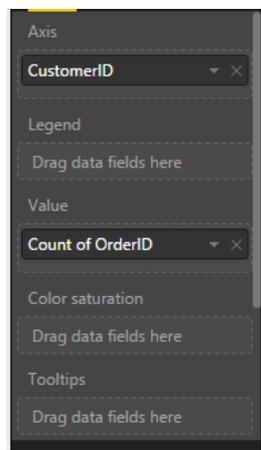


Figura 5.8.5 Total de pedidos por usuario, código

Esta información puede sernos útil, pero vamos a reducir el número de usuarios que han realizado pedidos, vamos a aplicar un filtro para mostrar los que más pedidos han realizado. En este caso vamos a mostrar todos los usuarios que hayan hecho más de quince pedidos, como se muestra en la figura 5.8.6. Se acompaña el código en la figura 5.8.7. Solo con esto podríamos decir que hay tres candidatos bastante claros que realizan una cantidad importante de pedidos. Por otro lado hemos decidido hacer un buen estudio del mercado, por tanto no vamos a conformarnos solo con una gráfica, vamos a realizar un estudio adicional.

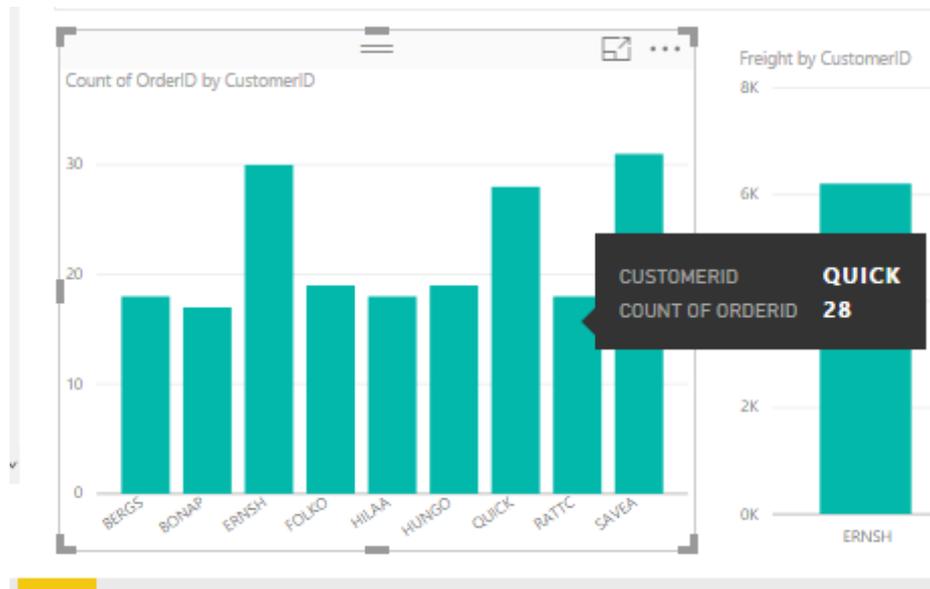


Figura 5.8.6 Mejores consumidores

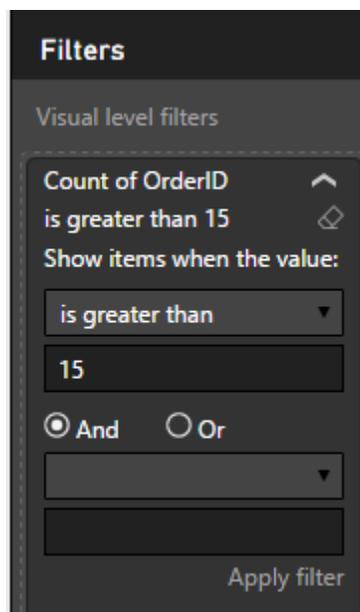


Figura 5.8.7 Código de mejores consumidores

No solo vamos a saber quién ha hecho el mayor número de pedidos, esta información por si sola puede ser engañosa, realizar cuatrocientos pedidos por un valor de un euro no supone una ventaja para la empresa, por tanto vamos a acompañar esta información con otra adicional, vamos a mostrar quienes han sido los usuarios que más dinero se han gastado. Mostrado en la figura 5.8.8.

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

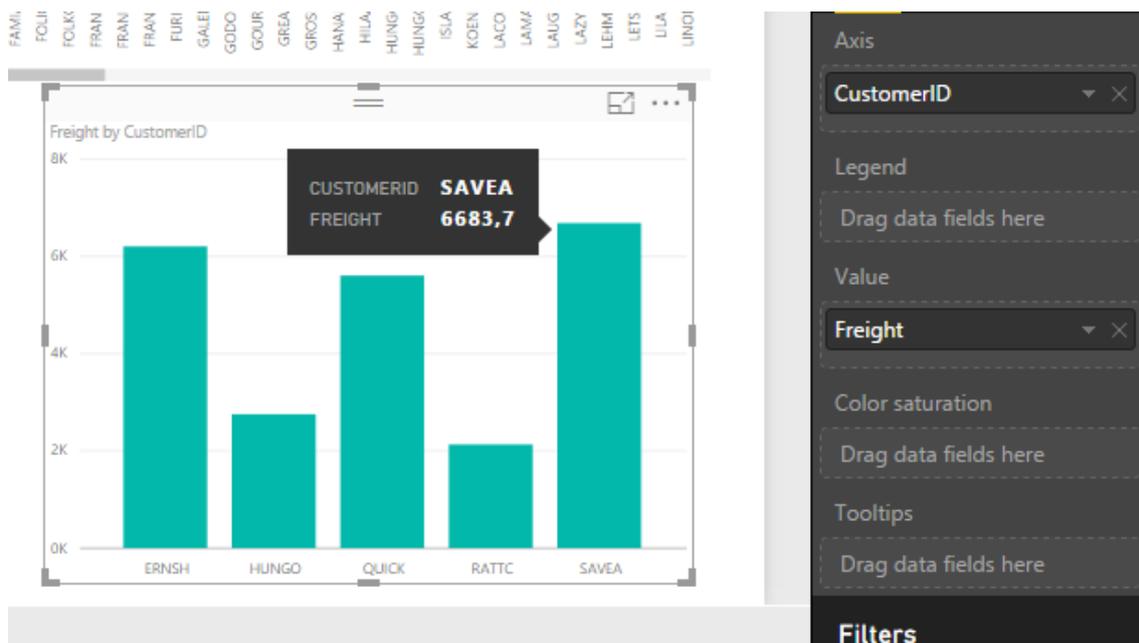


Figura 5.8.8 Total de pedidos por usuario

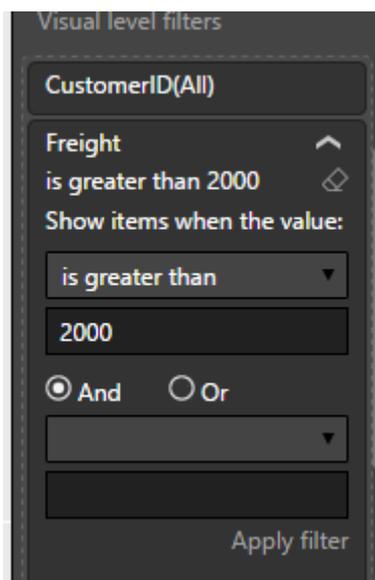


Figura 5.8.9 Total de pedidos por usuario, código

Comparando esta figura con la figura 5.8.6 nos damos cuenta de que hay 3 usuarios que siguen destacando, se mantienen los tres en ambas figuras, sin embargo en esta última figura 5.8.8 vemos como aparecen dos nombres nuevos que podemos mostrar también en el número máximo de pedidos.

Gracias a estas dos gráficas podemos estar seguros de que estos cinco últimos usuarios, mostrados en la figura 5.8.8 son los mejores consumidores y ellos deberían de conseguir el beneficio de clientes de fidelidad.

Como este mismo podemos hacer varios para sentirnos seguros en la toma de decisiones. Podemos ampliar los rangos en el filtro para aumentar el número de clientes. Power BI permite la selección de usuarios independientes para mostrarlos en varias gráficas al mismo tiempo.



Figura 5.8.10 Muestra de usuario en varias gráficas

Vamos a desarrollar otro estudio, en este caso queremos montar una nueva sede de nuestra empresa, posiblemente uno de los factores a investigar sea en que países estamos vendiendo más, para ello vamos a realizar unas gráficas donde se observará el mapamundi distribuido con los países donde estamos vendiendo, este mapa se va a ver reducido cuando apliquemos unos filtros para mostrar las mayores ventas por país.

Desarrollado en la página 2 y 3 del proyecto se muestra de qué ciudades son los usuarios que han realizado los pedidos, como se muestra en la figura 5.8.11 y como se ha obtenido este mapa en la figura 5.8.12.

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

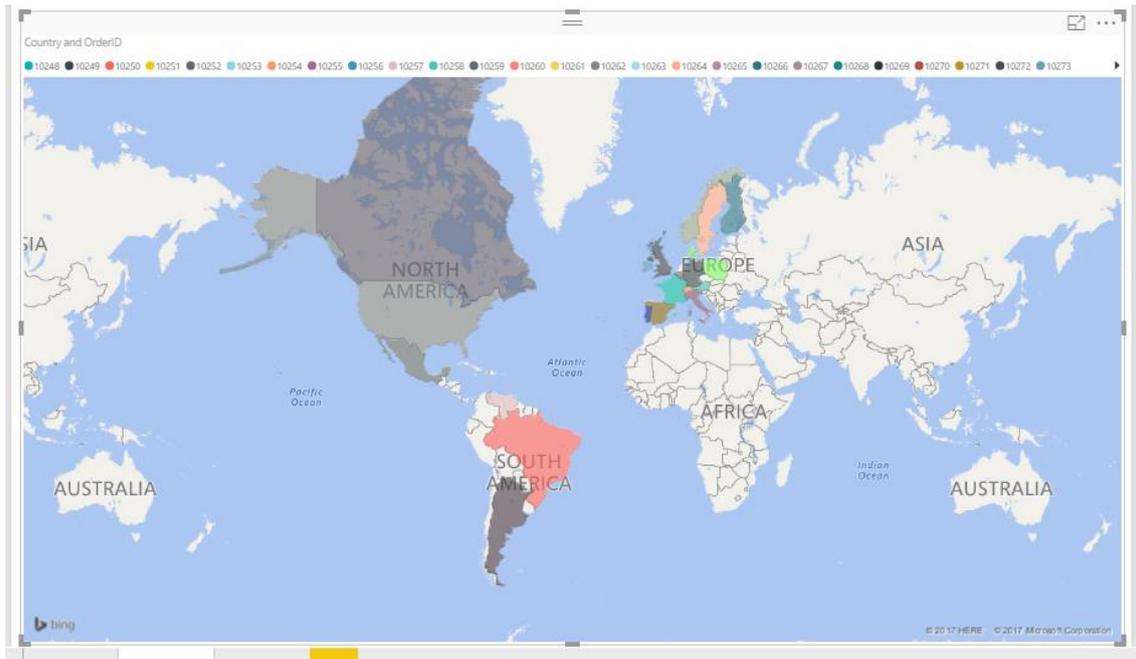


Figura 5.8.11 Mapa de la localización de los usuarios

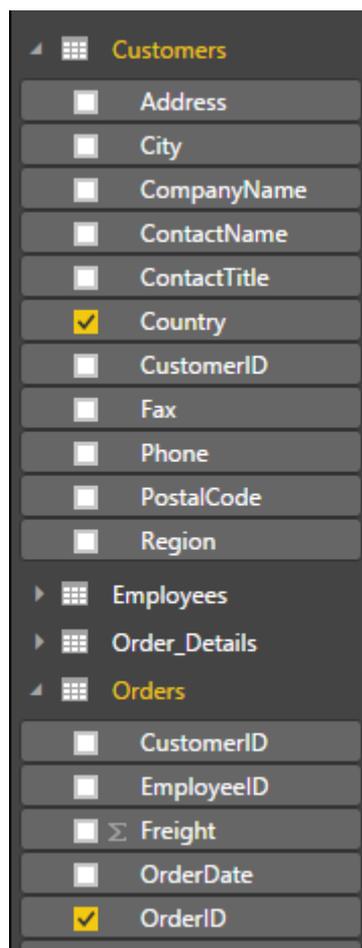


Figura 5.8.12 Mapa de la localización de los usuarios, código

Ahora vamos a realizar un filtro para mostrar solo los mayores consumidores, los que más pedidos realizan.



Figura 5.8.13 Mapa de la localización de los usuarios que más compran

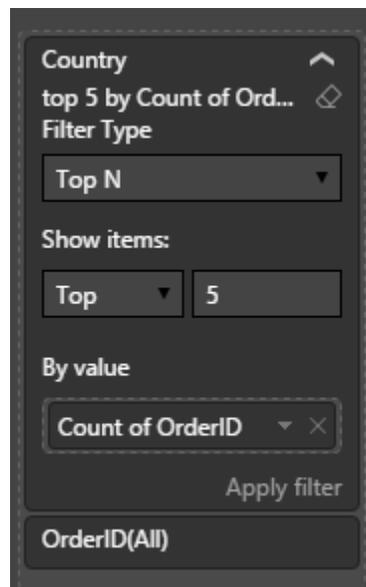


Figura 5.8.14 Mapa de la localización de los usuarios que más compran, código

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

Con estos dos ejemplos podríamos tener realizado un estudio bastante sencillo. Una vez realizado nos interesa compartirlo con el resto de la empresa, o con el resto de interesados. Para ello vamos a realizar una publicación de nuestros resultados con la herramienta Power BI Desktop, para ello vamos a utilizar una opción que tiene Power BI Desktop y vamos a ver el resultado en Mobile y Cloud.

Comenzaremos por publicar los informes desarrollados en la herramienta Desktop, para ello utilizaremos el botón que se encuentra en el panel superior como se muestra en la figura 5.8.15

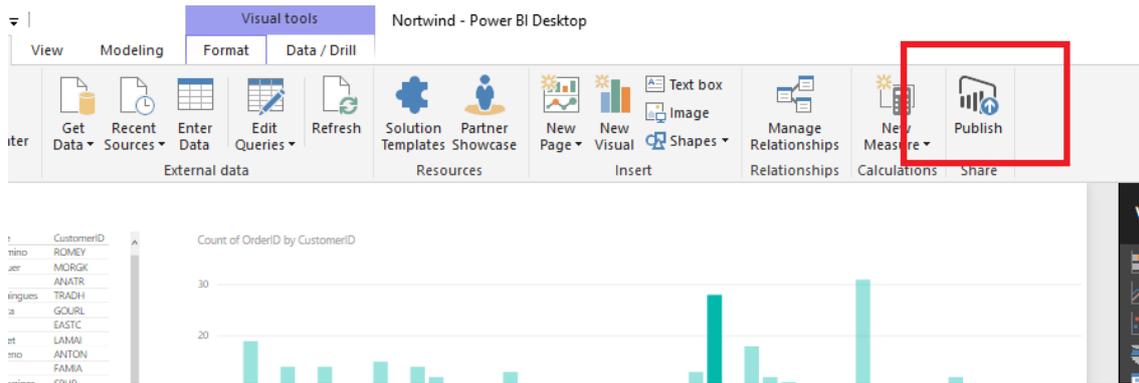


Figura 5.8.15 Publicar informes

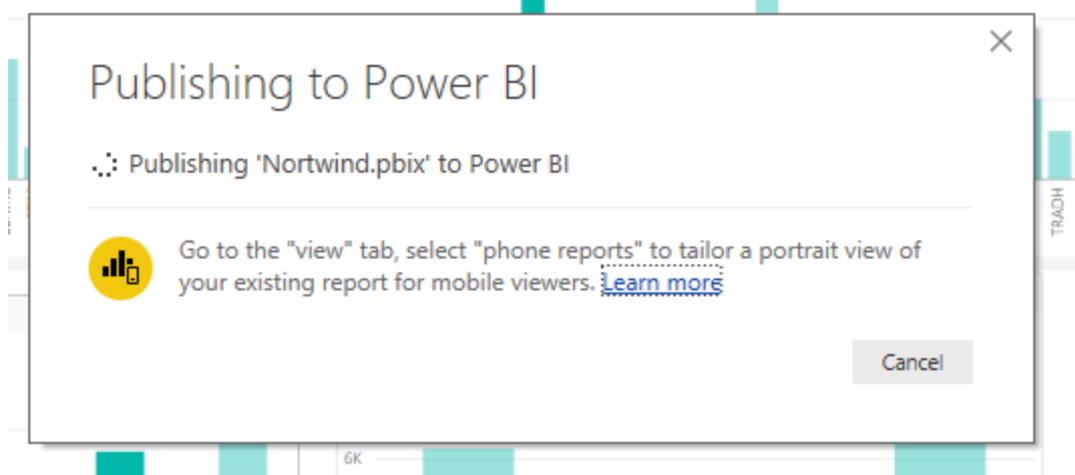


Figura 5.8.16 Publicando informes

Se podrán observar desde Power BI Cloud (Figura 5.8.17) y Power BI Mobile (Figura 5.8.18).

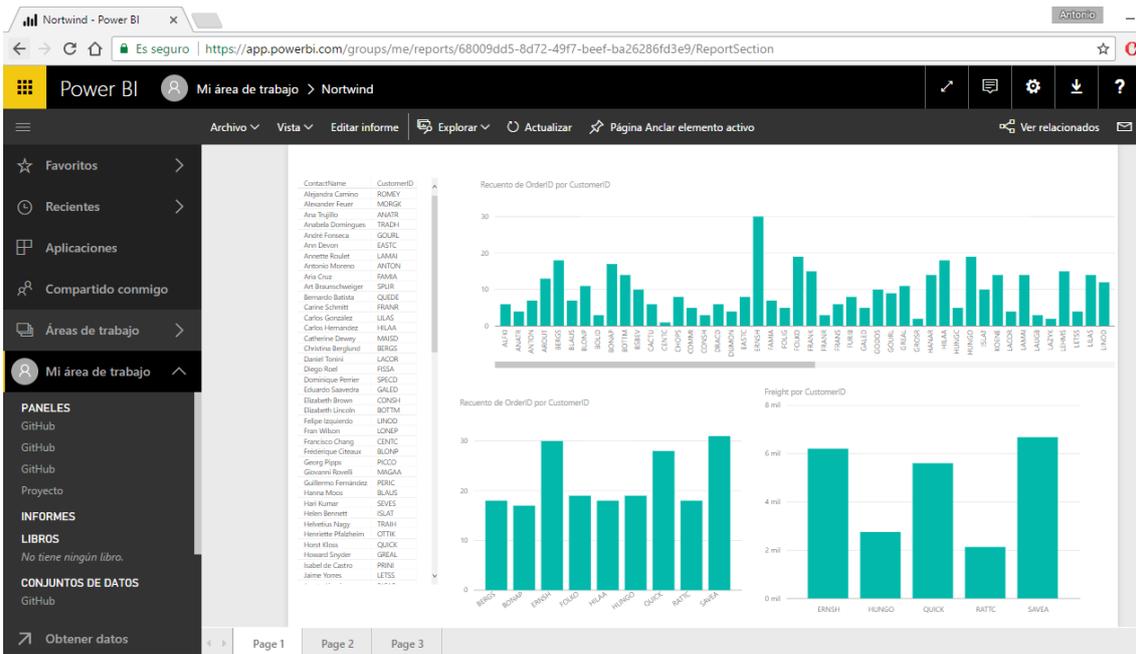


Figura 5.8.17 Mostrando informes en Power BI Cloud

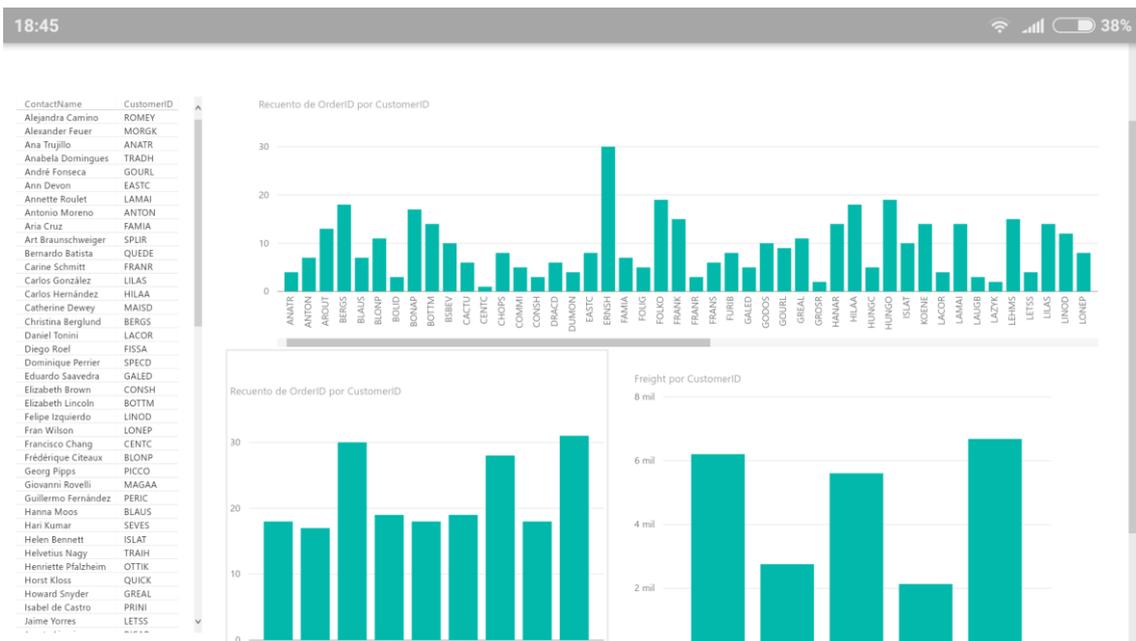


Figura 5.8.18 Publicando informes en Power BI Mobile

La publicación es rápida y sencilla y puede tenernos a todos informados al instante. En este momento tenemos que hacer un poco de concienciación en la empresa, con tres movimientos rápidos y estructuras que hemos hecho en Desktop hemos obtenido una información muy valiosa a la hora de tomar decisiones, pero no solo esto.

Comentábamos en el primer capítulo de la inteligencia de negocio las ventajas que podíamos obtener, una de ellos, como hemos visto hace un momento, puede ser la toma de decisiones. Por otro lado podemos ayudar a tener una buena consistencia en la base de datos, a continuación se muestra otra base de datos distinta donde se está haciendo el mismo estudio que acabamos de hacer, deseamos montar una empresa y queremos saber dónde estamos vendiendo más, al realizar el mapa encontramos que cuando mostramos por ciudades nos sale una buena distribución en el mapa, pero cuando a esas ciudad añadimos el código postal para saber la zona exacta donde viven comprobamos que hay errores, esto es debido a que los usuarios han rellenado la casilla ciudad correctamente pero introdujeron un numero postal al azar, por ello como podemos observar en la figura 5.8.19, hay una inconcluencia a la hora de mostrar los datos.

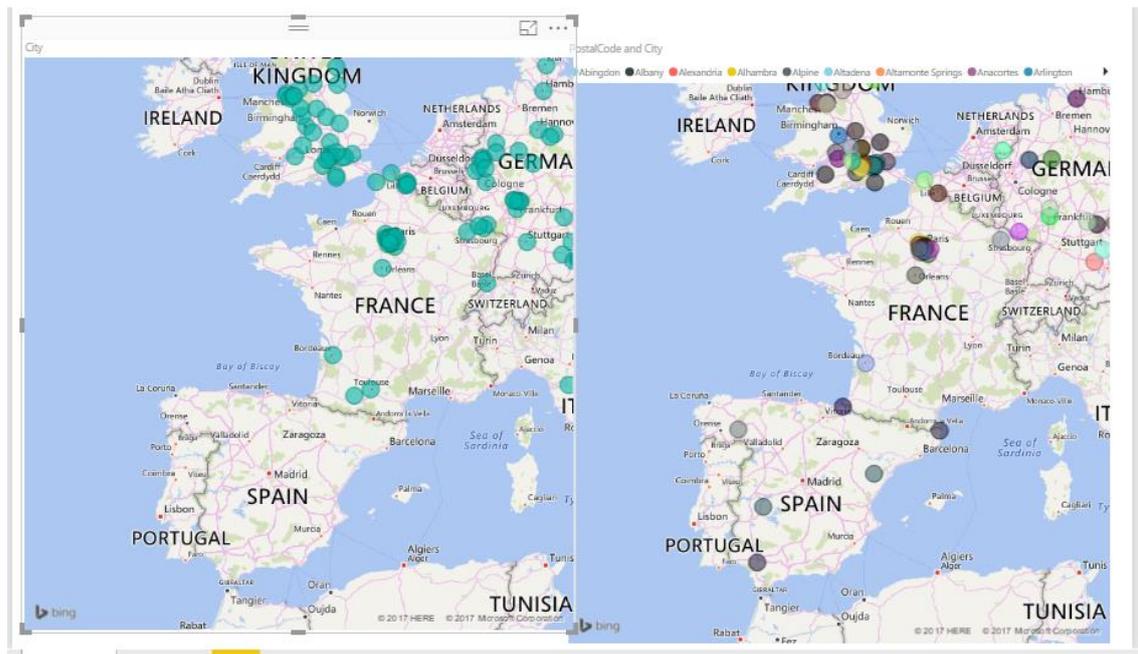


Figura 5.8.19 Inconcluencia en el mapa

Debemos darnos cuenta de lo importante que resulta realizar tanto un buen estudio, como una buena recopilación de datos. Esta inteligencia de negocio puede ser aplicado a cualquier tipo de empresa, ya sean grandes o pequeñas, siempre podemos aprender más sobre nuestro negocio y mejorarlo.

## Resumen

En este capítulo se ha desarrollado una guía paso a paso para crear una fuente de datos OData y se acompaña de la inserción de esos datos en un proyecto Power BI además de un pequeño ejemplo de cómo puede ayudar la inteligencia de negocio a una empresa.

# Capítulo 6

## Conclusión y Trabajo Futuro

---

### Contenido

- 6.1 Conclusión sobre el proyecto.
- 6.2 Cronograma del proyecto.
- 6.3 Trabajo futuro.

### 6.1 Conclusión sobre el proyecto.

Tras haber realizado todo el proyecto podemos destacar varios factores importantes. Por un lado nos sorprende que haya a nuestra disposición tan pocas fuentes OData, es una pena no haber encontrado más para poder hacer uso de ellas con el objetivo de aprender.

Por otro lado la experiencia de usar Power BI ha sido máxima, se han realizado varios proyectos de pruebas y estudios aunque se ha decidido usar ejemplos sencillos para poder introducir a Power BI. Ha sido muy gratificante haber podido hacer uso de esta herramienta gracias a poseer una cuenta en inlumine por la Universidad de Almería.

Se ha aprendido a usar las API REST, se comenzó por aprender a usar Phalcon para comprender su funcionamiento, tras ello se exploró Postman para realizar consultas a API REST y a continuación se realizaron consultas para fuentes OData.

Se ha mostrado por encima la funcionalidad de Power BI, pero sobre todo de la inteligencia de negocio, debemos insistir en aplicarla en las empresas, sin importar el tamaño de la misma, debemos apostar por la mejora informática, de cómo la inteligencia de negocio puede ayudarnos a la toma de decisiones, a una buena apariencia al público y a una buena recopilación de los datos.

Por todo lo anterior aquí nombrado denominamos nuestro proyecto un punto de partida para introducir al lector a interesarse por la inteligencia de negocio y proporcionarle un punto de comienzo con OData y Power BI.

## 6.2 Cronograma.

Se han presentado dificultades durante el proceso de creación del proyecto, por ello se ha visto modificado el cronograma que se desarrolló durante el anteproyecto de la manera que se muestra a continuación en la figura 6.2.1.

	Meses de desarrollo							
	Semana de cada mes							
	Marzo				Abril			
	1	2	3	4	1	2	3	4
Estudio de BI.								
Estudio de OData								
Configuración y uso de PowerBI								
Seguimiento del curso: Analyzing and Visualizing Data with Power BI								
Ejemplo de uso de fuentes OData con PowerBI								
Desarrollo de la Memoria								
Creación de presentación final del proyecto								

Figura 6.2.1 Mes Marzo y Abril Cronograma

En los dos primeros meses no se presentó nada fuera de lo esperado, se estudió BI y OData como conceptos mínimos para poder aprender sobre el proyecto que realizaríamos. Se configuró y se usó Power BI para comprender su funcionamiento. Se comenzó a realizar un curso online para aprender cómo funcionaba Power BI y sus mejores características, se comenzaron a realizar ejemplos con fuentes de datos OData en Power BI.

Aunque en los primeros meses no se modificó el cronograma, llegado mayo se incluyó un importante y gran bloque nuevo acompañado de modificaciones en los estudios y creaciones como se observa en la figura 6.2.2.

	Mes de desarrollo										
	Semana del mes										
	Mayo				Junio				Julio		
	1	2	3	4	1	2	3	4	1	2	
Estudio de BI.											
Estudio de OData											
Configuración y uso de PowerBI											
Seguimiento del curso: Analyzing and Visualizing Data with Power BI											
Ejemplo de uso de fuentes OData con PowerBI											
Creación de fuente de API REST para fuente Odata											
Desarrollo de la Memoria											
Creación de presentación final del proyecto											

Figura 6.2.2 Mayo, Junio y Julio en el cronograma

Se dejó de lado el curso para comenzar con los ejemplos de uso de fuentes OData en Power BI. Llegado a este punto donde usábamos fuentes OData con Power BI se decide que es buena idea desarrollar nuestra propia fuente de datos dado que solo conseguimos conectar con una fuente de datos en formato V4 para OData. Para realizar esto volvimos a investigar sobre OData y comenzamos con la creación de la fuente y el desarrollo de la memoria, donde realizamos más investigación en BI para poder aportarla a nuestro proyecto paso a paso.

Hemos marcado en rojo en la figura 6.2.2 las zonas que se había designado de tiempo a unas actividades que fueron sustituidas por la creación de la fuente OData y para el estudio de BI y OData.

### 6.3 Trabajo futuro.

Tras haber finalizado el proyecto y habernos encontrado con algunas dificultades en el camino, propongo como trabajo futuro los siguientes aspectos. Dado que nuestro proyecto se ha dividido en importancia en dos bloques igualmente importantes OData y Power BI se proponen dos proyectos para centrarse en estos temas de manera individual.

**6.3.1 Desarrollo de framework propio de OData,** crear un framework propio para crear fuentes OData.

**6.3.2 Creación de un proyecto real para Power BI,** acceder a los datos de una empresa y realizar un estudio de inteligencia de negocio con Power BI.

### **6.3.1 Desarrollo de framework propio de OData.**

En este proyecto se ha hecho uso del framework RESTier para la creación de una API REST que devolviera los datos en formato OData. Se propone crear un framework con una funcionalidad similar. Esto es debido a que cuando se estuvo buscando información sobre los framework la mayoría han dado problemas de versiones, y aunque OData ofrece una buena cantidad no todos están adaptados para la última versión. Parece un tema interesante dado que OData está creciendo en importancia, si sabemos crear el framework para V4 podremos desarrollar un framework para V5 si existiera y podríamos estar preparados.

Los framework que muestra OData tienen una documentación básica y la mayoría está en inglés, pudiendo desarrollar un framework completamente en español para adaptarlo a empresas españolas que estén interesados en hacer uso de uno de ellos y no posean conocimientos del inglés.

### **6.3.2 Creación de un proyecto real para Power BI.**

En este proyecto se hacen ejemplos ficticios para empresas ficticias. Se propone crear un proyecto real, con una empresa real a la que realizarle una consultoría de inteligencia de negocio y abordar el problema con Power BI.

Se debe de buscar una empresa que esté dispuesta a dar la suficiente información como para poder realizar un buen análisis de Power BI, para ello habría que realizar un gran estudio sobre Power BI y sobre inteligencia de negocio, pero no será necesario exponer los datos en formato OData, simplemente acceder a la fuente que nos ofrece la propia empresa.

Del mismo modo que hemos desarrollado dos ejemplos en el caso de crear bonos de fidelidad y de crear una nueva sede, deberían de desarrollarse estudios enfocados a los objetivos de negocio de la empresa.

# Bibliografía

---

[1] Buscador Google (Marzo 2017)

<http://www.google.es>

[2] Página Oficial OData (Marzo 2017)

<http://www.odata.org/>

[3] Página Oficial Power BI (Marzo 2017)

<https://powerbi.microsoft.com/es-es/>

[4] Github de RESTier (Junio 2017)

<https://github.com/OData/RESTier>

[5] Documentación RESTier (Junio 2017)

<http://odata.github.io/RESTier/>

[6] Información Bussiness Intelligence (Abril 2017)

[https://en.wikipedia.org/wiki/Business\\_intelligence\\_19/06/2017/](https://en.wikipedia.org/wiki/Business_intelligence_19/06/2017/)

[7] Información Bussiness Intelligence (Mayo 2017)

[https://www.microsoft.com/es-es/sql-server/business-intelligence\\_19/06/2017](https://www.microsoft.com/es-es/sql-server/business-intelligence_19/06/2017)

[8] Información Bussiness Intelligence (Junio 2017)

<https://www.wearemarketing.com/blog/yo-no-soy-tonto-o-las-mejores-herramientas-de-business-intelligence>

[9] Iniciación Power BI (Mayo 2017)

<https://powerbi.microsoft.com/es-es/get-started/>

[10] Iniciación Power BI (Mayo 2017)

<https://powerbi.microsoft.com/es-es/documentation/powerbi-desktop-connect-odata/>

## PRESENTACIÓN DE DATOS ODATA EN POWER BI

[11] Funcionamiento API REST (Mayo 2017)

<http://ualmtorres.github.io/howtos/RedisPHP/PhalconRedisAPIREST/>

[12] Consultas de fuentes OData (Mayo 2017)

<https://stackoverflow.com/questions/16990252/how-to-expose-a-mysql-database-as-odata>

[13] Investigación de framework LightSwitch (Mayo 2017)

<https://msdn.microsoft.com/es-es/library/hh674023.aspx>

[14] Investigación de framework odataphpprod (Mayo 2017)

<https://github.com/dineshkummarc/odataphpprod>

[15] Conexión de SQLserver con php (Abril 2017)

<https://juanalbertogt.wordpress.com/2014/04/12/configurar-conexion-sql-server-con-php-sql-server-2008/>

[16] Librerías OData (Mayo 2017)

<http://www.odata.org/libraries/>

[17] Referencias de servicios OData (Junio 2017)

<http://www.odata.org/odata-services/>

[18] Northwind base de datos (Junio 2017)

<https://www.microsoft.com/en-us/download/details.aspx?id=23654>

[19] Microsoft Imagine para Universidad de Almería (Marzo 2017)

<https://e5.onthehub.com/WebStore/ProductsByMajorVersionList.aspx?ws=6ec1b2c8-749b-e011-969d-0030487d8897&vsro=8&JSEnabled=1>

[20] Fuentes OData de Microsoft (Junio 2017)

<https://support.office.com/es-es/article/Obtener-acceso-a-fuentes-de-OData-desde-Power-Query-82bd48b9-6a2f-4f2a-83c9-6868ba1d30e2?ui=es-ES&rs=es-ES&ad=ES>

[21] Investigación de framework Olingo (Junio 2017)

<http://olingo.apache.org/>

[22] Investigación de framework SDL (Junio 2017)

<https://github.com/sdl/odata>

[23] Investigación de framework OData4j (Junio 2017)

<http://odata4j.org/>

[24] Postman página oficial (Junio 2017)

<https://www.getpostman.com/>





## Resumen

En este proyecto se pretende demostrar el uso y la gran utilidad de Power BI para la inteligencia de negocio. Power BI, como ya veremos a lo largo de este documento, admite distintas fuentes de datos para poder conseguir información sobre nuestra entidad. Este proyecto tratará de una empresa ficticia a la que someteremos al paso por la herramienta Power BI. Aunque esta herramienta admite distintas fuentes de datos, como puede ser un archivo Access o un archivo Excel, para este proyecto hemos querido centrarnos en la fuente de datos OData. Esto es debido a que, al igual que Power BI, está desarrollada por Microsoft. Aunque OData ofrece una gran fuente, veremos que no existen muchas más a disposición del usuario. Además, crear nuestra fuente de datos propia ofrece una gran funcionalidad y facilita el traspaso de datos.