



Diseño e implementación de una aplicación BI para TheNeoMarket.

Ramón Cotillas Jiménez
Computación
TFG Business Intelligence

Xavier Martínez Fontes
Atanasi Daradoumis Haralabus

Enero/2018



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diseño e implementación de una aplicación BI para TheNeoMarket.</i>
Nombre del autor:	<i>Ramón Cotillas Jiménez</i>
Nombre del consultor/a:	<i>Xavier Martinez Fontes</i>
Nombre del PRA:	<i>Atanasi Daradoumis Haralabus</i>
Fecha de entrega (mm/aaaa):	01/2018
Titulación::	<i>Computación</i>
Área del Trabajo Final:	<i>TFG Business Intelligence</i>
Idioma del trabajo:	Español
Palabras clave	<i>Qlik Sense, balanced scorecard</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Este trabajo de fin de grado describe los procesos necesarios para diseñar e implantar un cuadro de mando integral para una plataforma de venta por Internet.</p> <p>Las herramientas seleccionadas para el proceso han sido: Simba, para las conexiones con las fuentes de datos, y Qlik Sense, para el resto de procesos puesto que ofrece soluciones para realizar todos los pasos.</p> <p>El proceso de creación del cuadro de mando integral ha sido dividido en dos partes, la fase de diseño y la fase de implementación.</p> <p>La fase de diseño describe la recopilación de datos, el diseño de la arquitectura de la aplicación, la definición de KPIs y el diseño de las visualizaciones.</p> <p>La fase de implementación describe el proceso ETL así como la fase de creación de visualizaciones.</p> <p>Finalmente se describe los pasos necesarios para poder cargar el archivo resultante y poder visualizarlo en otro equipo.</p> <p>El producto final resultante es un cuadro de mando integral compuesto por varios paneles en los que se agrupan diferentes lógicas de visualización: panel financiero, panel de productos, panel de planes, panel de visitas, panel de tiendas y productos, y panel de control de zonas.</p>	

Abstract (in English, 250 words or less):

This final university project describes the necessary steps to design and implement a Balanced Scorecard for an Internet sales platform.

The tools selected for this process have been: *Simba*, it is used to create the connections with data sources, and Qlik Sense for the rest of the processes because Qlik Sense offers solutions to develop all steps.

The creating process of the balanced scorecard has been divided in two parts, the design phase and the implementation phase.

The design phase describes the data collection, the design of the application's architecture, the definition of KPIs and the design of the visualizations.

The implementation phase describes ETL process and the creation phase of visualizations.

Finally, it describes the necessary steps to load resulting file and to be able to view it on another platform.

The final product is an integral scorecard composed of several panels where different logics of display are grouped: the financial panel, the product panel, the plans panel, the visit panel, the shop and product panel, and the panel of zones control.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo.....	3
1.5 Breve resumen de productos obtenidos.....	6
2.- Fase de diseño.....	7
2.1.- Análisis de las Fuentes de datos:.....	7
2.2.- Arquitectura de la solución BI.....	13
2.3.- Definición de KPIs (Key performance indicator) :.....	16
2.4.- Diseño de las visualizaciones.....	18
3.- Implementación.....	28
3.1.- Proceso ETL.....	28
3.2.- Implementación del cuadro de mando integral.....	34
4. Producto final.....	55
5. Conclusiones.....	57
6. Bibliografía.....	58
7. Anexos.....	59
I.- Recopilación de datos de las tablas.....	59
II.- Script de carga.....	70

1. Introducción

Este trabajo trata sobre la creación de un cuadro de mando integral para el análisis B.I. de la plataforma www.theneomarquet.com.

El objetivo es la creación de diferentes cuadros de mando o Dashboard compuestos por diferentes visualizaciones e informes, pudiendo de esta manera analizar tanto los resultados de la lógica de negocio de la plataforma, así como los datos financieros.

En definitiva, dotar a esta startup de herramientas con las que analizar los resultados de su negocio y poder tomar decisiones basadas en datos.

1.1 Contexto y justificación del Trabajo.

Con la aparición de grandes plataformas de venta de productos por Internet y de las grandes superficies tales como Amazon, MediaMarkt, El Corte Inglés, etc. al pequeño comercio le está costando subsistir. Desde el 2007 al 2016 las ventas del pequeño comercio han descendido un 30%¹. Esto está obligando a los pequeños autónomos a cerrar o a reinventarse, pero la creación de una página web y, sobretodo, su posicionamiento en las redes puede ser una estrategia cara y con unos sobrecostes difícilmente asumibles por un sector en decadencia.

En este marco es donde nace TheNeoMarket, una startup que ha creado un portal web y una aplicación Android con la que ofrecer a profesionales un apoyo para afrontar estos problemas. TheNeoMarket ha desarrollado una plataforma en la que el pequeño comerciante puede crear sus tiendas de una manera sencilla y donde los usuarios pueden buscar productos que deseen comprar, teniendo la particularidad de que estas búsquedas se realizan en la zona del usuario, es decir, cuando un cliente busca un producto esta búsqueda sólo mostrará los resultados de aquéllos que estén cerca de él, y así el cliente puede acercarse en un momento a comprarlo, o el comerciante entregárselo o contratar un servicio de mensajería que lo haga por él.

En resumen, TheNeoMarket ofrece al pequeño comercio una herramienta para poder competir con las grandes plataformas de venta por Internet y darle una mayor visibilidad. Además de ofrecer a sus usuarios una alternativa para encontrar productos que deseen comprar y disponer de ellos sin tener que desplazarse largas distancias o esperar varios días para tenerlos.

TheNeoMarquet está actualmente en la fase final de implementación; existe una página web y se está terminando la app android por lo que se tiene prevista su lanzamiento al mercado en un futuro cercano. Esta situación plantea un problema añadido al proyecto, ya que aunque está

1 https://www.elespanol.com/economia/20170217/194481053_0.html

toda la estructura de las bases de datos creadas, éstas no contienen nada más que datos de prueba. Por lo que para el desarrollo de la aplicación B.I. no se contarán con datos reales y se tendrá que desarrollar con un juego de datos de simulación siendo parte del ámbito del proyecto el generar estos datos.

1.2 Objetivos del Trabajo

Tras la reunión mantenida con Sergio Fernández el día 13 de Octubre se han identificado los siguientes objetivos:

Se ha de dotar a TheNeoMarquet de una infraestructura BI que pueda ser utilizada por el Ceo (así como por diferentes usuarios autorizados) para analizar desde diferentes puntos de vista lo que está ocurriendo en el negocio, y de esta manera facilitar la toma de decisiones basada en hechos y en datos.

Se han identificado los siguientes indicadores que se han de recuperar:

- Peticiones de apertura de la plataforma por zonas, tipo de cliente o fecha con alertas para abrir la plataforma cuando se han cumplido los límites establecidos.
- Gráfica evolutiva de captación de usuarios por fecha y tipo de plan (diferenciando por planes de pago y gratuitos).
- Dashboard financiero con ingresos y gastos, por zonas y tiempo con alertas de objetivos.
- Tipos de plan contratados por zona.
- Listado de empresas que les caduca el plan con periodicidad establecida.
- Detalles sobre el funcionamiento de las campañas:
 - Gráfico de ingresos por campañas
 - Empresas que han cambiado del plan gratuito a uno de pago gracias a las campañas
- Ranking de empresas que más ingresan.
- Detalles de los productos,
 - Ranking de categorías de productos por zonas.
 - Número de productos por zonas, categorías y fechas.
 - Análisis de ventas por tipo de tienda, categorías

Además, esta infraestructura creada ha de ser usable, escalable² y responsive³, por lo que se ha acordado que el desarrollo de la plataforma será sobre Qlik Sense ya que esta las aplicaciones creadas con esta herramienta facilitan cumplir con estos objetivos.

² La **escalabilidad** es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos. <https://es.wikipedia.org/wiki/Escalabilidad>

³ Que se adapte a cualquier dispositivo sin perder usabilidad.

1.3 Enfoque y método seguido

Tal y como se ha acordado con el cliente se creará una nueva aplicación usando la herramienta de B.I. Qlik Sense, ya que ésta, además de disponer de todo lo necesario para desarrollar una aplicación B.I., cuenta, entre otras, con las siguientes características:

- Un potente motor de indexación de datos, con el que mejorar la escalabilidad del proyecto.
- Dispone de un gran DataMarket con multitud de bases de datos, algunas de éstas gratuitas.
- Las aplicaciones generadas con Qlik sense son Responsive, puesto que ha sido diseñado para la movilidad, adaptándose a la perfección en todos los dispositivos.
- Qlik Sense está diseñado y enfocado a la simplicidad, sencillez y autoservicio, por lo que una vez terminado el proyecto el usuario podrá crear sus propias visualizaciones.
- Los gráficos son intuitivos y simples ofreciendo pautas para ayudar a los usuarios con menos experiencia.
- Una gran comunidad activa con documentación y resolución de dudas.

Por otro lado se quiere seguir una metodología ágil en el que el usuario esté involucrado en el proyecto, por lo que una vez el proyecto se encuentre en la fase de desarrollo de los Dashboards, se le dará acceso a la aplicación para que pueda validar el trabajo y que lo que está logrando cumple con sus expectativas.

1.4 Planificación del Trabajo

Recursos necesarios:

Para la creación de la plataforma B.I. se va a usar Qlik Sense, ya que esta herramienta cuenta con todo lo necesario para realizar todo el proceso de desarrollo de la aplicación B.I.

Será necesario tener acceso a las dos bases de datos de las que dispone la aplicación.

Tareas a realizar:

- Contexto y justificación - 25-09-2017 al 02-10-2017:

Definir el punto de partida del trabajo: ¿Cuál es la necesidad a cubrir? ¿Por qué es un tema relevante? ¿Cómo se resuelve el problema de momento? y aportación realizada (¿Qué resultado se quiere obtener?).

- Objetivos del trabajo - 03-10-2017 al 09-10-2017:

Definir los objetivos del trabajo, para ello es necesario concertar una reunión con el cliente con el objetivo de conocer la aplicación y determinar las necesidades y recursos para implementar en el proyecto.

- Planificación del TFG - 10-10-2017 al 16-10-2017:

Descripción de los recursos necesarios para realizar el trabajo, las tareas a realizar y una planificación temporal de cada tarea utilizando un diagrama de Gantt o similar. Esta planificación tendría que marcar cuáles son los hitos parciales de cada una de las PEC.

- Entrega PCE1 (Hito) – 16-10-2017:

Entrega del documento de requisitos del proyecto.

- Creación del entorno de trabajo – 17-10-2017 al 23-10-2017:

Instalación y configuración del software necesario para el desarrollo del proyecto.

- Análisis de la base de datos principal – 24-10-2017 al 30-10-2017:

Se realizará un análisis de la base de datos principal, determinando las tablas y/o campos que serán necesarios para los objetivos, las transformaciones necesarias y campos calculados. Esta parte también contempla si es necesario recuperar más información de la aplicación por lo que es probable que se tenga que trabajar con la aplicación y las bases de datos de ésta.

- Análisis de la base de datos financiera – 31-10-2017 al 06-11-2017:

Se realizará un análisis de la base de datos financiera, determinando las tablas y/o campos que serán necesarios para los objetivos, las transformaciones necesarias y campos calculados. Esta parte también contempla si es necesario recuperar más información de la aplicación por lo que es probable que se tenga que trabajar con la aplicación y las bases de datos de ésta.

- Proceso ETL de la parte financiera – 07-11-2017 al 13-11-2017:

Se ha de realizar el proceso de extracción, transformación y carga de los datos de la base de datos financiera en la aplicación Qlik Sense siguiendo las pautas establecidas en la tarea de análisis.

Este proceso también contempla la carga de datos de simulación en la base de datos financiera.

- Entrega PCE2 (Hito) – 20-11-2017:

Esta tarea es uno de los hitos a alcanzar, coincidente con las entregas parciales del proyecto. La entrega constará de los documentos de análisis de las bases de datos y del script ETL de la carga de la parte financiera.

- Proceso ETL de la parte principal – 14-11-2017 al 27-11-2017:

Se ha de realizar el proceso de extracción, transformación y carga de los datos de la base de datos principal en la aplicación Qlik Sense siguiendo las pautas establecidas en la tarea de análisis. Este proceso también contempla la carga de datos de simulación en la base de datos principal.

- Proceso ETL fuentes externas – 28-11-2017 al 04-12-2017

Se han de identificar los recursos necesarios de fuentes externas y se han de introducir en la aplicación Qlik. También se contempla la creación de tablas autogeneradas, como el calendario.

- Dashboard Financiero – 05-12-2017 al 18-12-2017:

Se ha de crear el cuadro de mando donde se recojan todos los indicadores, gráficas, KPI'S, etc. que afecten a la parte financiera, así como la realización de las pruebas necesarias.

- Entrega PCE3 (Hito) – 18-12-2017:

Esta tarea es uno de los hitos a alcanzar, coincidente con las entregas parciales del proyecto. La entrega constará del script ETL de la carga de la parte general y de las fuentes externas, es decir, junto con la anterior entrega, del script completo. También se entregará el DashBoard Financiero.

- Dashboard Principal – 19-12-2017 al 01-01-2018:

Se ha de crear el cuadro de mando donde se recojan todos los indicadores, gráficas, KPI'S, etc. que afecten a la parte general, así como la realización de las pruebas necesarias.

- Reporting – 02-01-2018 al 08-01-2018:

Se ha de crear el área donde se creen los informes requeridos por el usuario, así como la realización de las pruebas necesarias.

- Finalizar memoria – 09-01-2018 al 15-01-2018:

Esta tarea se ha destinado a la revisión, corrección y finalización de la memoria del proyecto.

- Entrega del proyecto – 15-01-2018:

Fecha final del proyecto.

El diagrama de Gantt resultante es el siguiente:

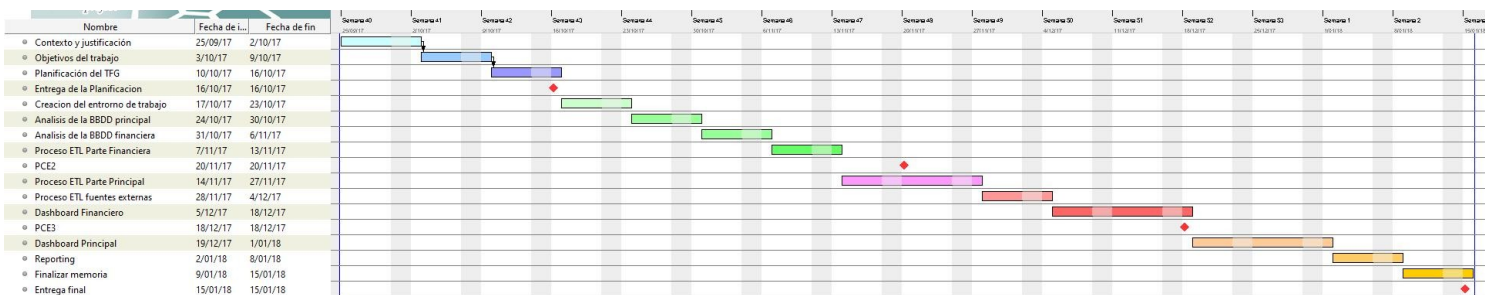


Diagrama de Gantt creado con la aplicación GanttProject.

1.5 Análisis de Riesgos:

Riesgos detectados:

1. El hecho de realizar un proyecto para la aplicación de un tercero hace que, en algunos casos, se dependa de la disponibilidad de éste.
2. El hecho de que la aplicación no esté terminada, es un riesgo a tener en cuenta, ya que, aunque esté en la fase final, ésta puede ser susceptible de sufrir algún cambio.
3. La aplicación trabaja sobre bases de datos poco convencionales.

1.6 Breve resumen de productos obtenidos.

Los productos a obtener son los siguientes:

- Modelo de datos en Qlik Sense desde el cual un usuario avanzado podría ampliar la aplicación BI con nuevas app's y visualizaciones.
- Un cuadro de mando integral desde el cual dar respuesta a las diferentes demandas recogidas en el apartado 1.2.

2.- Fase de diseño

En este apartado se abordan todos los pasos que se han dado para realizar el diseño del cuadro de mando integral.

2.1.- Análisis de las Fuentes de datos:

Con el fin de conocer mejor la aplicación y la lógica con que se generan los datos, se ha hecho un análisis de las fuentes de datos de las que dispone TheNeoMarket.

Este apartado se ha dividido en dos partes correspondientes a los dos almacenes de los datos que usa la aplicación, MongoDB y Stripe.

Nota: En el apartado de anexos se han recogido la descripción completa de los campos de cada tabla.

2.1.1.- Base de datos en MongoDB.

TheNeoMarket almacena la mayor parte de la información generada en MongoDB. El diagrama de la base de datos generada en mongoDB es el siguiente:

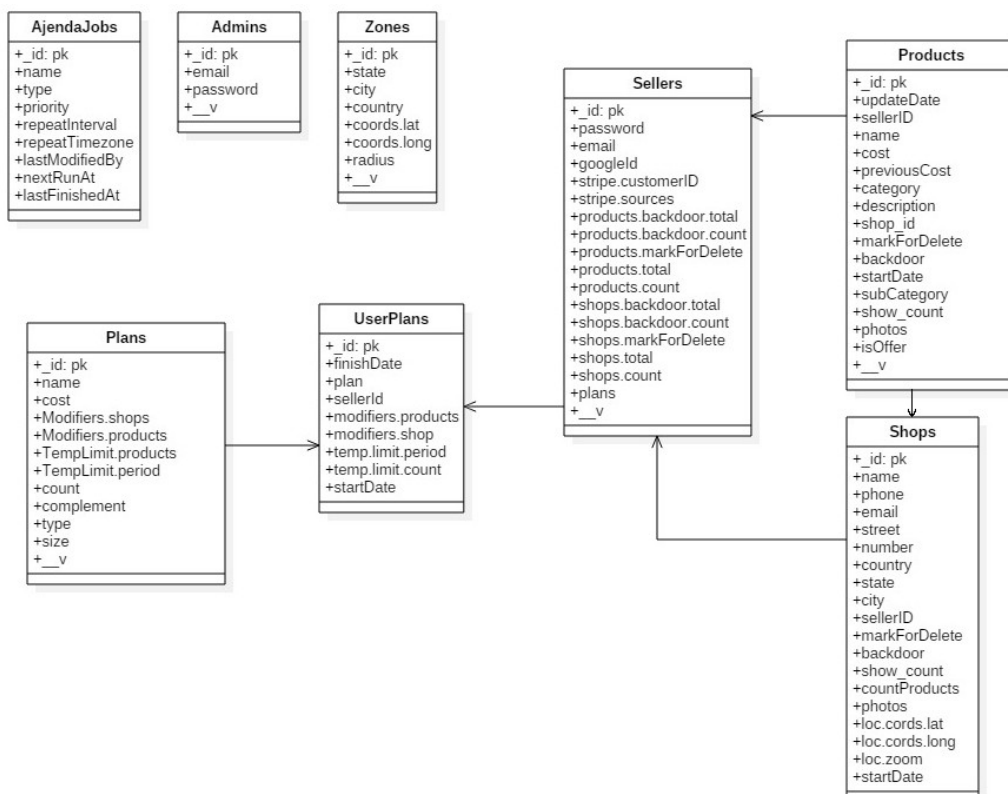


Tabla Admins.

Esta tabla contiene los datos de los administradores de la aplicación (usuario, contraseña, etc.).

Esta tabla no tiene datos relevantes para el análisis BI, por lo que no será cargada en el modelo de datos.

Tabla AgendaJobs.

Esta tabla es una tabla auxiliar donde se planifican tareas automáticas que se han de realizar en la aplicación.

Esta tabla no tiene datos relevantes para el análisis BI, por lo que no será cargada en el modelo de datos.

Tabla Plans.

Esta tabla es donde se almacena la información de los tipos de planes que dispone la aplicación (en su primera fase) la única fuente de ingresos de ésta.

Los campos más relevantes son su identificador (clave primaria), el nombre del plan y su tipo.

Tabla products.

Esta tabla es donde se almacena la información de los productos mostrados en la aplicación. Esta tabla está relacionada con la tabla shops y sellers, por lo que un producto pertenece a una única tienda y a un único vendedor.

La información más importante que almacena es:

- Identificador del producto
- Fecha de alta
- Identificador de la tienda
- Categoría
- Marca de trastienda (indica si el producto se muestra o no)
- Número de visitas al producto

Nota: Falta un histórico de los productos o una fecha de cuando el producto es eliminado.

Tabla sellers

Tabla donde se almacena la información de los vendedores. Las relaciones de esta tabla son las siguientes:

- Con la tabla de tiendas (cada usuario tiene entre 1 y n tiendas).
- Con la tabla de UsersPlans (cada usuario tiene entre 1 y n usersPlans).
- Con la tabla products, pero esta relación se eliminará para romper la relación circular.
- Con la tabla Customers de Stripe; la tabla customers es la tabla de clientes de stripe, es decir, seller (mongo) y customer (stripe) son tablas complementarias (un registro de customer tendrá su registro en seller y viceversa).

La información más importante que almacena es:

- Identificador del seller
- Email del seller
- Identificador de la tabla customer
- Identificador de la tabla de usersPlans

Tabla Shoppers

Tabla con la información de los compradores. La tabla shoppers existe como tal en la aplicación, pero su lógica aún no está implementada en ella de modo que no se están registrando datos de los compradores, por lo que el análisis de los datos de los compradores queda fuera del proyecto.

Tabla shops

Tabla con la información de las tiendas almacenadas en la aplicación. Cada tienda pertenece únicamente a un vendedor, y en cada tienda hay entre 0 y n productos. La información más importante que tiene la tienda es:

- Identificador de la tienda.
- Nombre.
- dirección y ubicación (lat/long).
- Identificador del vendedor.
- Número de productos.
- Fecha de creación.

Nota: Falta una tabla de históricos o una fecha de cuando la tienda es eliminada.

Tabla UserPlans

Tabla donde se guarda la información de los planes por usuario. Cada vez que un usuario contrata un plan, en esta tabla se guarda un registro. Esta tabla es complementaria con la tabla de stripe subscriptions.

Los datos más importantes que almacena son:

- Identificador del registro.
- Identificador del plan.
- Fecha de comienzo y de fin.
- Identificador del vendedor.

Tabla Zones

Esta tabla almacena la información (dirección, lat/long y radio) de las zonas habilitadas para que funcione la aplicación.

Tabla ZoneRequest

Tabla con la información de las solicitudes de apertura de la plataforma:

Solicitud de apertura: La aplicación sólo está operativa en las zonas que han sido dadas de alta (tabla zones), si te conectas en una zona que no lo esté, no se podrá operar con ella (te permite elegir otra zona operativa y operará como si estuvieras en ella). Sin embargo, si que se podrá hacer una solicitud de apertura de la aplicación.

En esta zona se almacena las direcciones, el email del solicitante y la latitud y longitud cogida del punto de conexión del usuario.

Nota: Esta tabla no estaba implementada, pero tras una reunión con el cliente se le hizo notar la importancia de que desarrollara la tabla y la lógica para tratarla.

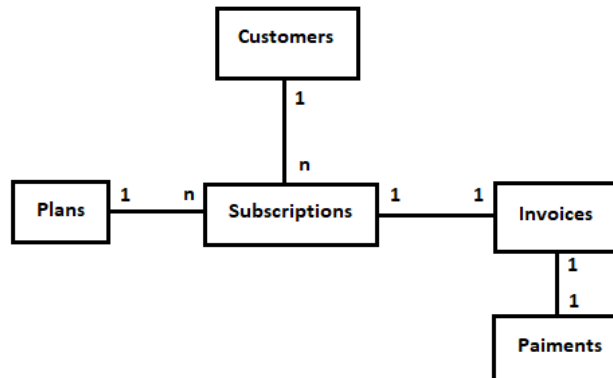
Nota: Sería interesante conseguir la lat/long de la ciudad, ya sea mediante la aplicación o por una tabla externa, con el objetivo de poder agrupar estos puntos en un mapa.

2.1.2.- Stripe (plataforma de pago).

Stripe es la plataforma que TheNeoMarquet ha elegido para aceptar los pagos a través de su página web y su plataforma móvil. Con ella se gestionan tanto las suscripciones como todo lo referente a la facturación y los impuestos.

Esta plataforma ofrece diferentes tablas prediseñadas para gestionar y almacenar todos los movimientos económicos hechos por Internet, por ello, estas tablas tienen campos para contemplar toda la información que puede generar un pago por Internet. Como es obvio, TheNeoMarket usa únicamente los que necesita.

Rompiendo las relaciones circulares, el esquema de tablas de Stripe es el siguiente:



A continuación se han analizado las diferentes tablas que se usan de Stripe y que campos son los más interesantes.

Tabla payments

En esta tabla se almacena un registro por cada pago que se realiza. Esta tabla está relacionada con las tablas customers y invoices, es decir, un pago tiene una factura y esta hecha por un cliente.

De esta tabla los campos que interesan son la fecha del pago, el importe del pago, todos los cargos y/o impuestos que puede tener el pago así como las tasas de la aplicación Stripe.

Tabla Customers

En esta tabla de Stripe se almacena los datos de los clientes de theNeoMarket, es una tabla complementaria de sellers. De esta tabla sólo interesa la fecha de creación (no existe este campo en sellers), ya que el resto de campos o no contienen información, o no es importante, o está duplicada.

Nota: No se dispone de información personal almacenada en la aplicación, única y exclusivamente se dispone del email (necesario para el registro); por lo que en los listados en los que se referencie al cliente sólo se podrá hacer por su email.

Tabla Invoices

Tabla con los datos de las facturas generadas por los pagos en la plataforma Stripe. En TheNeoMarket, las facturas se generan automáticamente cuando se genera un pago (no hay deuda, ni tipos de facturación), por lo que una factura complementa la información de un pago (todos los importes, descuentos y tasas ya están contemplados en la tabla pagos).

Esta tabla está relacionada con la tabla de subscriptions, ya que una factura corresponde siempre a una suscripción.

Tabla Plans

Tabla de Stripe con los datos de los planes. En esta tabla se almacenan datos complementarios (y duplicados) de los planes que ofrece la aplicación.

Tabla Subscriptions

En esta tabla se almacena la información de las suscripciones, en ella se relaciona el cliente con el plan al que se ha suscrito y la cantidad de éstos. También se almacena la información de comienzo del plan y la de final de éste.

Esta tabla es complementaria a la tabla de usersPlans de mongoDB.

2.1.3.- Fuentes externas.

Costes

Actualmente no se están recogiendo los datos de los costes de mantenimiento de la aplicación ya que éstos son inapreciables (60 euros al año por hosting de la página). Se creará un excel simulando estos costes.

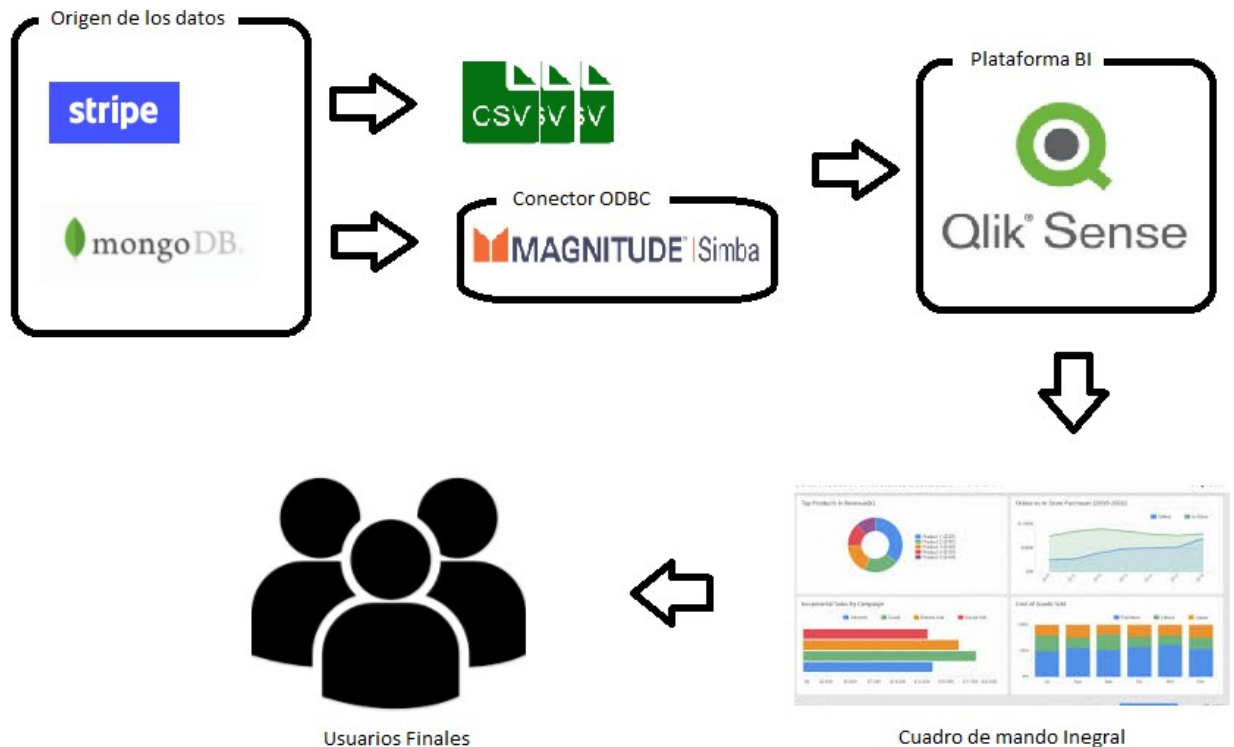
Lat/long

Aunque se está recogiendo la lat/long de la ubicación de las tiendas, si se quiere agrupar estos puntos en una localidad/ciudad dentro de un mapa, es necesario contar con la lat/long de ésta; por lo que será necesario una fuente de datos con las latitudes y longitudes de las localidades.

2.2.- Arquitectura de la solución BI.

Este apartado se ha analizado cuál es el flujo que ha de seguir la información desde las fuentes de datos hasta el usuario receptor, y qué herramientas se han de usar para ello.

A continuación se muestra un gráfico con la arquitectura de la solución BI en el que se puede observar este flujo.



Orígenes de los datos.

Stripe: Es una compañía estadounidense de tecnología que opera en más de 25 países y que permite tanto a particulares como a empresas aceptar pagos a través de Internet. Es la plataforma de pagos que usa TheNeoMarket. Desde Stripe se pueden generar archivos CSV que serán recogidos por Qlik sense.

MongoDB: TheNeoMarket usa la base de datos MongoDB para almacenar los datos generados por la aplicación. MongoDB es una base de datos NoSQL, en lugar de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB los guarda en estructuras de datos en documentos similares a JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

Conector ODBC.

Aunque Qlik Sense dispone de múltiples posibilidades para conectarse a diferentes fuentes de datos, muchas de éstas están disponibles en su opción de pago, por lo que para la conexión con la base de datos Mongo DB se ha optado por un producto externo.

El conector ODBC es un controlador para conectarse a un servidor de base de datos MySQL a través de la interfaz de programa de aplicación (API) Open Database Connectivity (ODBC), que es el medio estándar para conectarse a cualquier base de datos. Los usuarios pueden conectarse desde aplicaciones y entornos de programación comunes, como Microsoft Access, Excel o Qlik.

Magnitude Simba: Simba Technologies (www.simba.com) es una empresa que ofrece soluciones y productos de análisis y acceso a datos basado en estándares para fuentes de datos relacionales y multidimensionales. Simba proporciona soluciones de conectividad de datos a muchas de las compañías líderes en el mundo en diversas plataformas, incluidas Windows, Mac, UNIX, Linux y a muchas plataformas de dispositivos móviles.

Los productos de conectividad de datos de Simba admiten múltiples interfaces, incluidos ODBC, JDBC, OLE DB, OLE DB para OLAP (ODBO), XML para análisis (XMLA), ADO, ADO.NET, ADOMD y ADOMD.NET.

Plataforma BI.

La plataforma de análisis de datos que se emplea en este proyecto es Qlik Sense Desktop, la cual es la versión gratuita de Qlik Sense Server.

Qlik Sense es una herramienta que engloba todos los procesos necesarios para la creación de una aplicación BI: un editor de carga de datos donde realizar la fase ETL al completo y una plataforma de visualización la cual, gracias a una sencilla interfaz y a la funcionalidad de arrastrar y soltar, permite a los usuarios de negocio crear informes personalizados, cuadros dinámicos y visualizaciones interactivas de manera fácil y rápida.

Sus principales características con las que cuenta son las siguientes:

- Modelo de selección asociativo: Hacer selecciones es el principal método de interacción con Qlik Sense. Las selecciones filtran un subconjunto de los datos que se han cargado en Qlik Sense. Las selecciones se utilizan para centrarse en algún elemento sobre el que desee obtener más información. Qlik Sense responde mediante valores codificados por colores en función de sus diferentes estados.
- Apps: La visualización de datos en Qlik Sense está estructurada en APP's. Una apps es una recopilación de elementos reutilizables de datos (dimensiones, medidas y visualizaciones), hojas e historias. Es una

entidad autónoma que incluye los datos que se han de analizar en un modelo de datos estructurado.

- **Responsive:** El diseño de las apps se realiza sobre un diagrama que no se puede modificar, lo que induce al usuario a crear visualizaciones sencillas que respeten las principales normas de diseño minimalista y enfocado en el análisis, simplemente arrastrando las visualizaciones y campos de fórmulas deseadas. De esta manera, las visualizaciones se adaptan automáticamente a la resolución de la pantalla que se esté utilizando, logrando de esta manera un diseño auto-responsivo
- **Motor de indexación de Qlik;** Explorar, asociar y combinar información de nuevas formas que no eran posibles en otros productos sin tener que empezar desde cero para crear nuevas visualizaciones. Gracias al uso del motor de indexación de Qlik y su experiencia asociativa, los usuarios pueden explorar y hacer cualquier pregunta de forma dinámica.

Cuadro de mando Integral (Balanced ScoreCard).

Según el libro *The Balanced ScoreCard: Translating Strategy into Action*, Harvard Business School Press, Boston.

"El BSC es una herramienta revolucionaria para movilizar a la gente hacia el pleno cumplimiento de la misión a través de canalizar las energías, habilidades y conocimientos específicos de la gente en la organización hacia el logro de metas estratégicas de largo plazo. Permite tanto guiar el desempeño actual como apuntar al desempeño futuro. Usa medidas en cuatro categorías -desempeño financiero, conocimiento del cliente, procesos internos de negocios y, aprendizaje y crecimiento- para alinear iniciativas individuales, organizacionales y trans-departamentales e identifica procesos enteramente nuevos para cumplir con objetivos del cliente y accionistas. El BSC es un robusto sistema de aprendizaje para probar, obtener retroalimentación y actualizar la estrategia de la organización. Provee el sistema gerencial para que las compañías inviertan en el largo plazo -en clientes, empleados, desarrollo de nuevos productos y sistemas- más que en gerenciar la última línea para bombear utilidades de corto plazo. Cambia la manera en que se mide y maneja un negocio".

El Cuadro de mando integral sugiere que veamos a la organización desde cuatro perspectivas, cada una de las cuales debe responder a una pregunta determinada:

- **Perspectiva de Desarrollo y Aprendizaje:** ¿Podemos continuar mejorando y creando valor?
- **Perspectiva Interna del Negocio:** ¿En qué debemos sobresalir?
- **Perspectiva del Cliente:** ¿Cómo nos ven los clientes?

- Perspectiva Financiera: ¿Cómo nos vemos a los ojos de los accionistas?

Por lo tanto, el cuadro de mando integral resultante de este proyecto ha de responder todas estas cuestiones.

2.3.- Definición de KPIs (Key performance indicator) :

Los KPI o indicadores de negocio son una serie de métricas que se utilizan para tener un mejor conocimiento de la productividad de las acciones que se realizan en la actividad analizada, cuyo objetivo es el poder medir, comparar y decidir qué tipo de acciones son las mejores para los objetivos marcados y, así, servir de guía para la toma de decisiones.

Pero antes de definir los indicadores es necesario conocer los objetivos de la empresa y qué problemas se quieren resolver. Para ello se ha usado un análisis DAFO⁴

Análisis Interno.	Fortalezas	Debilidades
	<ul style="list-style-type: none"> - Los costes de mantenimiento son muy bajos ya que la aplicación esta diseñada para que no se disparen los costes. - Gran oportunidad de clientes. -La aplicación está diseñada por zonas, por lo que podría funcionar en algunos sitios u orientar la publicidad de manera local, por lo que los costes no se disparan -Con la aplicación funcionando se puede conocer los intereses y tendencias de una determinada zona, lo que puede llegar a monetizarse. 	<ul style="list-style-type: none"> - La competencia es principalmente multinacionales muy fuertes. - No hay inversores y no se cuenta con un gran capital para publicitarlo. - El proyecto todavía no se ha comercializado, por lo que es necesario coger una masa crítica para que este empiece a andar. - En un principio la fuente de ingresos viene exclusivamente de la venta de planes (un único producto con diferentes nombres).
Análisis Externo.	Oportunidades	Amenazas
	<ul style="list-style-type: none"> - La aplicación está orientada principalmente al pequeño comercio dándole herramientas para competir con los grandes. 	<ul style="list-style-type: none"> - La falta de capital puede hacer atrasar o incluso abortar el proyecto. - Los servicios ofrecidos están de moda, por lo que es fácil que surjan muchos competidores en un corto plazo de tiempo.

4.- El análisis DAFO es una herramienta de estudio de la situación de una empresa, institución, proyecto o persona, analizando sus características internas (Debilidades y Fortalezas) y su situación externa (Amenazas y Oportunidades) en una matriz cuadrada.

Teniendo en cuenta el análisis DAFO, es más fácil definir los objetivos de la empresa y basándose en éstos definir los indicadores necesarios. Estos indicadores han sido clasificados en función de la perspectiva del cuadro de mando integral al que pertenezcan.

Perspectiva	Indicadores	Fórmula	Objetivo
Financiera	<ul style="list-style-type: none"> • Margen de beneficios. <ul style="list-style-type: none"> ◦ Del año en curso ◦ Del año anterior 	Sum (Beneficios – costes – tasas - impuestos)	<ul style="list-style-type: none"> - Conseguir los suficientes ingresos para que los creadores se puedan dedicar al 100% a la aplicación - Tener un control de los ingresos y gastos de la aplicación.
	<ul style="list-style-type: none"> • Costes. 	Sum (costes)	
Interna	<ul style="list-style-type: none"> • Número total de productos publicados. 	count(productos)	<ul style="list-style-type: none"> - Hacer un pequeño análisis de tendencias en la aplicación con el objetivo de, en un futuro, poder monetizar esta información. - Conoces la demanda de la aplicación para habilitar las zonas que no están operativas (crecer).
	<ul style="list-style-type: none"> • Número de productos mostrados. 	count(productos) donde backdoor = '0'	
	<ul style="list-style-type: none"> • Número total de visitas a productos. 	sum(productShowCount)	
	<ul style="list-style-type: none"> • Número total de visitas a tiendas. 	Sum(shopShowCount)	
	<ul style="list-style-type: none"> • Número de peticiones de apertura. 	Count(zoneRequest)	
Cliente	<ul style="list-style-type: none"> • Número total de usuarios. 	count(Sellers)	<ul style="list-style-type: none"> - Recabar información de los usuarios de la plataforma.
	<ul style="list-style-type: none"> • Número total de tiendas. 	count(shops)	
Desarrollo y aprendizaje	<ul style="list-style-type: none"> • Número de planes gratuitos. 	count(suscriptions) donde nombrePlan = plan gratuito	<ul style="list-style-type: none"> - Estudio del funcionamiento de los diferentes planes y si estos funcionan en la aplicación.
	<ul style="list-style-type: none"> • Número de planes de pago. 	count(suscriptions) donde nombrePlan <> plan gratuito	
	<ul style="list-style-type: none"> • Número total de planes. 	count(suscriptions)	

2.4.- Diseño de las visualizaciones.

Una vez se han definido los KPIs, los objetivos y teniendo en cuenta las peticiones del cliente recogidas en la primera fase, es necesario diseñar las visualizaciones que estarán contenidas en el cuadro de mando integral.

A continuación, se definen las visualizaciones necesarias; su finalidad, su categoría, el KPI que usa (si lo necesitara), la fuente/s y el gráfico que se va a usar para mostrarlo.

1.

Nombre	Beneficios por mes y año.
Finalidad	Mostrar la tendencia de los ingresos en el tiempo para llevar un control financiero. Informar a posibles inversores .
Categoría	Operacional.
KPI	Margen de beneficios.
Frecuencia medición	Mensual (aunque al uso es diario), así se tiene conciencia de los beneficios del mes en curso.
Fuente	Paiments.
Gráfico	Gráfico de barras.
Observaciones	e muestra una gráfica con los ingresos y los márgenes por mes en un diagrama de barras.

2.

Nombre	Beneficios del año actual y el anterior.
Finalidad	Comparar los márgenes del trimestre en curso con los del anterior trimestre, ver los totales del año en curso y llevar un control financiero. Informar a posibles inversores.
Categoría	Operacional.
KPI	Margen de beneficios.
Frecuencia medición	Diaria.
Fuente	Paiments.
Gráfico	KPI.
Observaciones	El objetivo inicial es superar siempre el trimestre anterior, por lo que el trimestre en curso se mostrará en rojo mientras no supere el mes anterior y en verde en el momento que lo supere. Estos indicadores reaccionarán a los filtros que se usen (menos a los de tiempo).

3.

Nombre	Beneficios por trimestre (año actual vs año anterior).
Finalidad	Mostrar la tendencia de los ingresos en el tiempo para llevar un control financiero. Informar a posibles inversores .
Categoría	Operacional.
KPI	Margen de beneficios.
Frecuencia medición	Trimestral (aunque al uso es diario), así se tiene conciencia de los beneficios del trimestre en curso.
Fuente	Paiments.
Gráfico	Gráfico de barras.
Observaciones	Se muestra una gráfica con los ingresos y los márgenes por trimestre en un diagrama de barras.

4.

Nombre	Beneficios por zonas.
Finalidad	Conocer cuáles son las zonas de mayor éxito de la app y cuáles las de menor. Informar a posibles inversores y llevar un control financiero.
Categoría	Operacional y estratégico.
KPI	Margen de beneficios.
Frecuencia medición	Diaria.
Fuente	Paiments y Zonas.
Gráfico	Tarta.
Observaciones	Cada porción muestra la cuota de beneficios por zona.

5.

Nombre	Beneficios por tipo de plan.
Finalidad	Conocer cuáles son los planes que más están funcionando en la aplicación. Informar a posibles inversores y llevar un control financiero.
Categoría	Operacional y estratégico.
KPI	Margen de beneficios.
Frecuencia medición	Diaria.
Fuente	Paiments y Plans.
Gráfico	Gráfico de tarta.

6.

Nombre	Ingresos por categoría de tienda.
Finalidad	Conocer qué tipos de tiendas son las que más ingresan. Informar a posibles inversores y llevar un control financiero.
Categoría	Operacional, estratégico.
KPI	Margen de beneficios.
Frecuencia medición	Diaria.
Fuente	Los ingresos se obtienen de la tabla de pagos de Stripe.
Gráfico	Gráfico de bloques.
Observaciones	Las tiendas no están categorizadas todavía, pero se prevé que lo estarán para la finalización del proyecto.

7.

Nombre	Costes.
Finalidad	Informar a posibles inversores y llevar un control financiero.
Categoría	Operacional.
KPI	Costes.
Frecuencia medición	Diaria.
Fuente	Tabla de costes simulada.
Gráfico	KPI
Observaciones	Como ya se ha comentado, actualmente no se está llevando un control de costes, pero se simularán con una tabla externa.

8.

Nombre	Ranking de pagos por clientes.
Finalidad	Conocer el listado de clientes vip.
Categoría	Operacional.
KPI	Margen de beneficios.
Frecuencia medición	Diaria.
Fuente	Tablas de pago + tablas de clientes (stripe o mongo)
Gráfico	Tabla.
Observaciones	Es interesante el filtrado por zona. Se recogerán todos los datos del cliente.

9.

Nombre	Total de productos publicados.
Finalidad	Conocer el número de productos que hay actualmente en la aplicación. Estudio de tendencias para poder monetizar la información.
Categoría	Estratégico.
KPI	Numero total de productos publicados.
Frecuencia medición	Diaria.
Fuente	Products.
Gráfico	KPI de la aplicación Qlik Sense.
Observaciones	Se muestra el total de productos publicados en la aplicación, este número reaccionará a los filtros que se usen.

10.

Nombre	Número de productos por mes y año.
Finalidad	Conocer la evolución de la app de manera diaria. Hacer un estudio de tendencias para poder monetizar la información. Llevar un control de los productos de la aplicación.
Categoría	Estratégico y operacional.
KPI	Número total de productos publicados.
Frecuencia medición	Diaria.
Fuente	Products.
Gráfico	Gráfico de líneas.

11.

Nombre	Porcentaje de productos por zonas y categoría.
Finalidad	Conocer la evolución de la app de manera diaria. Llevar un control de los productos de la aplicación. Estudio de tendencias para poder monetizar la información.
Categoría	Estratégico y operacional.
KPI	Número total de productos publicados.
Frecuencia medición	Diaria.
Fuente	Products, zonas.
Gráfico	Gráfico de bloques.

12.

Nombre	Número de productos por categoría.
Finalidad	Conocer la evolución de la app de manera diaria. Hacer un estudio de tendencias para poder monetizar la información. Llevar un control de los productos de la aplicación.
Categoría	Estratégico y operacional.
KPI	Número total de productos publicados.
Frecuencia medición	Diaria.
Fuente	Products, zonas.
Gráfico	Tabla, tarta.
Observaciones	La información presentada se adaptará con los filtros que se realicen (fecha, zona, plan, etc).

13.

Nombre	Número de productos por zonas.
Finalidad	Conocer la evolución de la app de manera diaria. Hacer un estudio de tendencias para poder monetizar la información. Llevar un control de los productos de la aplicación.
Categoría	Estratégico y operacional.
KPI	Número total de productos publicados.
Frecuencia medición	Diaria.
Fuente	Products, zonas .
Gráfico	Tabla, tarta.
Observaciones	La información presentada se adaptará con los filtros que se realicen (fecha, zona, plan, etc).

14.

Nombre	Peticiones de apertura por zona.
Finalidad	Ampliar las zonas donde la aplicación está operativa. Gestionar el crecimiento de la aplicación.
Categoría	Operacional.
KPI	Número de peticiones de apertura.

Frecuencia medición	Diaria.
Fuente	Tabla zoneRequest de mongoDB + tabla externa lat/long con los pueblos y ciudades
Gráfico	Tabla y mapa
Observaciones	Es necesario encontrar Datos externos con la lat/long de los pueblos y ciudades de España. Cuando la aplicación se expanda será necesario ampliar esta tabla con los diferentes países.

15.

Nombre	Número de visitas a productos por zona y categoría.
Finalidad	Estudio de tendencias.
Categoría	Estratégico.
KPI	Número total de visitas a productos.
Frecuencia medición	Diaria.
Fuente	Tabla products, Tabla de zonas.
Iniciativas	Conocer los intereses de los que visitan la app.
Gráfico	Tabla pivotante.

16.

Nombre	Número de visitas a tiendas por zonas.
Finalidad	Llevar un control de la app. Estudio de tendencias.
Categoría	Estratégico.
KPI	Numero total de visitas a tiendas.
Frecuencia medición	Diaria.
Fuente	Tabla shops + zonas.
Gráfico	Diagrama de Barras.

17.

Nombre	Listado de las tiendas más visitadas por zonas.
Finalidad	Estudio de tendencias.
Categoría	Estratégico
KPI	Número total de visitas a tiendas.
Frecuencia medición	Diaria.
Fuente	Tabla Shops.
Gráfico	Tabla.
Observaciones	El objetivo de este gráfico es combinarlo con un filtrado de zona.

18.

Nombre	Número total de usuarios.
Finalidad	Control de la aplicación. Conocer el numero de vendedores que hay en la aplicación
Categoría	Operacional.
KPI	Número total de usuarios.
Frecuencia medición	Diaria.
Fuente	Tabla Customers o Sellers.
Gráfico	KPI.

19.

Nombre	Número de tiendas por zonas.
Finalidad	Control de la aplicación . Gestión de campañas publicitarias. Conocer el estado de salud de la aplicación.
Categoría	Operacional y estratégico.
KPI	Contador de tiendas agrupado por zonas.
Frecuencia medición	Diaria.
Fuente	Tabla Shops + zonas.
Gráfico	Mapa.

20.

Nombre	Listado de vendedores que les caduca el plan en el mes en curso.
Finalidad	Evitar la pérdida de clientes. Gestión de ofertas comerciales.
Categoría	Operacional.
KPI	-
Frecuencia medición	Mensual.
Fuente	Tablas planes + sellers.
Gráfico	Tabla.
Observaciones	Se recuperará la información del vendedor así como el plan en curso y la fecha de finalización

21.

Nombre	% planes por zonas.
Finalidad	Campañas de promociones. Aprender de que planes son los que mejor funcionan.
Categoría	Operacional.
KPI	Número total de planes.
Frecuencia medición	Diaria.
Fuente	Suscripciones, UsersPlans, customers.
Gráfico	Gráfico de bloques.

22.

Nombre	Planes gratuitos vs planes de pago.
Finalidad	Campañas de promociones. Control de salud de la aplicación.
Categoría	Operacional y estratégica.
KPI	Número de planes Gratuitos, Número de planes de Pago.
Frecuencia medición	Diaria.
Fuente	Suscripciones + usersPlans.
Gráfico	Indicador.
Observaciones	Es interesante que se pueda filtrar por fechas y zonas.

23.

Nombre	Planes por zonas y tipo de plan.
Finalidad	Campañas de promociones. Control de salud de la aplicación.
Categoría	Operacional y estratégica.
KPI	Número total de planes.
Frecuencia medición	Diaria.
Fuente	Suscripciones + usersPlans + zonas.
Gráfico	Gráfico de bloques.

3.- Implementación.

Una vez finalizada la etapa de diseño se comienza la fase de implementación.

Esta fase está dividida en dos partes: el proceso ETL (*Extraction, Transformation and Load*) y la creación de las visualizaciones.

3.1.- Proceso ETL.

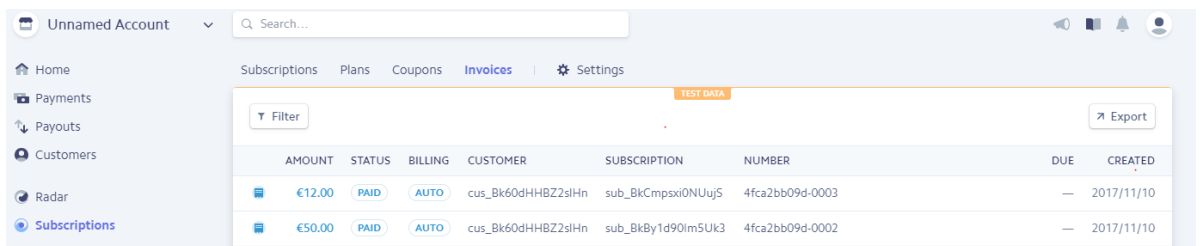
Como ya se observó en la arquitectura de la plataforma, Qlik da herramientas para abordar casi todas las fases por las que pasa el desarrollo de una solución B.I., por esa razón se ha usado principalmente Qlik Sense para el desarrollo del proceso ETL; aunque también ha sido necesario el uso de herramientas externas (principalmente por no contar con la plataforma de pago de Qlik sense).

A continuación se explica todo el proceso:

3.1.2.- Fase de Extracción.

1. Plataforma Stripe

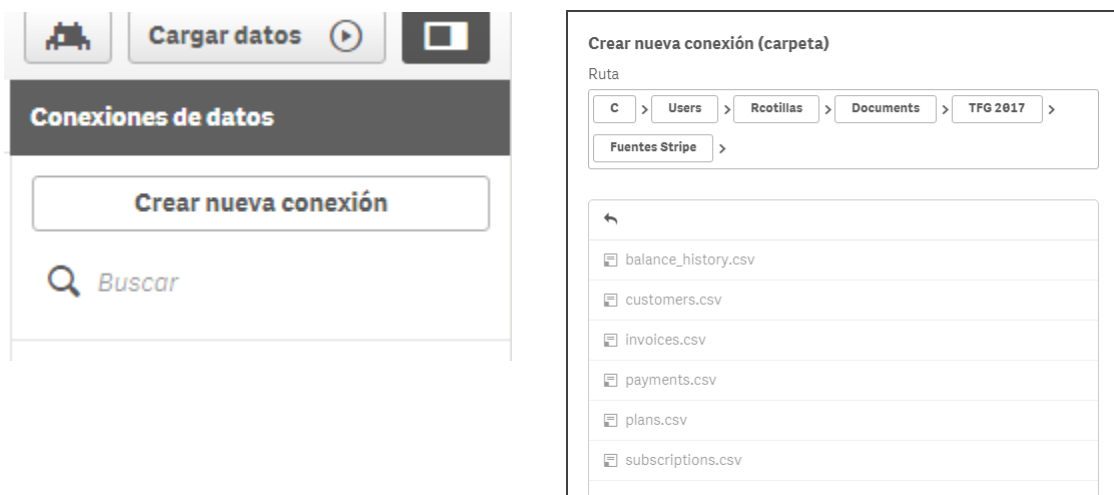
Para realizar el proceso de extracción en la plataforma Stripe, se cuenta con un sistema para poder exportar los datos en formato csv.



The screenshot shows the Stripe Invoices page. The table contains the following data:

AMOUNT	STATUS	BILLING	CUSTOMER	SUBSCRIPTION	NUMBER	DUE	CREATED
€12.00	PAID	AUTO	cus_Bk60dHHBZ2slHn	sub_BkCmpsi0NUuj\$	4fca2bb09d-0003	—	2017/11/10
€50.00	PAID	AUTO	cus_Bk60dHHBZ2slHn	sub_BkBy1d90Im5Uk3	4fca2bb09d-0002	—	2017/11/10

Estos archivos hay que conectarlos a Qlik, para ello se ha creado una conexión a la carpeta donde se han almacenado los archivos csv descargados de Stripe:

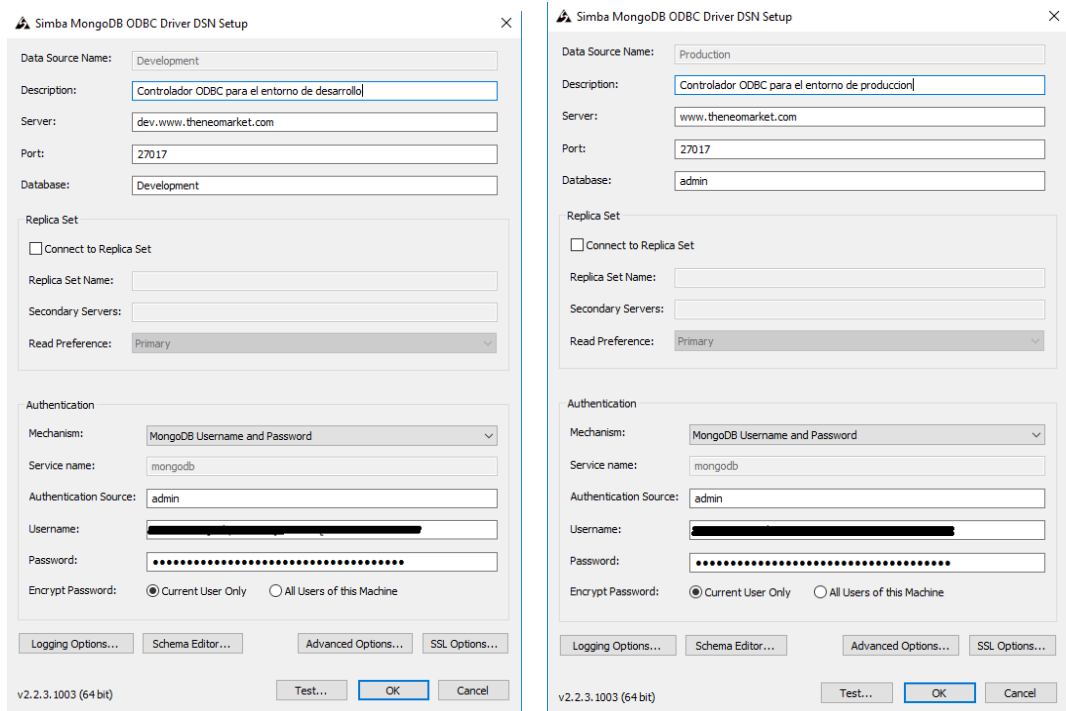


The screenshot shows the Qlik Sense interface for creating a new data connection. The left panel shows the 'Conexiones de datos' section with a 'Crear nueva conexión' button. The right panel shows the 'Crear nueva conexión (carpeta)' dialog with the following path: C > Users > Rcotillas > Documents > TFG 2017 > Fuentes Stripe >. Below the path, a list of CSV files is displayed:

- balance_history.csv
- customers.csv
- invoices.csv
- payments.csv
- plans.csv
- subscriptions.csv

2. Mongo Db:

Para la conexión a la base de datos mongoDB es necesario usar el controlador ODBC. Se han configurado dos controladores, uno para el entorno de desarrollo y otro para el entorno de producción.



Una vez configurados se pueden usar para crear la conexión con la base de datos MongoDB.



Una vez creada la conexión, se usa ésta para traer las tablas de MongoDB de datos al script.

3. Otras fuentes:

Se ha creado una conexión a la carpeta donde se han almacenado fuentes de datos externas.

3.1.2.- Fase de transformación.

En esta fase se han usado las conexiones creadas en el apartado anterior para traer los scripts de carga de las diferentes fuentes de datos, y ,sobre éstos, se han realizado las diferentes transformaciones para crear el modelo de datos resultante. El script ha sido organizado por pestañas, pero antes de comentarlas, se explican las acciones comunes que se han realizado en todo el script.

- Cargar sólo los campos necesarios: cuando se crea el script de carga de cada tabla, sólo se recogen los campos necesarios; dejando la información replicada y la innecesaria.
- Renombrar campos: Qlik crea las asociaciones entre las tablas por campos con el mismo nombre, por lo que al cargar las diferentes tablas se pueden generar asociaciones no deseadas y desaparecer las necesarias. Por ejemplo, si hay dos o más tablas con el campo Date, Qlik creará una asociación entre estas tablas por este campo, haciendo necesario renombrarlo en las diferentes tablas para romper esta asociación.
- Creación de claves compuestas y de un solo campo (en los casos necesarios): En Qlik las relaciones entre una tabla y otra están limitadas a un único campo, por lo que si existen foreign Keys de varios campos, éstos se han de transformar en uno sólo.
- Unión de tablas: Se han seguido dos criterios para los join de las tablas.
 - Tablas con información duplicada. En la plataforma Stripe existen tablas duplicadas respecto a las tablas de Mongo DB (se han unido para complementar la información de una con la otra).
 - Unión de tablas con información Especializada o categorizada. La arquitectura de Qlik está diseñada para que no se dupliquen los datos cuando están cargados. Las tablas solo guardan referencias a posiciones de memoria donde están los datos guardados, así éstos, solo se guardan una vez; es decir, si dos tablas tienen un registro con el dato "Barcelona", Qlik lo almacena una vez y en las tablas tiene almacenada 2 referencias a la posición de memoria del dato. Por ello es recomendable (siempre que no se pierda información) hacer un proceso de desnormalización, teniendo cuidado de que no se incremente el número de registros a un nivel insostenible.

- Tratamiento de nulos.
- En las tablas que se tengan que hacer conteos se ha creado un campo [nombreTabla]Count con valor 1 en todos sus campos, ya que es más eficiente usar la función sum() que la función count().

Una vez descritos los procesos comunes se describen las transformaciones realizadas en cada una de estas pestañas:

- Main:

En esta pestaña se recogen las configuraciones del modelo, tales como los formatos por defecto (fechas, moneda, separador decimal... etc). Además también se ha usado para la creación de variables globales y para la declaración de la conexión a mongo DB⁵

- Lar/Long:

En esta pestaña se carga una tabla auxiliar donde se recogen los diferentes municipios y ciudades de España, a qué comunidad pertenecen, su latitud y longitud, y un identificador (compuesto por la población la provincia y el país). Esta tabla es necesaria para geolocalizar las zonas en un gráfico de mapa.

Cuando se expanda la aplicación a diferentes países en este apartado se deberán cargar tablas con la misma información, pero del país deseado. Estas nuevas tablas se han de unir a la ya existente para hacer una única tabla de latitudes/longitudes.

Una vez se ha modelado la tabla se almacena en el archivo externo de datos llamado LatLong.qvd, esto es porque se ha estimado que este proceso solo se ha de realizar en la primera carga (o cuando se amplíe); después el código de este apartado se ha de dejar asteriscado.

- Paiments:

En este apartado se cargan las tablas de Stripe: payments y invoices. Estas dos tablas se han unido en una única tabla que recoge todos los datos de los pagos.

Aparte, a esta tabla se le ha añadido una tabla maestra de fechas la cual tiene las diferentes particularidades:

- La tabla maestra de fechas recoge todas las fechas comprendidas entre la fecha de la primera factura y la fecha del día de carga, de este modo se tendrán registros de días que no se han realizado pagos y éstos se podrán mostrar en las gráficas.

⁵ Se ha dejado operativa la conexión a la base de datos de desarrollo y asteriscada la conexión a la base de datos de producción.

- A partir de la las fechas dadas se han creado una serie de campos para enriquecer la información aportada por esta. Estos campos son:
 - El mes, año, día, día de la semana, trimestre, n.º de semana del año, mes y año.
 - Los siguientes indicadores booleanos:
 - CQFlag: Indica si la fecha está en el trimestre actual
 - LQFlag: Indica si la fecha está en el trimestre anterior al actual.
 - CYTDFlag: Indica si la fecha está en el año actual.
 - LYTDFlag: Indica si la fecha está en el año anterior al actual.
 - IsInYTD: Indica si el día del año de la fecha dada es mayor que el día del año actual.
 - IsInQTD: Indica si el día del trimestre de la fecha dada es es mayor al día del trimestre actual.
 - IsInMTD: Indica si el día del mes de la fecha dada es mayor del día del mes actual.
 - IsCurrentMonth: indica si la fecha se encuentra en el mes actual.
 - IsLastMonth: Indica si la fecha se encuentra en el mes anterior al actual.

- Zones Request:

En esta pestaña se cargan las tablas zoneRequest y LatLong (creada en la pestaña lat/long) y se unen, así la tabla zoneRequest contará con la geolocalización de la ciudad a la que referencia y se podrán situar los datos en un gráfico de mapa.

- Seller

En esta pestaña se cargan las tablas relacionadas con los vendedores, siendo éstas: customers de stripe y sellers de MongoDB. La tabla de customers ha sido cargada como tabla de mapeo ya que únicamente interesa de ella la fecha de creación del vendedor (puesto que el resto de datos están duplicados), la cual ha sido añadida a la tabla seller.

- Plans

En esta pestaña se cargan las tablas plans tanto de stripe como de mongoDB. Estas tablas se han unido en una ya que muestran información duplicada y complementaria.

- UsersPlans

En esta pestaña se cargan las tablas de userPlans (mongo) y subscriptions (stripe) y se unen. Como en la tabla de stripe no se guardan las suscripciones a

los planes gratuitos, se ha creado un campo “número de planes” que en el caso de ser gratuito es 1 y cualquier otro sera quantity.

- Products

En esta pestaña se carga la tabla products.

- Shops

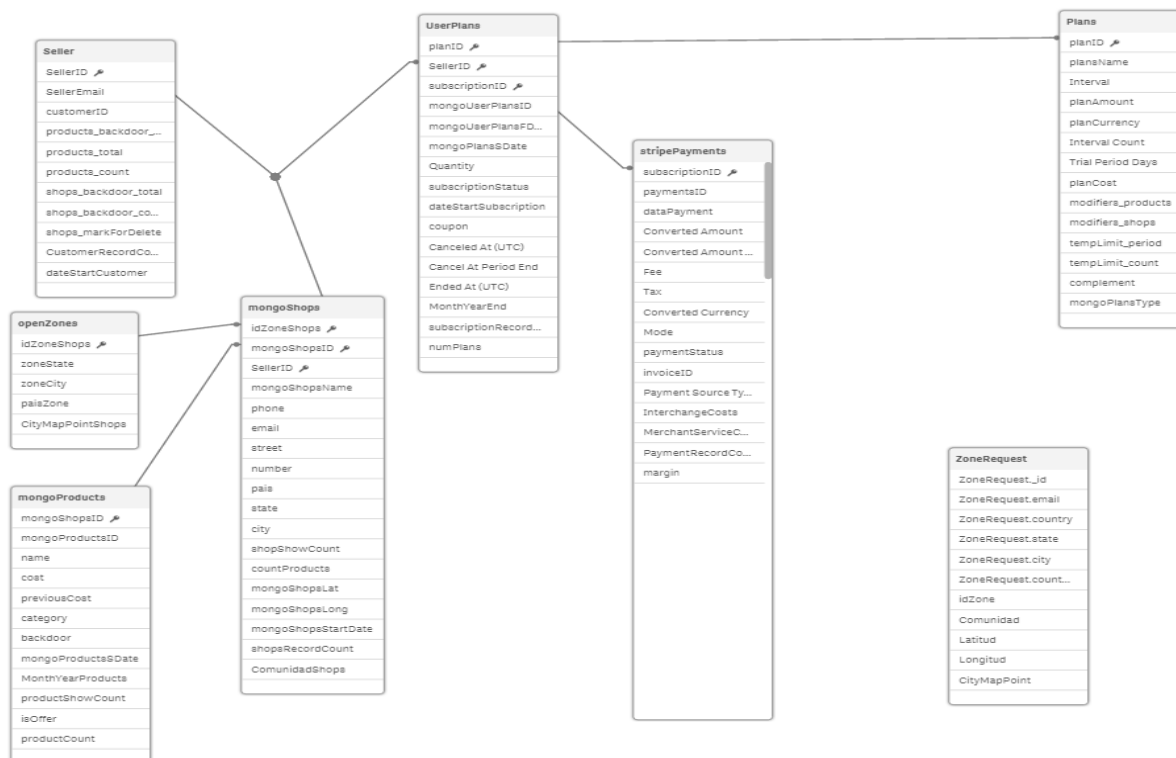
En esta tabla se carga la tabla shops y la tabla de openZones (zonas donde están ubicadas estas tiendas). Además se usa la tabla lat/long para añadir la comunidad.

*El script resultante ha sido adjunto en los apéndices.

3.1.3.- Fase de carga.

La fase de carga es sencilla, al disponer de un solo script de carga. Únicamente hay que pulsar el botón de “Cargar datos” en la vista de *Editor de carga de datos* y esperar a que éstos estén cargados en la app (Siempre que no se genere ningún error).

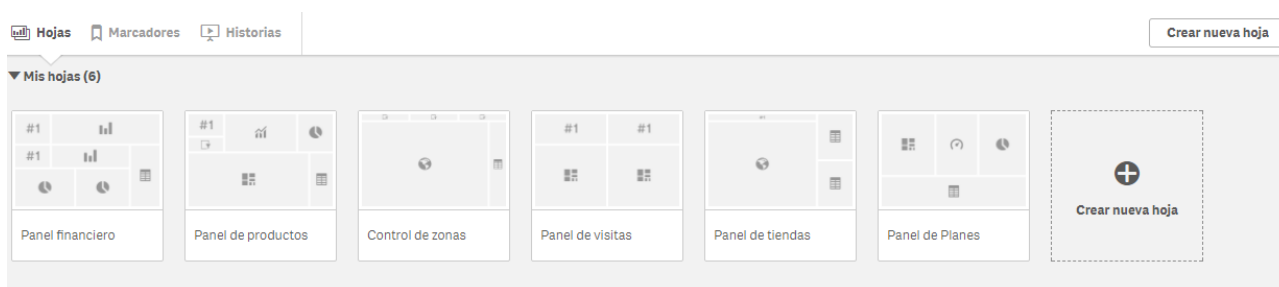
El Modelo de datos resultante es el siguiente:



3.2.- Implementación del cuadro de mando integral.

Una vez se ha completado la implementación del modelo de datos (proceso descrito en el apartado anterior), se procede a la creación de las visualizaciones. Para ello, dentro de la aplicación "theNeoMarket" creada en Qlik Sense, en el apartado de vista general, se han creado diferentes hojas, en las cuales se agrupan las distintas visualizaciones por funcionalidad.

Los Paneles (hojas) creados son los siguientes: Panel financiero, Panel de productos, Control de zonas, Panel de visitas, Panel de tiendas y Panel de planes.



Estos paneles se corresponden con las perspectivas que sugiere un cuadro de mando integral:

- Perspectiva Financiera → Panel financiero.
- Perspectiva Interna de Negocio → Panel de productos, panel de visitas, Panel de tiendas, panel de planes.
- Perspectiva del Cliente → Control de zonas, panel de planes.
- Perspectiva de Desarrollo y Aprendizaje → Control de zonas

A continuación se describe en detalle cada una de estos paneles:

Nota1: Los datos presentados en las visualizaciones son datos de simulación, estos datos se han añadido a la aplicación de tres maneras diferentes: mediante el uso natural del entorno de desarrollo, mediante scripts y manualmente.

De la generación de los scripts, del borrado de las tablas y de la actualización de las tablas se ha encargado el dueño de la aplicación. Sin embargo debido a la falta de tiempo del dueño hay datos que se han quedado inconsistentes, he arreglado la mayoría de éstos manualmente quedándose todas las visualizaciones coherentes, sin embargo en algunas sale el mensaje: "El conjunto de datos contiene valores negativos o cero que no se pueden mostrar en este gráfico".

Nota2: Las visualizaciones de Qlik no son estáticas, es decir, reaccionan en tiempo real a las selecciones, por lo que si se selecciona un tipo de plan o una fecha el resto de los gráficos se ven afectados por estos filtros (siempre que

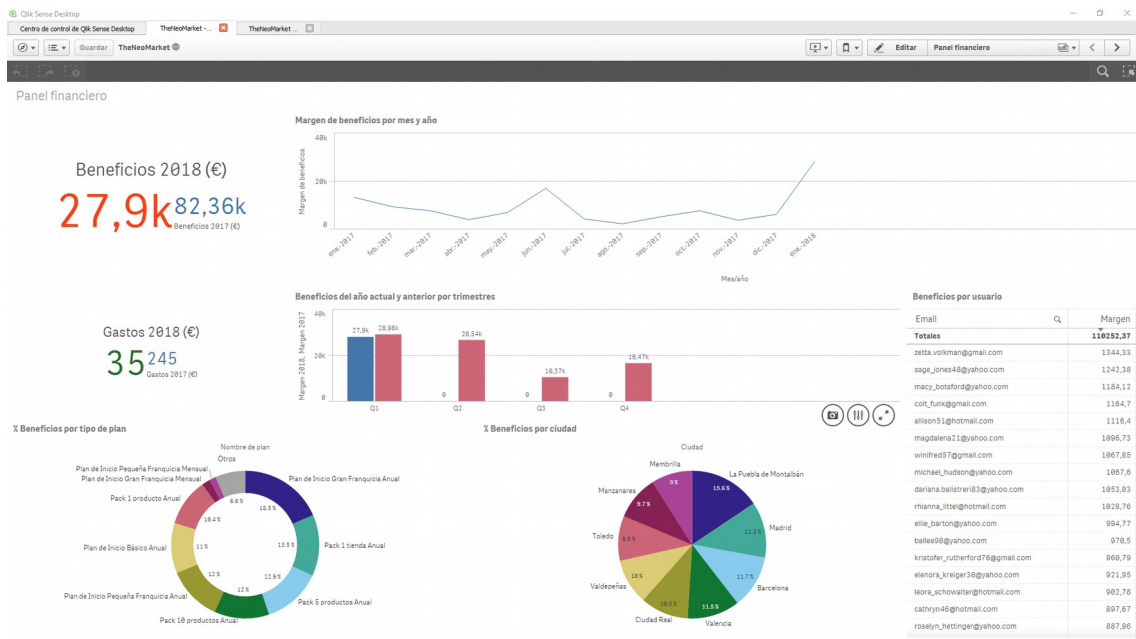
las tablas afectadas estén relacionadas en el modelo de datos). Esto se puede evitar por código. Estas restricciones (si las hay) se indican en las observaciones.

3.2.1.- Panel Financiero.

El panel financiero agrupa todas las visualizaciones de los indicadores relacionados con los beneficios y los gastos que genera la aplicación.

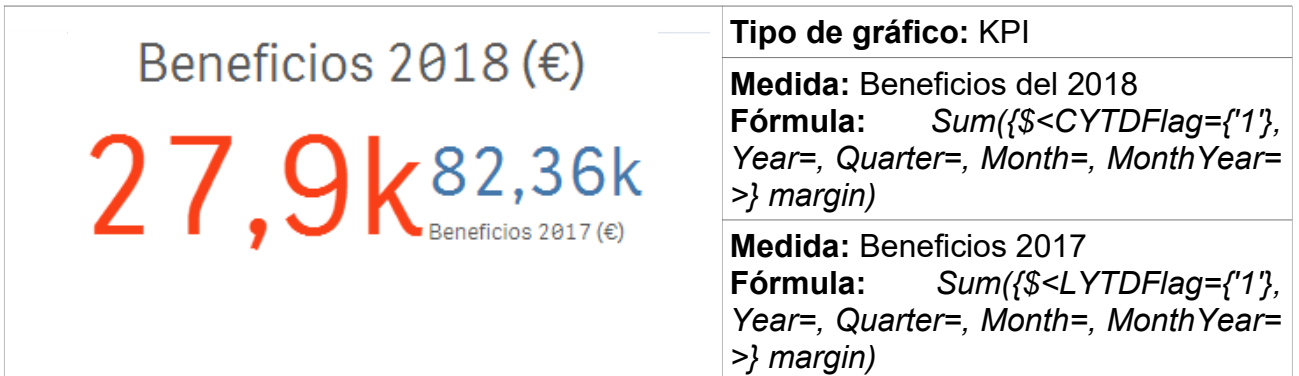
El objetivo de este panel es tener conocimiento del origen de las fuentes de los ingresos (por zona, por producto, por fecha), de esta manera detectar posibles riesgos a tiempo y poder tomar decisiones en consecuencia.

Para presentar esos beneficios se ha tenido en cuenta el factor tiempo, la fuente y la localización. El resultado es el siguiente:



A continuación se describe en detalle cada visualización que lo compone.

- **Beneficios del año en curso vs Beneficios del año anterior:** En este indicador se muestra la acumulación del Margen de beneficio del año en curso en contraste del margen de beneficio del año anterior.



Observaciones:

Si los beneficios del año en curso son menores a los beneficios del año anterior estos se muestran en color rojo, por contra, si son mayores estos se mostraran en verde. Para ello se ha añadido un limite (propiedades del gráfico --> Aspecto --> color --> Añadir limite) con la siguiente formula:

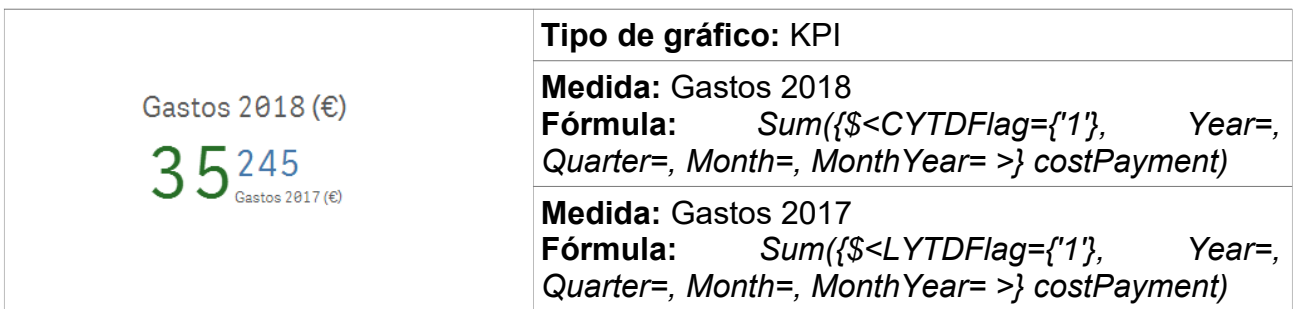
$=Sum(\{\$ < LYTDFlag = \{ ' 1 ' \}, Year =, Quarter =, Month =, MonthYear = > \} margin)$ y los colores rojo para los valores inferiores y verde para los superiores.

Si, por ejemplo, se filtran los datos por el plan "pack 10 productos mensual" del gráfico % Beneficios por tipo de plan se observa esta característica, puesto que para este plan no han habido beneficios en el año anterior:

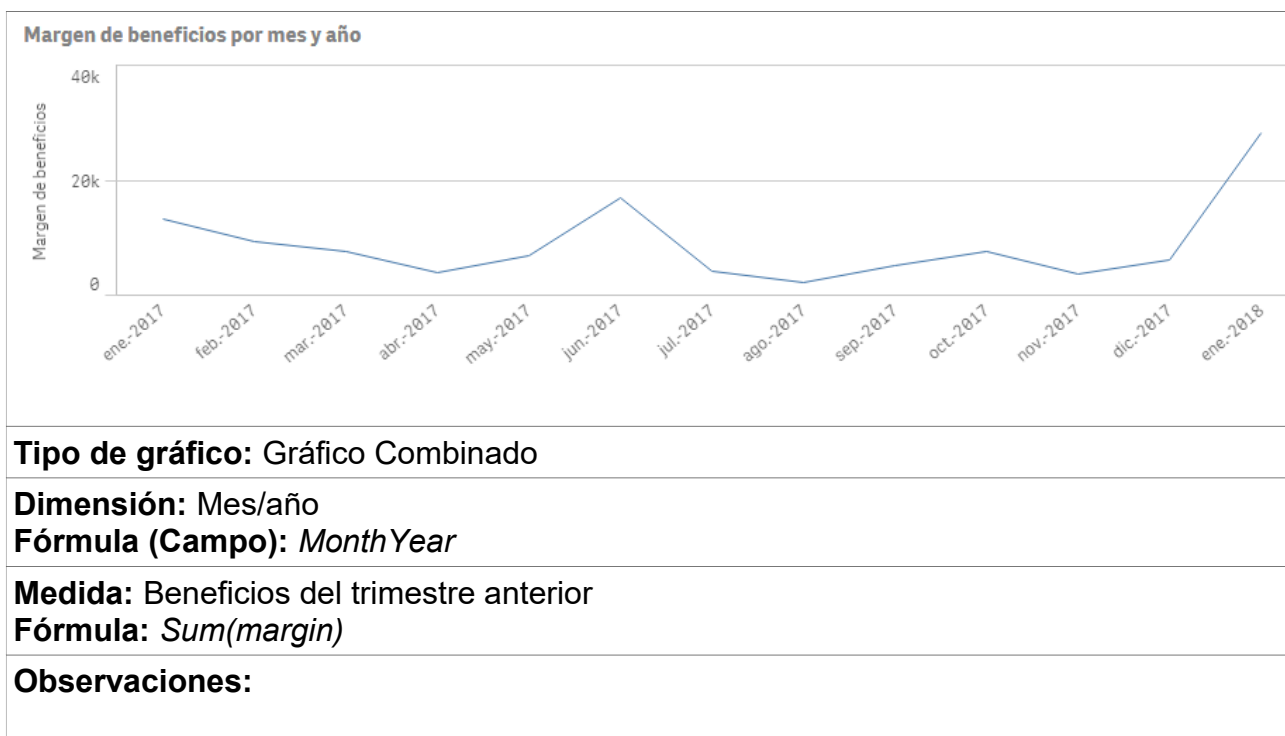


Este kpi reacciona a todos los filtros aplicados en la pantalla (tal y como lo hacen todos los gráficos en qlik), menos a los relacionados a las fechas, esto es debido a la fórmula.

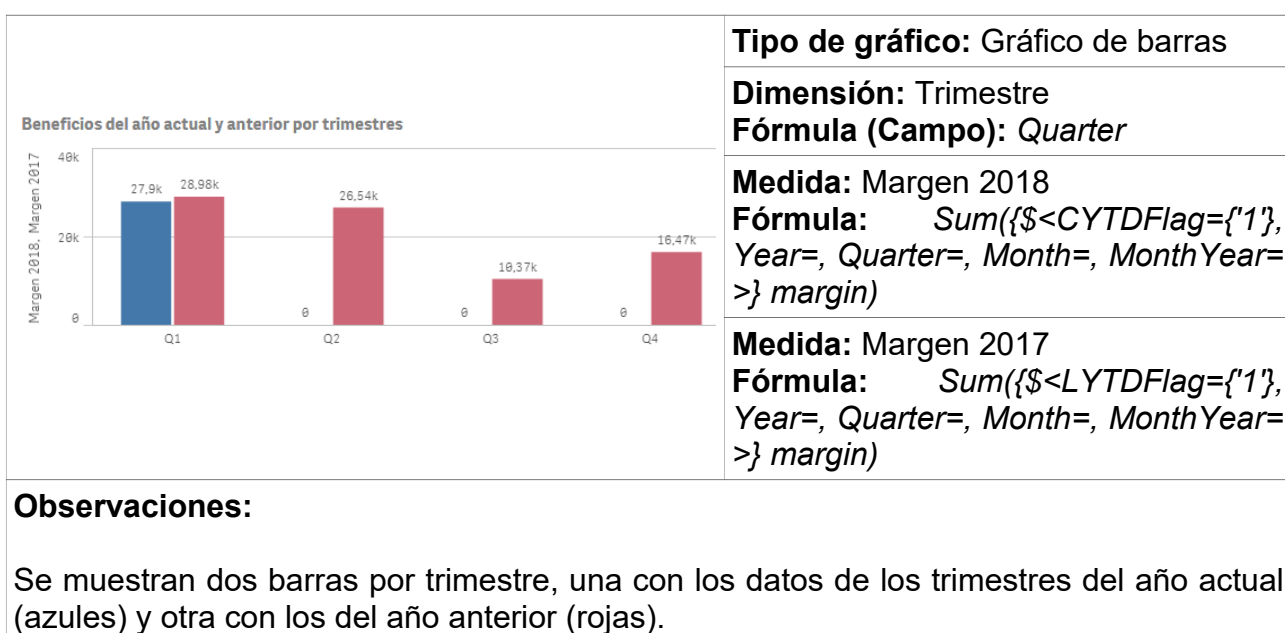
- **Gastos:** Este gráfico es igual que el anterior pero con los costes de la aplicación.



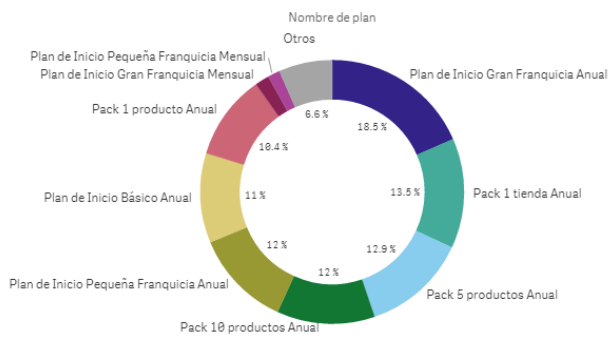
- **Margen de beneficios por mes y año:** En este gráfico se muestran los beneficios de la aplicación agrupados por el mes/año.



- **Beneficios del año actual y el anterior por trimestre:** En este gráfico de barras se muestran los beneficios por trimestre tanto del año actual como los del año anterior.



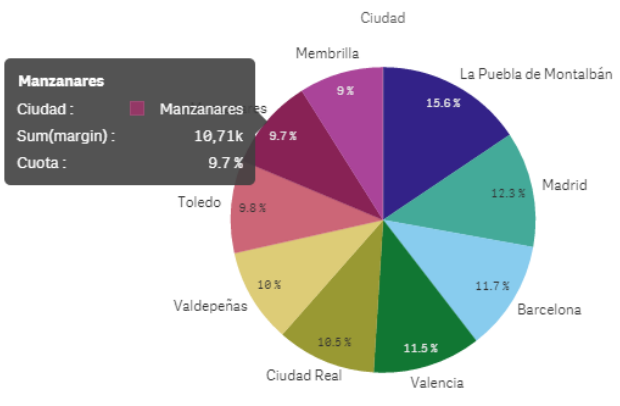
- **% Beneficios por tipo de plan:** En este gráfico de tarta se muestran los beneficios totales repartidos por el producto (tipo de plan).

<p>% Beneficios por tipo de plan</p> 	<p>Tipo de gráfico: Gráfico de tarta</p> <p>Dimensión: Nombre de plan</p> <p>Fórmula (Campo): <i>PlansName</i></p> <p>Medida: Margen</p> <p>Fórmula: <i>Sum(margin)</i></p>
---	---

Observaciones:

En las opciones de visualización se ha seleccionado la opción de "Donut" simplemente por una cuestión estética.

- **% de Beneficios por ciudad:** En este gráfico de tarta se muestran los beneficios totales repartidos por ciudades.

<p>% Beneficios por ciudad</p>  <p>Manzanares Ciudad : Sum(margin) : 10,71k Cuota : 9.7 %</p>	<p>Tipo de gráfico: Gráfico de tarta</p> <p>Dimensión: Ciudad</p> <p>Fórmula (Campo): <i>PlansName</i></p> <p>Medida: Margen</p> <p>Fórmula: <i>Sum(margin)</i></p>
---	---

Observaciones:

Este gráfico es interesante en la primera fase de la aplicación. Una vez la aplicación esté funcionando e implantada en un gran número de ciudades, sería interesante cambiar el filtro a provincia y añadir una gráfico de bloques para la ciudades.

- **Beneficios por usuario:** En esta tabla se muestran el usuario (mail) y su margen de beneficios aportado.

Beneficios por usuario		Tipo de gráfico: Tabla
Email	Q	Columna: Email Fórmula (Campo): <i>SellerEmail</i>
Totales		Columna: Margen Fórmula: <i>Sum(margin)</i>
zetta.volkman@gmail.com		
sage_jones48@yahoo.com		
macy_botsford@yahoo.com		
colt_funk@gmail.com		
allison51@hotmail.com		
magdalena21@yahoo.com		
winifred57@gmail.com		

Observaciones:

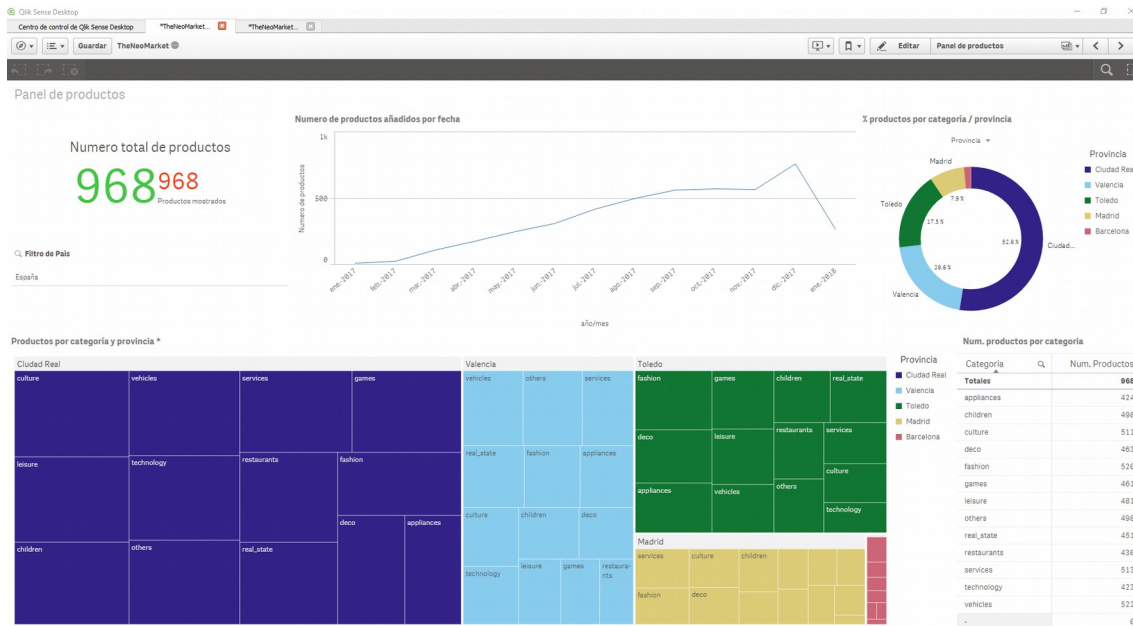
Esta tabla está pensada principalmente para operar con las selecciones de tipo de plan y zona, de las visualizaciones mostradas en el mismo panel. Así, se podrán visualizar los rankings absolutos de los usuarios que más han aportado (usuarios VIP) por ciudad, zona o incluso fecha.

3.2.2.- Panel de productos.

En el panel de productos se agrupan todas las visualizaciones relacionadas con los productos que son añadidos en la aplicación.

El propósito de este panel es conocer como se reparte la información y tener un conocimiento aproximado de la cantidad de información que dispone la aplicación. También conocer cuáles son las tendencias de venta (categorías de productos) por zonas y por tiempo.

El panel resultante es el siguiente:

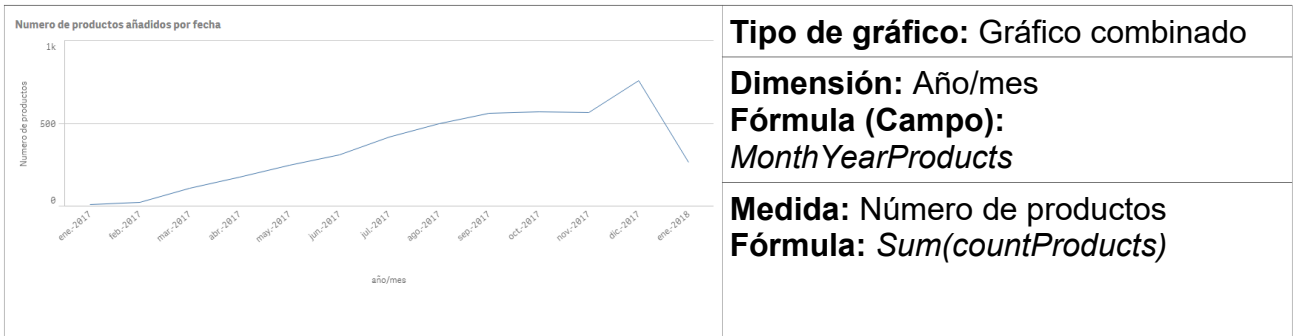


A continuación se describe en detalle cada visualización que lo compone.

- Número total de productos vs productos mostrados:** Este indicador muestra el número total de productos que hay en la aplicación y cuántos de ellos se están mostrando (pues los productos que están en la trastienda no son mostrados).

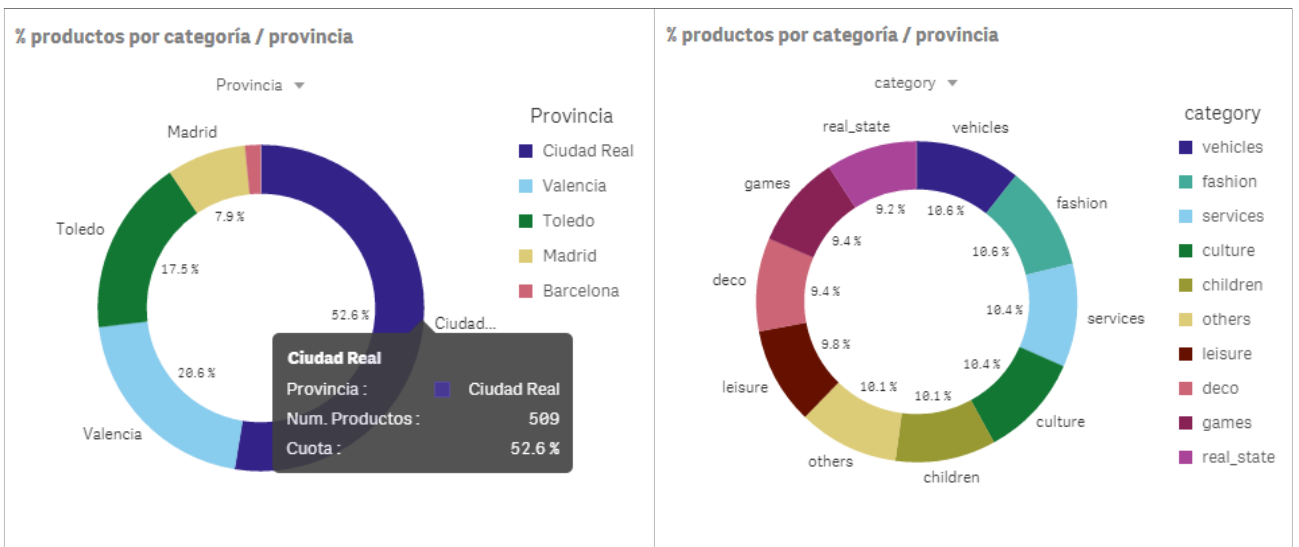
<p>Numero total de productos</p> <p>968 968 Productos mostrados</p>	<p>Tipo de gráfico: KPI</p> <p>Medida: Número total de productos Fórmula: $Sum(countProducts)$</p> <p>Medida: Productos mostrados Fórmula: $sum(\\${backdoor} = '0'} countProducts)$</p>
<p>Observaciones:</p> <p>Este gráfico está pensado principalmente para mostrar en la parte más visible de la pantalla el número total de productos y, que al usar los filtros de la pantalla, tener este dato como primera referencia.</p>	

- **Número de productos añadidos por fecha:** Este gráfico muestra el número de productos que se han añadido a la aplicación por año/mes.



Observaciones: Este gráfico se ha agrupado por mes/año pensando sobretodo para el principio, pues no habrá una gran cantidad de actividad, pero en un futuro se podría cambiar para mostrar los datos por semana o incluso por día.

- **% de productos por categoría / provincia:** En este este gráfico se muestra como están repartidos los productos por territorio o por categoría de producto, dependiendo de la selección de la pestaña situada arriba gráfico.



Tipo de gráfico: Gráfico de tarta

Dimensión: Provincia

Fórmula (Campo): state

Dimensión alternativa: Categoría

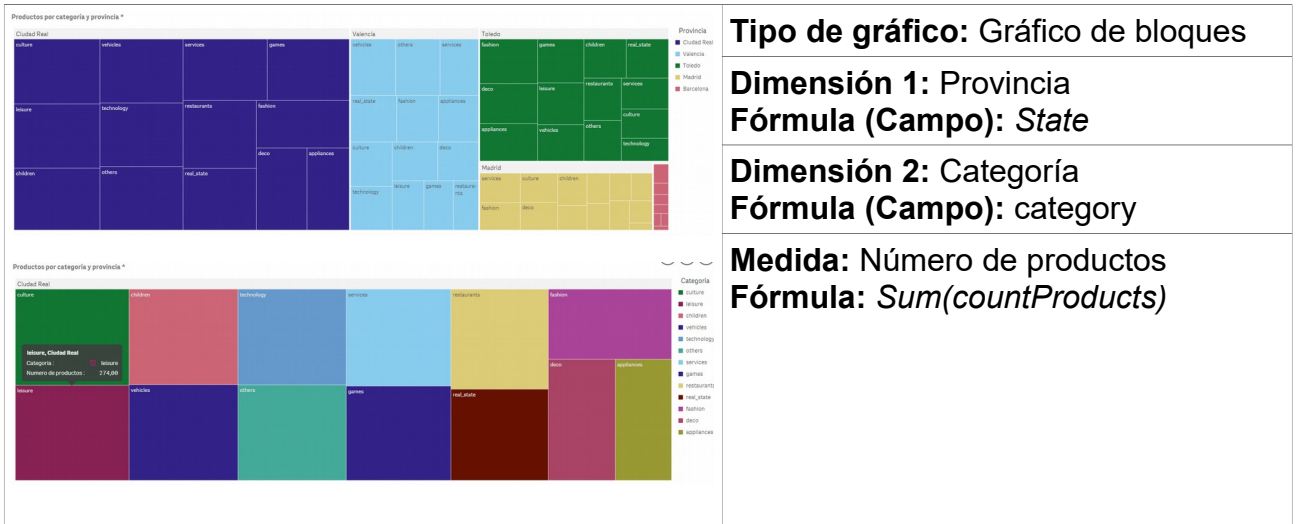
Fórmula (Campo): *category*

Medida: Número de productos

Fórmula: *Sum(countProducts)*

Observaciones: Este gráfico agrupa dos visualizaciones, una que muestra el porcentaje de distribución de productos por zonas y en la otra se muestra el porcentaje de distribución de productos por categoría.

- **Productos por categoría y provincia:** En este gráfico de bloques se muestra de un vistazo cómo están repartidos los productos entre las zonas y las categorías.



Observaciones:

Este gráfico es ideal para conocer rápidamente cuáles son las zonas con más productos y qué categorías son las que predominan en éstas, sin tener que realizar ningún filtrado. Una vez se aplican filtros (ya sean del propio gráfico o externos) se logra información más detallada y/o focalizada.

- **Número de productos por categoría:** En esta tabla se muestra el número total de productos por categoría, ordenado descendientemente por categoría. De esta manera se tiene el dato numérico directamente sin necesidad de entrar en los otros gráficos.

Num. productos por categoría	
Categoría	Num. Productos
Totales	509
appliances	172
children	264
culture	277
deco	206
fashion	219
games	250
leisure	274

Tipo de gráfico: Tabla

Columna: Categoría
Fórmula (Campo): category

Columna: Número de productos
Fórmula: $Sum(countProducts)$

Observaciones:

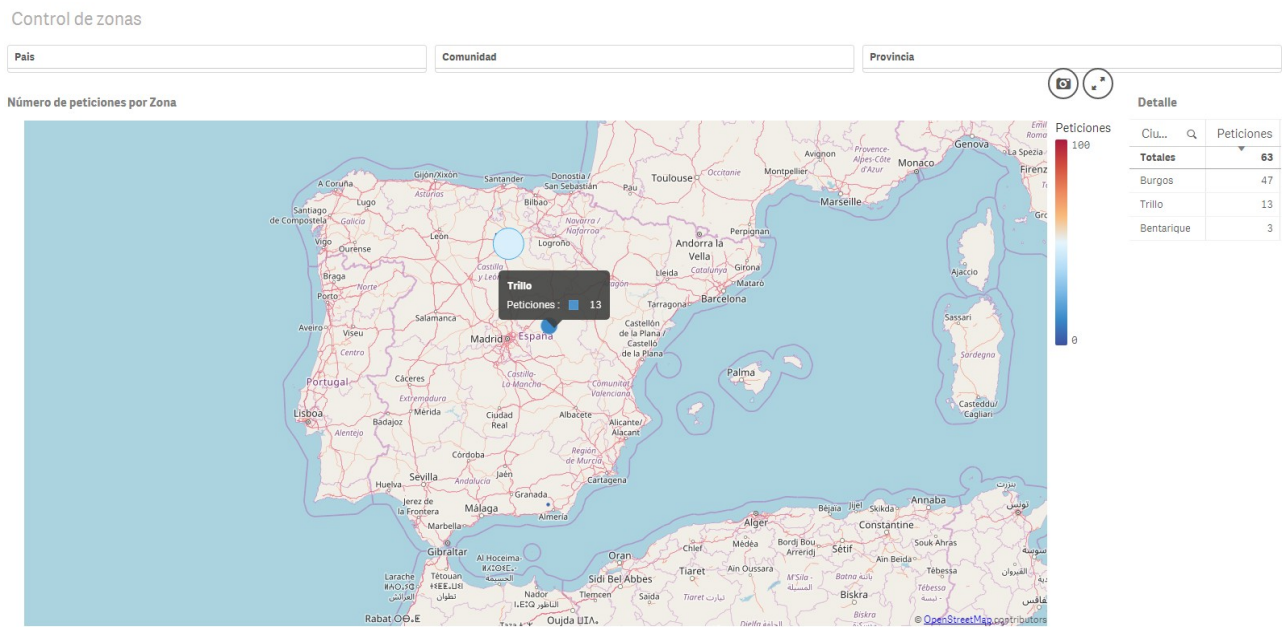
- **Filtro por país:** Este gráfico es un panel de filtrado para poder filtrar por país.

<p>🔍 Filtro de País</p> <p>España</p>	<p>Tipo de gráfico: Panel de filtrado</p> <p>Dimensión: País</p> <p>Fórmula (Campo): <i>País</i></p>
<p>Observaciones:</p> <p>En una primera fase la aplicación saldrá para España, pero la intención es que salga pronto para países cercanos.</p>	

3.2.3.- Control de zonas.

Este panel contiene las visualizaciones relacionadas con las solicitudes de apertura, con el objetivo de estudiar las posibles nuevas zonas en las que operar.

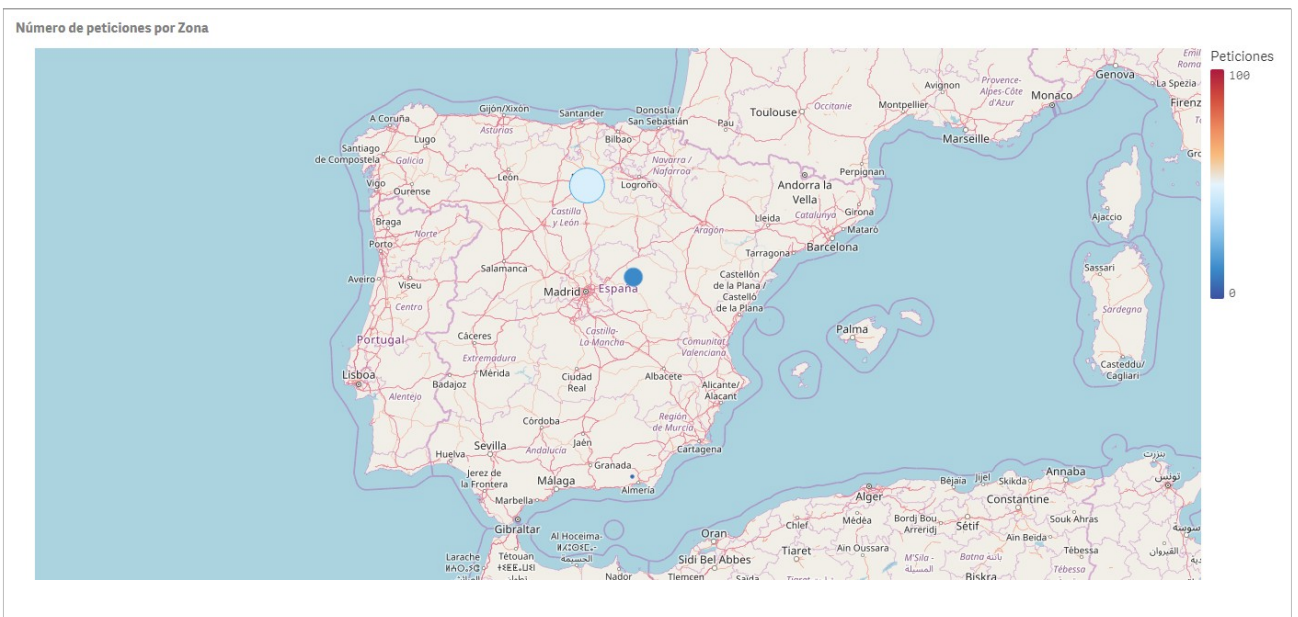
El resultado es el siguiente:



A continuación se describe en detalle cada visualización que lo compone.

Nota: Los filtros aplicados en este gráfico no afectan a las demás visualizaciones ya que atacan a la tabla ZoneRequest, la cual no está relacionada con ninguna de las demás tablas del modelo.

- **Numero de peticiones por zonas:** En este gráfico se muestra un mapa sobre el cual se indica qué zona es la que está recibiendo peticiones y cuántas de éstas se están realizando.



Tipo de gráfico: Mapa

Medida: Peticiones

Fórmula: $\text{sum}([\text{ZoneRequest.countRequest}])$

Dimensión: ciudad

Campo: ZoneRequest.city

URL del mapa: <http://tile.openstreetmap.org/{z}/{x}/{y}.png>

Atribución: © OpenStreetMap contributors

Observaciones:

- Los puntos mostrados tienen un color más cálido según más peticiones tengan, para ello el color se aplica por medida, aplicando la siguiente fórmula $\text{sum}([\text{ZoneRequest.countRequest}])$ y seleccionando un rango de colores. También los puntos se muestran mayores según el valor de la medida, pero esto es una opción automática.
- El mapa es navegable y se puede realizar zoom hasta el nivel de calles.

- **Detalle:** Este gráfico es simplemente una tabla con las zonas y su número de peticiones ordenado descendientemente. Su principal función es la de mostrar los datos de una manera más directa, pero también se puede usar como filtro y para navegar por el mapa.

Detalle	Tipo de gráfico: Tabla										
<table border="1"> <thead> <tr> <th>Ciu...</th> <th>Peticiones</th> </tr> </thead> <tbody> <tr> <td>Totales</td> <td>63</td> </tr> <tr> <td>Burgos</td> <td>47</td> </tr> <tr> <td>Trillo</td> <td>13</td> </tr> <tr> <td>Bentarique</td> <td>3</td> </tr> </tbody> </table>	Ciu...	Peticiones	Totales	63	Burgos	47	Trillo	13	Bentarique	3	Columna: Ciudad Fórmula (Campo): <i>ZoneRequest.city</i> Columna: Peticiones Fórmula: <i>Sum([ZoneRequest.countRequest])</i>
Ciu...	Peticiones										
Totales	63										
Burgos	47										
Trillo	13										
Bentarique	3										
Observaciones:											

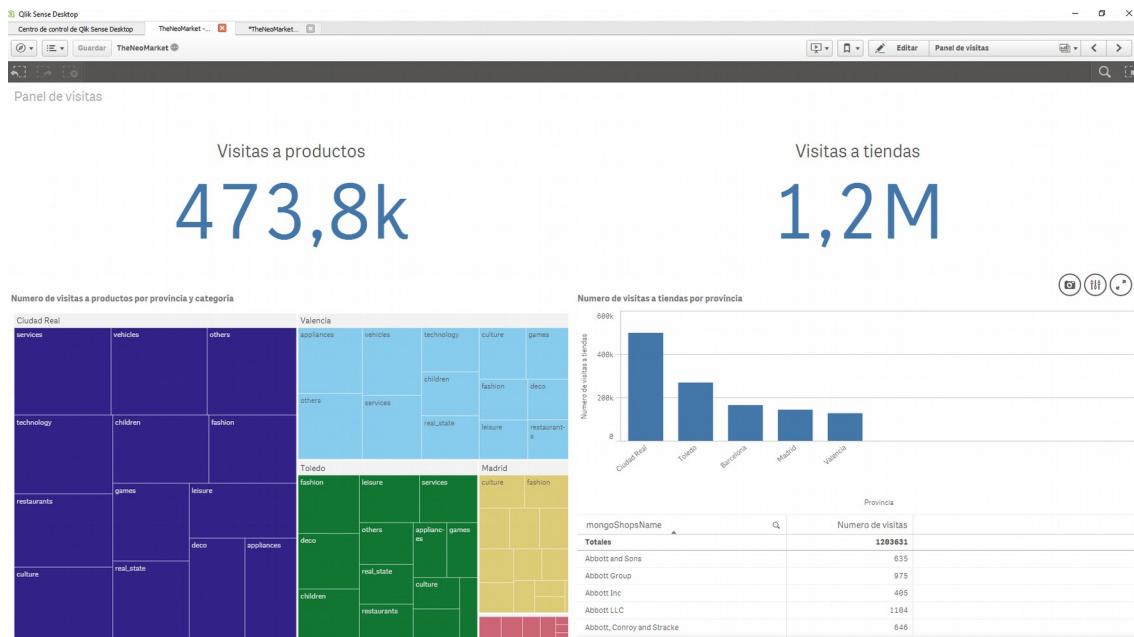
- **Paneles de filtrado:** Se han añadido varios paneles de filtrado al panel, para hacerlo más usable. Los campos para el filtrado son *ZoneRequest.country*, Comunidad y *ZoneRequest.state*.

3.2.4.- Panel de visitas:

En este panel se agrupan todas las visualizaciones relacionadas con las visitas que realizan los compradores a los productos y a las tiendas, para así tener un conocimiento de los intereses de los compradores y poder hacer un análisis de tendencias.

Nota: La funcionalidad de las visitas no está implementada en la aplicación, únicamente se cuenta con los campos que las han de recoger. Debido a ello, los datos con los que se cuentan han sido introducidos a mano.

El resultado es el siguiente:



A continuación se describe en detalle cada visualización que lo compone.

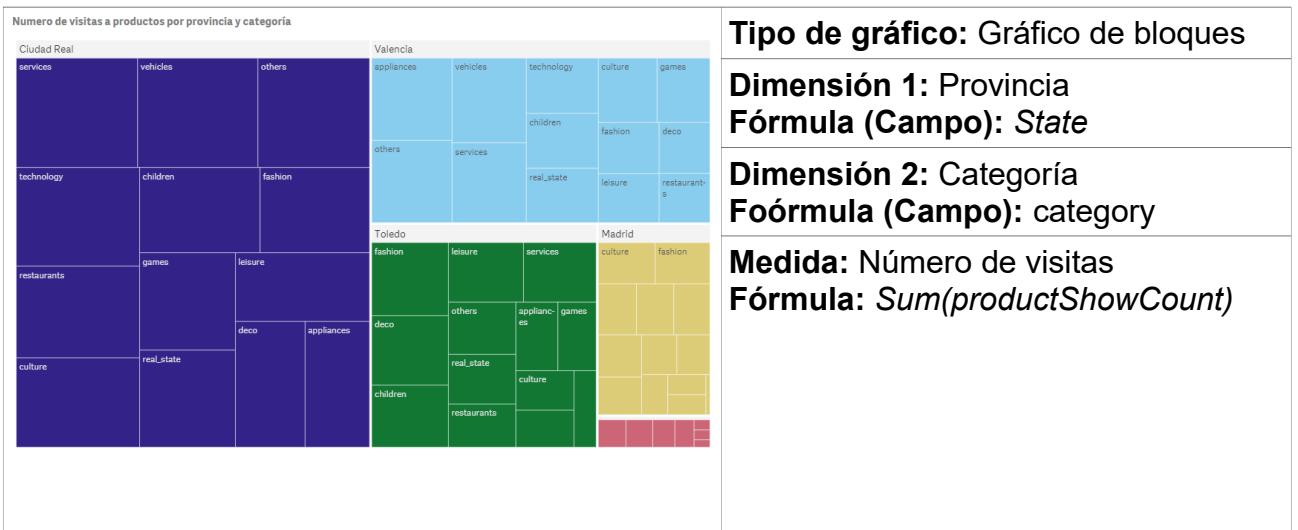
- **Visitas a productos:** Muestra el número total de visitas a productos.

<p>Visitas a productos</p> <h1>473,8k</h1>	<p>Tipo de gráfico: KPI</p> <p>Medida: Visitas a productos</p> <p>Fórmula: $Sum(productShowCount)$</p>
<p>Observaciones:</p>	

- **Visitas a productos:** Muestra el número total de visitas a productos.

<p>Visitas a tiendas</p> <h1>1,2M</h1>	<p>Tipo de gráfico: KPI</p> <p>Medida: Visitas a tiendas</p> <p>Fórmula: $Sum(shopShowCount)$</p>
<p>Observaciones:</p>	

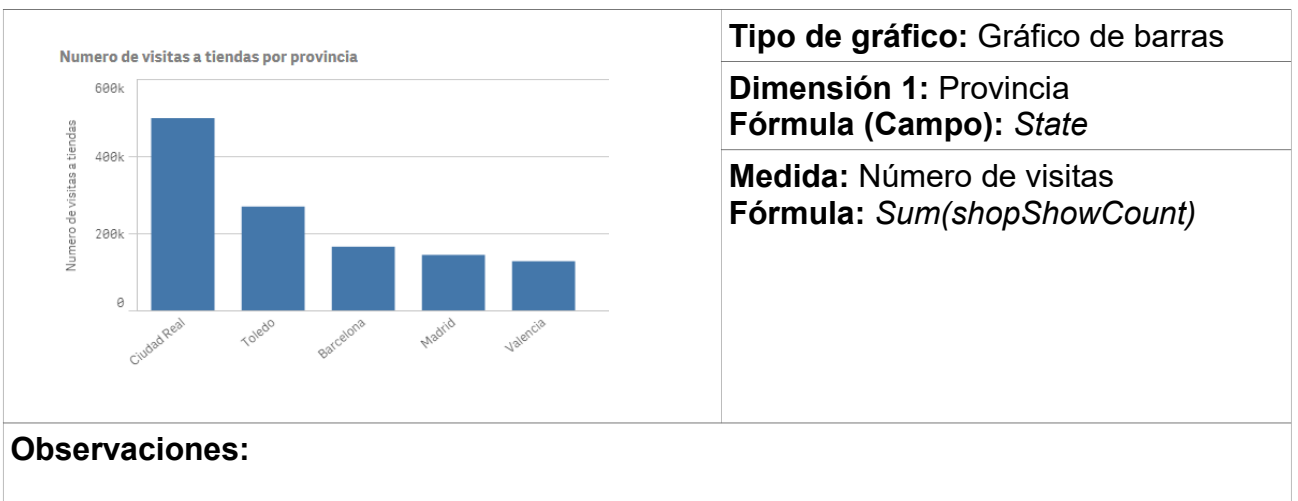
- **Número de visitas a productos por provincia y categoría:** En este gráfico de bloques se muestra de un vistazo cómo están repartidas las visitas a los productos entre las zonas y las categorías de la aplicación.



Observaciones:

Este gráfico es ideal para conocer rápidamente cuáles son las zonas con más visitas a productos, y qué categorías son la que predominan en éstas sin tener que realizar ningún filtrado. Una vez se aplican filtros (ya sean del propio gráfico o externos) se logra información más detallada y/o focalizada.

- **Número de visitas a tiendas por provincia:** En este gráfico de barras muestra cómo están repartidas las visitas a las tiendas entre las zonas de la aplicación.



Observaciones:

- **Número de visitas por tienda:** Esta tabla muestra el número de visitas por tienda ordenado.

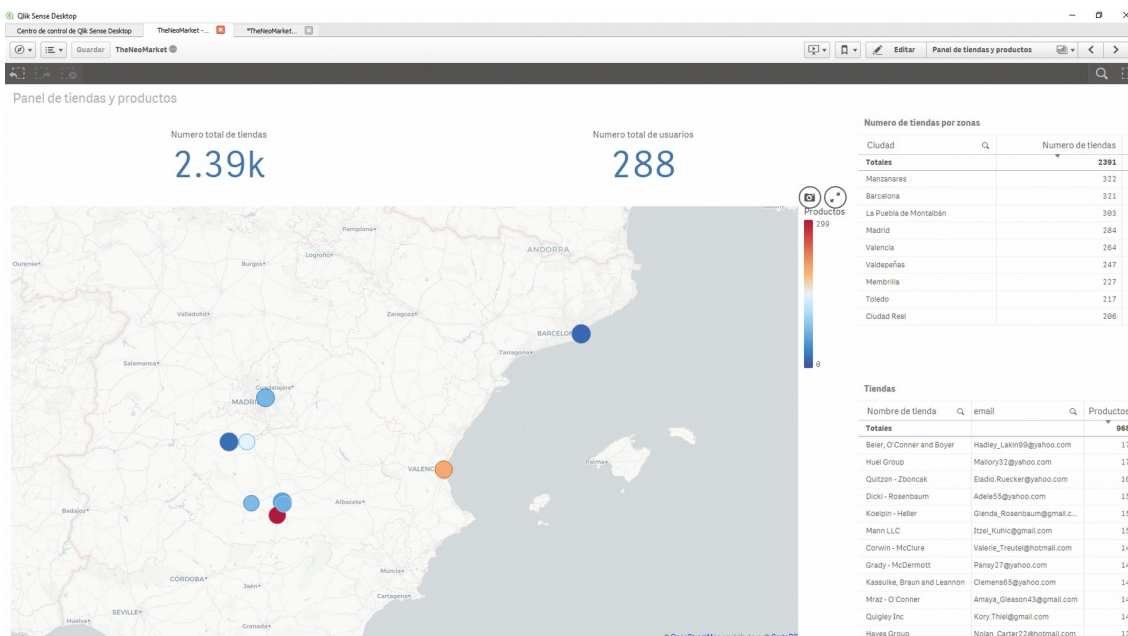
Numero de visitas por tienda		Tipo de gráfico: Tabla
Nombre de la tienda	Numero de visitas	Columna: Ciudad Fórmula (Campo): <i>ZoneRequest.city</i>
Totales	1203631	Columna: Peticiones Fórmula: <i>Sum([ZoneRequest.countRequest])</i>
Larson Group	2275	
Howell LLC	1888	
Runolfsson and Sons	1867	
Gislason Group	1836	

Observaciones: Está pensado sobre todo para usar con los filtros.

3.3.5.- Panel de tiendas y usuarios.

En este panel se agrupan los gráficos que están relacionados con las tiendas de la aplicación (número de tiendas, localizaciones, datos de la tienda).

El resultado es el siguiente:



A continuación se describe en detalle cada visualización que lo compone.

- **Número total de tiendas:** Muestra el número total de tiendas de la aplicación.

<p>Numero total de tiendas</p> <h1>2.39k</h1>	<p>Tipo de gráfico: KPI</p> <p>Medida: Número total de tiendas Fórmula: <i>Sum(shopsRecordCount)</i></p>
Observaciones:	

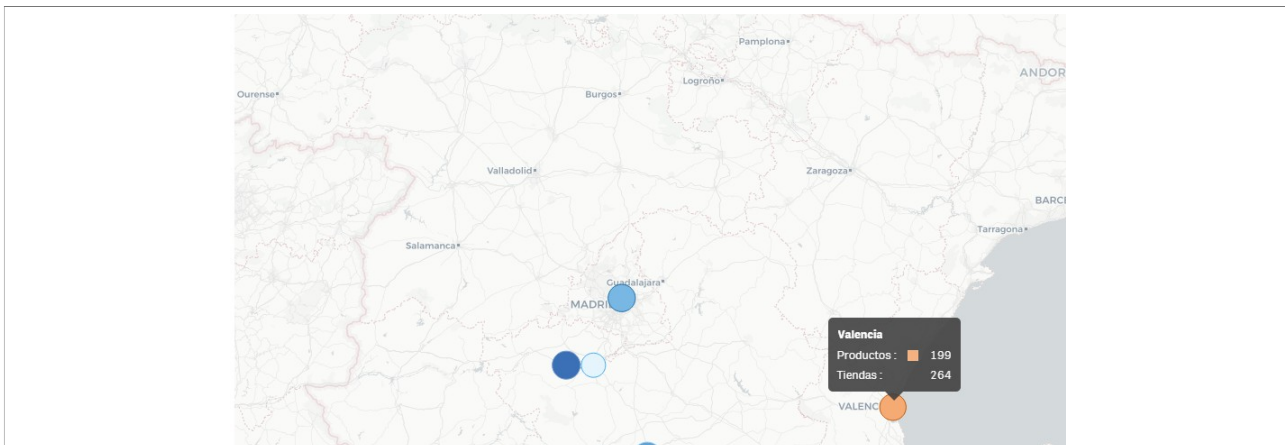
- **Número total de Usuarios:** Muestra el número total de usuarios de la aplicación.

<p>Numero total de usuarios</p> <h1>288</h1>	<p>Tipo de gráfico: KPI</p> <p>Medida: Número total de usuarios Fórmula: <i>Sum(CustomerRecordCounter)</i></p>
Observaciones:	

- **Número de tiendas por zonas:** Esta tabla muestra el número de tiendas que hay por zonas de manera descendente. Su principal función es la de mostrar los datos de una manera más directa, pero también se puede usar como filtro y para navegar por el mapa.

<table border="1"> <thead> <tr> <th colspan="2">Numero de tiendas por zonas</th> </tr> <tr> <th>Ciudad</th> <th>Numero de tiendas</th> </tr> </thead> <tbody> <tr> <td>Totales</td> <td>2391</td> </tr> <tr> <td>Manzanares</td> <td>322</td> </tr> <tr> <td>Barcelona</td> <td>321</td> </tr> <tr> <td>La Puebla de Montalbán</td> <td>303</td> </tr> <tr> <td>Madrid</td> <td>284</td> </tr> <tr> <td>Valencia</td> <td>264</td> </tr> <tr> <td>Valdepeñas</td> <td>247</td> </tr> <tr> <td>Membrilla</td> <td>227</td> </tr> <tr> <td>Toledo</td> <td>217</td> </tr> <tr> <td>Ciudad Real</td> <td>206</td> </tr> </tbody> </table>	Numero de tiendas por zonas		Ciudad	Numero de tiendas	Totales	2391	Manzanares	322	Barcelona	321	La Puebla de Montalbán	303	Madrid	284	Valencia	264	Valdepeñas	247	Membrilla	227	Toledo	217	Ciudad Real	206	<p>Tipo de gráfico: Tabla</p> <p>Columna: Ciudad Fórmula (Campo): <i>zoneCity</i></p> <p>Columna: Número de tiendas Fórmula: <i>Sum(shopsRecordCount)</i></p>
Numero de tiendas por zonas																									
Ciudad	Numero de tiendas																								
Totales	2391																								
Manzanares	322																								
Barcelona	321																								
La Puebla de Montalbán	303																								
Madrid	284																								
Valencia	264																								
Valdepeñas	247																								
Membrilla	227																								
Toledo	217																								
Ciudad Real	206																								
Observaciones:																									

- **Mapa de tiendas por Zona:** Muestra en un mapa las zonas en las que la aplicación está abierta, y cuántas tiendas y productos hay en ellas.



Tipo de gráfico: Mapa

Medida: Tiendas

Fórmula: $sum(shopsRecordCount)$

Dimensión: ciudad

Campo: city

URL del mapa: <http://tile.openstreetmap.org/{z}/{x}/{y}.png>

Atribución: © OpenStreetMap contributors

Observaciones:

- Los puntos mostrados tienen un color más cálido según más productos tengan, para ello el color se aplica por medida, aplicando la siguiente fórmula $sum(countProducts)$, y seleccionando un rango de colores. Además, los puntos se muestran más grandes según la medida indicada en el mapa (número de tiendas).
- El mapa es navegable y se puede realizar zoom hasta el nivel de calles.

- **Tiendas:** Esta tabla muestra las tiendas de manera individual y los productos que tienen cada una de estas. La tabla está ordenada de manera descendente por número de tiendas.

Tiendas			Tipo de gráfico: Tabla
Nombre de tienda	Q	email	Q
Productos			
Totales			968
Beier, O'Conner and Boyer		Hadley_Lakin99@yahoo.com	17
Huel Group		Mallory32@yahoo.com	17
Quitzon - Zboncak		Eladio.Ruecker@yahoo.com	16
Dicki - Rosenbaum		Adele55@yahoo.com	15
Koelpin - Heller		Glenda_Rosenbaum@gmail.c...	15
Mann LLC		Itzel_Kuhic@gmail.com	15
Corwin - McClure		Valerie_Treutel@hotmail.com	14
Grady - McDermott		Pansy27@yahoo.com	14

Columna: Nombre de la tienda	Fórmula (Campo): <i>mongoShopsName</i>
Columna: Email	Fórmula (Campo): <i>Email</i>
Columna: Productos	Fórmula: <i>Sum(countProducts)</i>

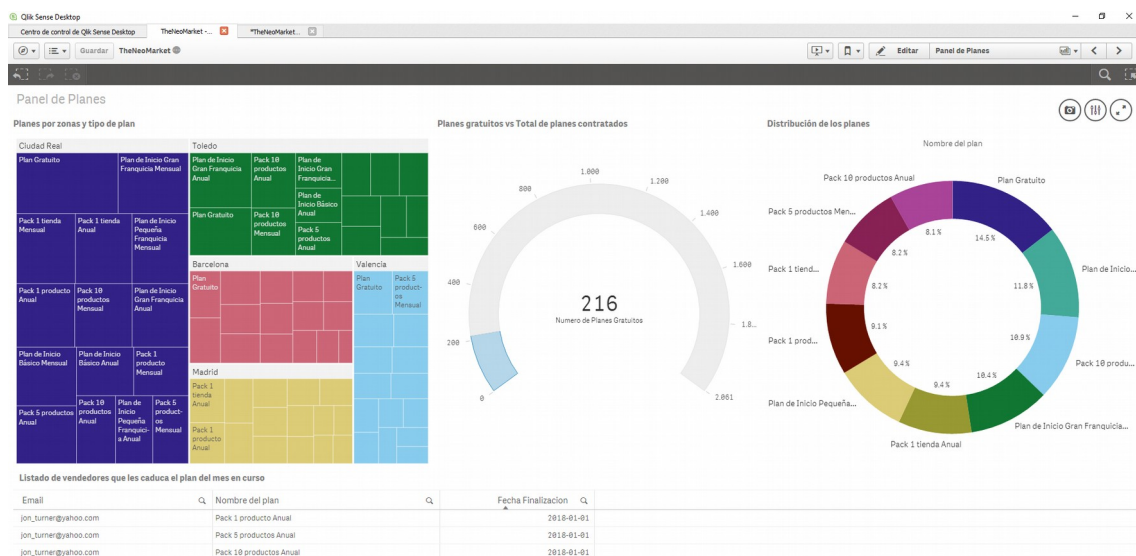
Observaciones: La tabla esta pensada para operar con los filtros de zona.

3.2.6.- Panel de planes.

En este panel se han agrupado todos los gráficos relacionados con los planes. Está pensado para conocer cuáles son los planes predominantes, y cómo se organizan éstos, así como cuál es la proporción de planes de pago con respecto a los gratuitos.

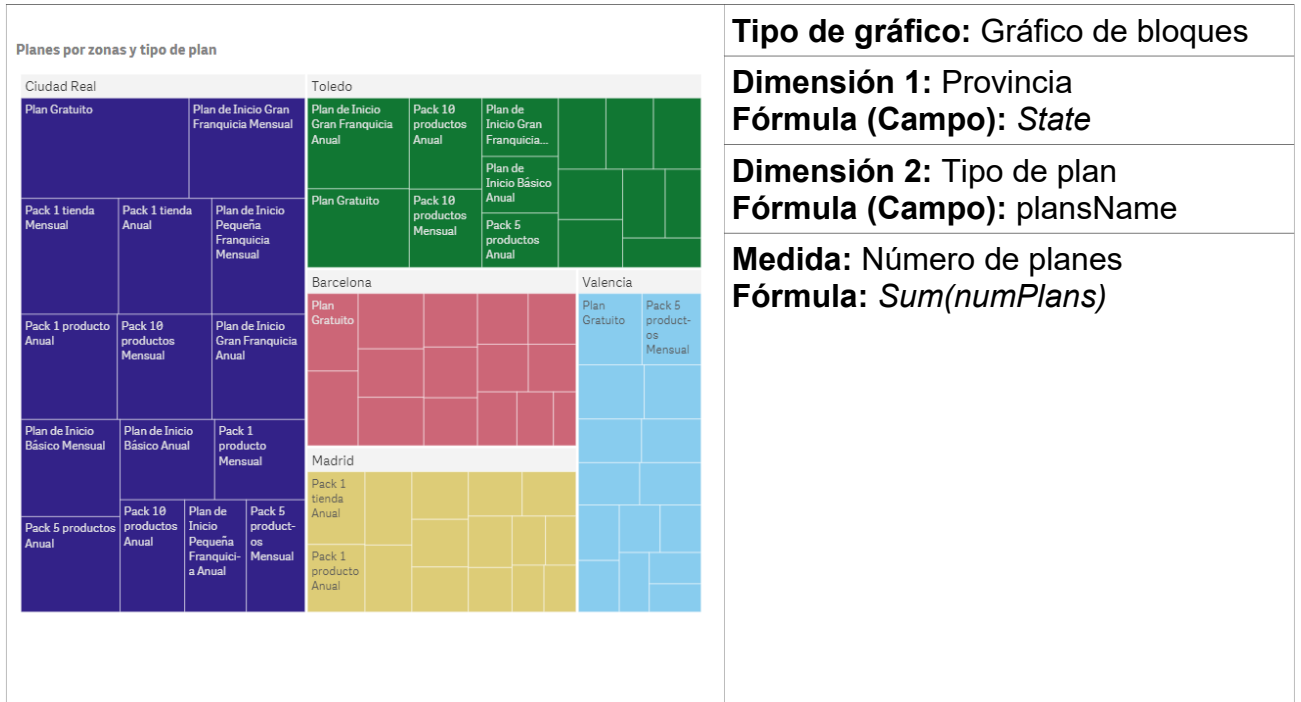
También muestra a qué usuarios les caduca la suscripción al plan en el mes en curso, con el fin de poder tomar decisiones al respecto.

El resultado es el siguiente:



A continuación se describe en detalle cada visualización que lo compone.

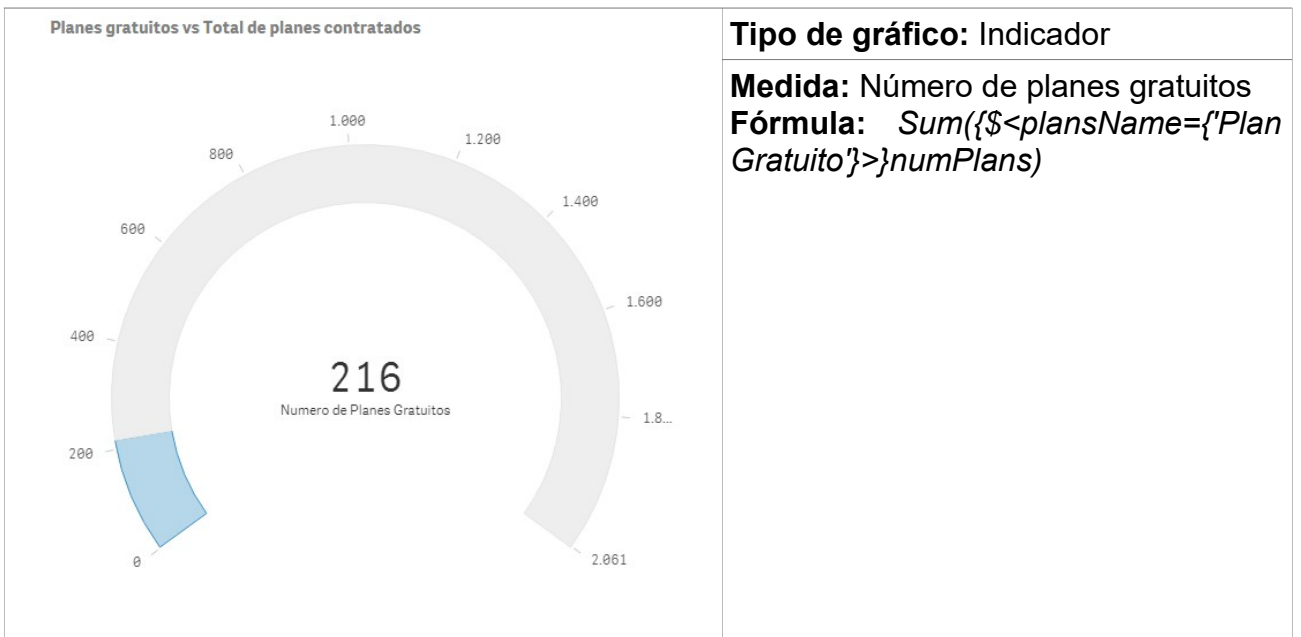
- **Planes por zonas y tipo de plan:** En este gráfico se muestran cuáles son los tipos de planes predominantes por zonas



Observaciones:

Este gráfico es ideal para conocer rápidamente cuáles son los planes que más predominan por zonas, y de esta manera, ver cómo están funcionando y poder tomar decisiones acordes.

- **Planes gratuitos vs total de planes contratados:** Este gráfico contrasta cuántos planes gratuitos hay en la aplicación frente a los de pago.

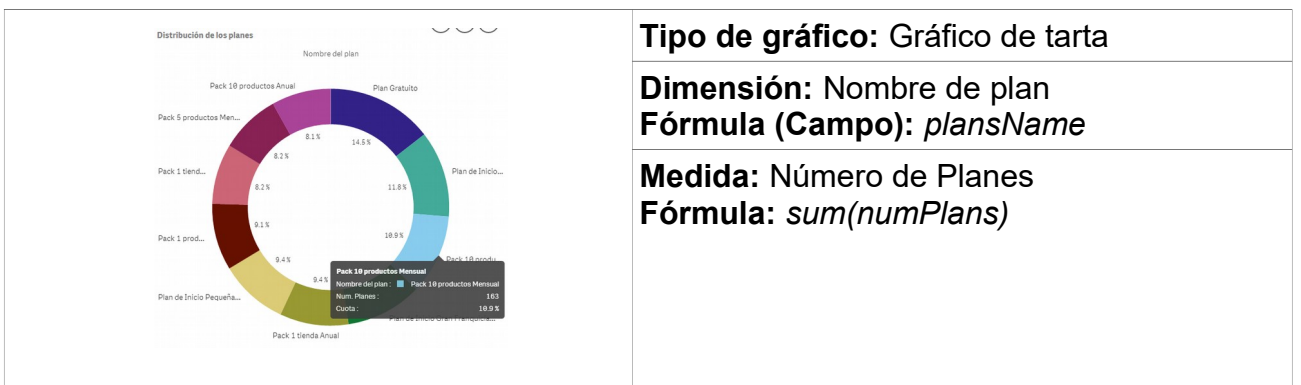


Observaciones:

El rango del indicador ha sido fijado con el siguiente límite: 0 para el límite inferior y el número total de planes $Sum(\{\$<plansName=>numPlans)$ para el límite superior, de esta forma se puede observar la proporción que hay.

A este gráfico no le afectan los filtrados por tipo de plan.

- **Distribución de los planes:** En este gráfico muestra la distribución total de los planes contratados que hay en la aplicación.



Observaciones:

Es interesante combinarlo con las fechas (de otros paneles o del cuadro de filtrado) para verlo por fechas.

- **Listado de vendedores que les caduca el plan del mes en curso:**
Este gráfico se trata de un listado en el que se recogen los planes que caducan en el mes en curso y el usuario que lo ha contratado, y así poder tomar decisiones comerciales y de marketing/publicidad.

Listado de vendedores que les caduca el plan del mes en curso			
Email	Nombre del plan	Fecha Finalizacion	
sergiodfdez@gmail.com	Pack 5 productos Mensual	2018-01-13 18:27	
lawela@gmail.com	Pack 5 productos Anual	2018-01-13 18:27	

Tomar captura

Abrir librería de capturas

Exportar como imagen

Exportar a PDF

Exportar datos

Tipo de gráfico: Tabla

Columna: Nombre del plan
Fórmula (Campo): *SellerEmail*

Columna: Email
Fórmula (Campo): *plansName*

Columna: Fecha Finalizacion
Fórmula: *=if(MonthYearEnd = Date(MonthStart(Today()),'MMM-YYYY'), [Ended At (UTC)], null())*

Observaciones: Este listado es interesante exportarlo como datos (excel) o como pdf para poder hacer uso de él de una manera más cómoda, en la imagen se muestran las opciones de exportado.

Nota: Actualmente no se están almacenando en las bases de datos de la aplicación los datos personales de los cliente (como se puede observar en los anexos, en el apartado de recopilación de información) por lo que sólo se muestra el mail del mismo. Una vez se añadan los datos personales de los usuarios, seria interesante añadirlos a este listado.

4. Producto final.

Terminada la implementación, el resultado es el archivo de Qlik Sense TheNeoMarket.qvf, el cual contiene tanto el cuadro de mando integral y el script de carga (que se han comentado en la fase de implementación).

Para poder hacer uso de él hay dos caminos, usando Qlik Sense Desktop ,para el cual es necesario la instalación del software, o usando la herramienta web Qlik Sense Cloud. A continuación se describen los dos caminos en detalle.

Tanto para Desktop como para Cloud el primer paso es registrarse en la página de Qlik, para ello hay que ir al siguiente link:

<https://www.qlik.com/es-es/try-or-buy/download-qlik-sense>

- Qlik Sense Desktop:

Una vez Registrado, hay que loguearse en la página de Qlik e ir de nuevo al link, o entrar desde la zona de descarga de Qlik Sense:

<https://www.qlik.com/es-es/try-or-buy?ga-link=bjp-gs-hero>

Descargar el archivo e instalarlo.

Con el software instalado, hay que copiar el .qvf en la carpeta:

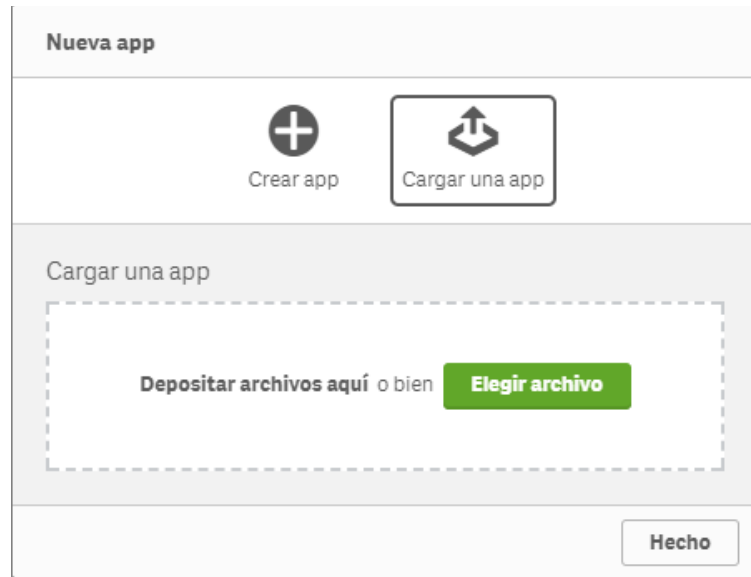
C:\Users\[nombre de usuario]\Documents\Qlik\Sense\Apps

- Qlik Sense Cloud:

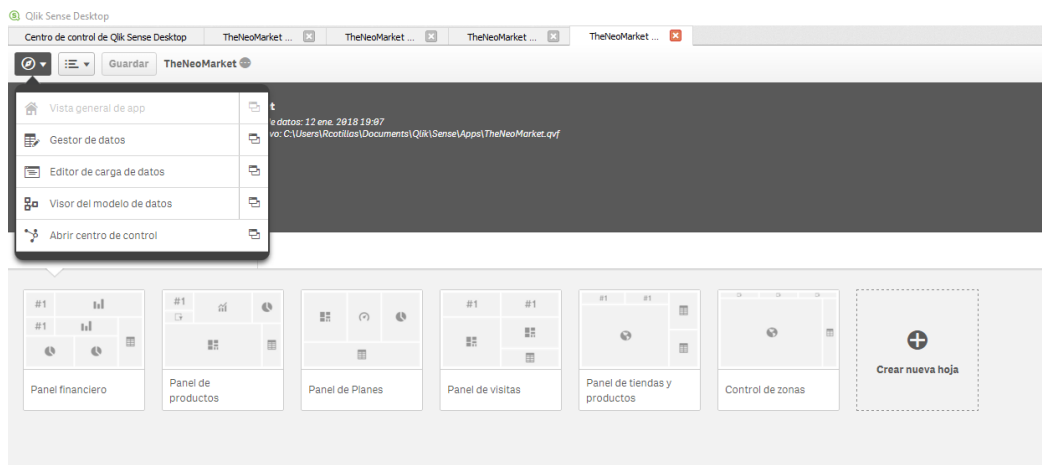
Qlik Sense Cloud se ejecuta desde el siguiente link:

<https://eu.qlikcloud.com/>

Este link lleva a una pantalla de registro en la que, una vez logueado, se redirige al usuario a la pantalla “vista general de app”, pero esta no contendrá ninguna app. Para cargar TheNeoMarker.qvf hay que pulsar el boton “Nueva app” (situado arriba a la izquierda), seleccionar “cargar una app” y arrastrar el archivo al área indicada.



Una vez se han hecho estos pasos, al abrir Qlik Sense Desktop o Cloud, en el centro de control de Qlik Sense Desktop (pagina inicial), aparecerá una carpeta con el nombre del archivo cargado “TheNeoMarket”, que al pulsar sobre él se abrirá la vista general de app:



En la cual, al pulsar en cualquier hoja, se podrá navegar al *Dashboard* deseado.

Y si se quiere ver el modelo o el script, en el botón de la brújula situado en la parte superior izquierda, solo hay que seleccionar “Editor de carga de datos” para el script y “visor de modelo de datos” para el modelo.

5. Conclusiones.

Con la realización de este trabajo de fin de grado no sólo se ha logrado implantar una solución B.I. en Qlik Sense, necesaria para la toma de decisiones y la gestión de la información generada en TheNeoMarket; sino que además se le ha dotado al usuario de una herramienta, fácil de utilizar, en la que con muy poca formación éste pueda customizarse sus propias visualizaciones.

Cabe destacar el grado de satisfacción del cliente con la solución entregada, ya que el principal objetivo de esta startup es el poder conseguir inversores, y esta solución B.I. permitirá presentar los resultados de una manera profesional. Pero no solamente se ha entregado una aplicación para mostrar resultados, se ha conseguido una herramienta útil para la toma de decisiones en todo el espectro del negocio.

Analizando la planificación, y aunque al final se ha cumplido con los objetivos, no ha sido posible cumplir con los plazos de la planificación inicialmente propuesta, retrasándose las entregas parciales e incluso no llegando a lo planificado originalmente. Esto ha sido debido a diferentes motivos personales que se han ido dando durante el proceso; en un principio se contaba con una licencia de Qlik Sense Server y de los conectores oficiales, pero un cambio de trabajo y de ciudad, no sólo obligó a replantear las herramientas, sino que si ya el tiempo era ajustado, todavía lo fue más, obligándome a realizar la mayor parte del proyecto en la última fase de éste.

Otro problema con el que me he encontrado durante el desarrollo de este proyecto ha sido el hecho de que la aplicación de TheNeoMarker no hubiera estado terminada. Esto ha implicado que se haya tenido que lidiar con varios problemas, ya que todos los datos se han tenido que generar mediante scripts y manualmente, que había funcionalidades que no estaban (y no están) terminadas, o que tenían errores. Sin embargo y por otro lado, también ha servido para detectar faltas funcionales y correcciones de la aplicación sobre la que se trabajaba, por lo que ambos proyectos se han retroalimentado.

Como líneas futuras, hay que destacar dos vertientes, la primera es de desarrollo y mejora, y la segunda es la migración a la versión de pago Qlik Sense Server.

Como mejora, cabe destacar el análisis de tendencias, interesaría meter una dimensión tiempo y lugar a las visitas, de este modo se podrían realizar visualizaciones que tengan una mayor precisión. También hay que destacar que cuando la aplicación lleve un tiempo funcionando convendría buscar maneras de conectar Stripe directamente con Qlik Sense, ya que cuando la aplicación comience a crecer la descarga de datos de Stripe podría llevar demasiado tiempo.

En resumen, en este trabajo se ha visto los pasos de creación de una herramienta BI en Qlik Sense, con la cual la StartUp TheNeoMarket tendrá mayores opciones de sacar su proyecto adelante.

6. Bibliografía.

1. <https://es.wikipedia.org/wiki/MongoDB>. 28-10-2017.
2. <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/qu-son-los-procesos-etl>. 01-12-2017
3. <http://help.qlik.com/es-ES/qlikview/12.1/Subsystems/Client/Content/Scripting/synthetic-keys.htm>. 27.12-2017
4. Data Modeling for Qlik Sense. QlikTech International. April 2016 Release, Qlik Sense v2.2
5. Create Visualizations with Qlik Sense. QlikTech International. April 2016 Release, Qlik Sense v2.2
6. <https://www.nicolasmarchal.com/marketing/kpis-que-es-como-ayudan-y-ejemplos/> . 19-11-2017
7. [https://en.wikipedia.org/wiki/Stripe_\(company\)](https://en.wikipedia.org/wiki/Stripe_(company)). 28-10-2017.
8. <http://searchoracle.techtarget.com/definition/MySQL-Connector-ODBC-Connector-ODBC-or-MyODBC>. 27-11-2017
9. <https://www.simba.com/news/magnitude-software-acquires-simba/>. 27-11-2017
10. <https://help.qlik.com/es-ES/sense/1.1/pdf/Conceptos%20de%20Qlik%20Sense.pdf>. 17-11-2017.
11. <http://www.elmundofinanciero.com/noticia/31920/empresas/qlik-presenta-qlik-sense-nueva-generacion-de-aplicacion-de-visualizacion-y-descubrimiento-de-datos.html>. 17-11-2017.
12. <https://mine-class.com/qliksense/>. 17-11-2017.
13. https://es.wikipedia.org/wiki/Cuadro_de_mando_integral. 27-11-2017.
14. <https://www.nicolasmarchal.com/marketing/kpis-que-es-como-ayudan-y-ejemplos/>. 03-12-2017
15. <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/65065/7/ehernandezfeTFG0617mem%C3%B2ria.pdf>. 09-12-2017
16. <https://www.businessintelligence.info/varios/longitud-latitud-pueblos-espana.html>. 27-12-2017
17. <http://danielgarciamejia.com/qlik/qlik-sense/porque-qlik-sense/>. 11-10-2016.

7. Anexos

I.- Recopilación de datos de las tablas.

En este apartado se ha recogido la información recopilada de las reuniones con el cliente respecto a las fuentes de datos.

Tabla Admins:

Campo	Descripción
_id	Número interno identificador de la fila.
email	Email del administrador (usuario con el que se conecta).
password	Password de conexión del administrador.
__v	Versión de la tabla.

Tabla AgendaJobs

Campo	Descripción
_id	Número interno identificador de la fila.
name	Nombre de la tarea.
type	Tipo de tarea.
data	Fecha de la tarea.
priority	Prioridad de la tarea.
repeatInterval	Intervalo de repetición.
repeatTimezone	Indicador de comienzo en una hora concreta.
LastModifiedBy	Última modificación.
nextRunAt	Fecha de comienzo de la próxima tarea.
LastFinishedAt	Fecha en la que finalizó la última tarea.

Tabla Plans.

Campo	Descripción
_id	Número interno identificador de la fila.
name	Nombre del plan.
cost	Columna donde se indica el coste del plan.
Modifiers.shops	Número de tiendas que ofrece el plan.
Modifiers.products	Número de productos que ofrece el plan.

TempLimit.period	Unidad de tiempo para la duración del plan (years, months,days,etc).
TempLimit.Count	Número de unidad.
complement	Campo opcional. Indica qué tipo de complemento es, en el caso de que sea de tipo complemento.
type	Indica el tipo de plan (inicio, complemento,Test).
size	Número de elementos que tiene el plan.
__v	Versión de la tabla.

Tabla products.

Campo	Descripción
_id	Número interno identificador de la fila.
updateDate	Fecha en la que actualiza el producto.
sellerId	Identificador del vendedor.
name	Nombre del producto.
cost	Coste del producto (aplicando descuentos).
previousCost	Coste original del producto.
category	Categoría principal en la que se engloba el producto.
description	Texto de descripción del producto.
shop_id	Identificador de la tienda.
markForDelete	Indicador de borrado del producto.
backdoor	Indicador de si el producto se encuentra en la trastienda.
starDate	Fecha en la que el producto es agregado en el sistema
subCategory	Subcategorías del producto.
show_count	Número de veces que el producto ha sido visitado.
photos	Fotografías del producto.
IsOffer	Indicador de si el producto es una oferta o no.
__v	Versión de la tabla.

Tabla sellers

Campo	Descripción
_id	Identificador de los vendedores.
password	Password de acceso a la aplicación para los vendedores.
email	Email del vendedor.

googleId	Identificador para la plataforma con de google.
Stripe.customerId	Identificador del usuario del Stripe.
Stripe.sources	Tarjetas que tiene guardadas del usuario.
products.backdoor.total	Total de productos que el vendedor puede guardar en la trastienda.
products.backdoor.count	Total de productos guardados en la trastienda.
products.markForDelete	Número de productos marcados para ser eliminados.
products.total	Número total de productos.
products.count	Número total de productos a mostrar.
shops.backdoor.total	Número total de tiendas que puede almacenar la trastienda de tiendas.
shops.backdoor.count	Número de tiendas que hay en la trastienda.
shops.markForDelete	Número de tiendas marcados para ser eliminados.
shops.total	Número total de tiendas a usar.
shops.count	Número total de tiendas.
plans	Referencias a la tabla de usersplanes, array de Ids.
__v	

Tabla Shoppers

Campo	Descripción
_id	Identificador de la fila de...
Password	Pasword de identificación en la aplicación para los compradores.
email	Email del comprador, es usado como usuario para identificarse en la aplicación.
googleId	Código Oauth de google para identificarse en la aplicación.
__v	

Tabla shops

Campo	Descripción
_id	Identificador tienda.
name	Nombre de la tienda.
phone	Número de teléfono de la tienda.
email	Email de la tienda.
street	Dirección de la tienda.
number	Número de la dirección de la tienda .
country	País donde se encuentra la tienda.
state	Provincia donde se encuentra la tienda.
city	Ciudad donde se encuentra la tienda.
sellerID	Identificador del vendedor propietario de la tienda.
markForDelete	Indicador de fila borrada (o a borrar).
backdoor	Indicador de si la tienda se encuentra en la trastienda.
show_count	Número de visitas que ha tenido la tienda.
countProducts	Número de productos que tiene la tienda en la actualidad.
photos	Fotos de la tienda.
loc.cords.lat	Latitud donde está geolocalizada la tienda.
loc.cords.long	Longitud donde está geolocalizada la tienda.
Loc.zoom	Zoom a aplicar en el mapa para mostrar la tienda.
startDate	Fecha en la que la tienda es creada.

Tabla UserPlans

Campo	Descripción
_id	Identificador de registro de la tabla.
finishDate	Fecha en la que termina el plan.
plan	Referencia del plan que tiene el usuario.
sellerId	Clave del vendedor.
modifiers.products	Número de productos que tiene el plan (opcional).
modifiers.shop	Número de tiendas que tiene el plan (opcional).
tempLimit.period	Unidad de tiempo del periodo.
tempLimit.count	Tiempo que dura el plan (referencia rápida).
startdate	Fecha en la que comienza el plan.

Tabla Zonas

Campo	Descripción
_id	Identificador de la zona.
state	Provincia de la zona.
city	Ciudad de la zona.
country	País de la zona.
coords.lat	Latitud de la geolocalización del centro de la zona.
coords.long	Longitud de la geolocalización del centro de la zona.
radius	Radio que engloba la zona.
__v	

Tabla ZoneReques

Campo	Descripción
_id	Identificador de la zona.
state	Provincia de la zona.
city	Ciudad de la zona.
country	País de la zona.
coords.lat	Latitud de la geolocalización del centro de la zona.
coords.long	Longitud de la geolocalización del centro de la zona.
Email	Email del usuario que ha realizado la petición.
__v	

Stripe (plataforma de pago)

Tabla payments

Campo	Descripción
id	Identificador del pago.
Description	Descripción del pago.
Created (UTC)	Fecha y hora en la que se realizó el pago formato (YYYY-MM-DD MM:SS).
Amount	Cantidad del pago.

Amount(refund)	Cantidad reintegrada.
Currency	Moneda del pago.
Coverted Amount	Pago convertido a la moneda por defecto.
Coverted Amount refunded	Cantidad reintegrada en la moneda por defecto.
Fee	Cuantía del pago que se queda la plataforma.
Tax	Tasas.
Coverted currency	Moneda por defecto.
Mode	Modo en el que está el pago (actualmente esta en modo Text por que la aplicación no está en activo).
Status	Estado el pago (puede ser pending o paid).
Statement Description	???
Customer ID	Identificador del cliente.
Customer Description	Descripción del cliente.
Customer Email	Email del cliente.
Captured	???
Card ID	Identificador de la tarjeta.
Card Last4	Últimos 4 dígitos de la tarjeta.
Card Brand	Marca de la tarjeta.
Card Founding	Tipo de tarjeta (crédito, débito...).
Card Exp Month	Mes de caducidad de la tarjeta.
Card Exp year	Año de caducidad de la tarjeta.
Card Name	Nombre del titular de la tarjeta.
Card Address Line1	Dirección de la tarjeta 1.
Card Address Line2	Dirección de la tarjeta 2.
Card Address city	Ciudad de la tarjeta.
Card Address	Provincia de la tarjeta.

State	
Card Address Country	País de la tarjeta.
Card Address Zip	Código postal de la tarjeta.
Card Issue country	???
Card Fingerprint	???
Card CVC Status	???
Card AVS Zip Status	???
Card AVS Line1 Status	???
Card Tokenization Method	???
Disputed Amount	???
Dispute Status	???
Dispute Reason	???
Dispute date (UTC)	???
Dispute Evidence Due (UTC)	???
Invoice ID	Identificador de la factura.
Payment Source Type	Tipo de pago.
Destination	Destino del pago.
Transfer	Transferencia.
Interchange Costs	Costes del intercambio.
Merchant Service Charge	Cargo por servio comercial.
Transfer group	Grupo de transferencia.

Order_id (metadata)	???
User_id (metadata)	???

Customers

Campo	Descripción
Id	Identificador del vendedor.
Description	Descripción.
Email	Email del comprador.
Created (UTC)	Fecha de creación del comprador formato (YYYY-MM-DD).
Delinquent	???
Card ID	Identificador de la tarjeta por defecto.
Card Las4	Últimos 4 dígitos de la tarjeta por defecto.
Card Brand	Marca de la tarjeta por defecto.
Card Funding	Tipo de tarjeta (crédito, débito...).
Card Exp Month	Mes de caducidad de la tarjeta por defecto.
Card Exp Year	Año de caducidad de la tarjeta por defecto.
Card Name	Nombre de la tarjeta.
Card Address Line1	Dirección de la tarjeta por defecto línea 1.
Card Address Line2	Dirección de la tarjeta por defecto línea 2.
Card Address City	Dirección de la tarjeta por defecto ciudad.
Card Address State	Dirección de la tarjeta por defecto comunidad.
Card Address Country	Dirección de la tarjeta por defecto país.
Card Address zip	Dirección de la tarjeta por defecto código postal.

Card Issue Contry	???
Card Fingerprint	???
Card CVC Status	???
Card AVS Zip Status	???
Card Tokenization Method	???
Plan	Identificador del plan.
Status	Estatus del plan (active, inactive).
Cancel At Period End	Indicador de si el plan ha de cancelarse cuando termine su periodo.
Acount Balance	Balance de ganancias por el plan.
Currency	Moneda.

Tabla Invoices

Campo	Descripción
Id	Identificador de la factura.
Amount Due	Cantidad adeudada en céntimos.
Application Fee	Tasa de solicitud.
Billing	Tipo de facturación.
Charge	Identificador del pago.
Closed	Indicador de cierre.
Currency	Moneda.
Customer	Identificador del comprador.
Date (UTC)	Fecha de factura formato (YYYY-MM-DD).
Discount	Descuento.
Due Date (UTC)	Fecha de vencimiento de la factura.
Ending Balance	Balance final.
Forgiven	

Number	Número de factura .
Paid	Indicador de pagado.
Period End (UTC)	Fecha de finalización de la factura.
Period Start (UTC)	Periodo .
Starting Balance	
Subscription	Identificador de la suscripción.
Subtotal	Cuantía subtotal.
Tax	Tasas.
Tax percent	Porcentaje de las tasas.
Total	Cuantía total.

Tabla Plans

Campo	Descripción
Id	Identificador del plan.
Name	Nombre del plan.
Created (UTC)	Fecha de creación del plan .
Interval	Tipo de intervalo (month, year...).
Amount	Cuantía (en céntimos).
Currency	Tipo de moneda.
Interval Count	???
Trial Period Days	Días de prueba.
Statement Descriptor	Descripción.

Tabla Subscriptions

Campo	Descripción
Id	Identificador de la suscripción.
Customer ID	Identificador del comprador.

Customer Description	Descripción del comprador.
Customer Email	Mail del comprador.
Plan	Plan del comprador.
Quantity	Cantidad.
Interval	Tipo de intervalo del la suscripción.
Amount	Importe.
Status	Indicador del estado de la suscripción (active o inactive).
Created (UTC)	Fecha de creación .
Start (UTC)	Fecha de comienzo.
Current Period Start (UTC)	Fecha de periodo actual comienzo.
Current Period End (UTC)	Fecha de periodo actual fin.
Trial Start (UTC)	Fecha de inicio del periodo de prueba formato (YYYY-MM-DD).
Trial End (UTC)	Fecha de fin del periodo de prueba.
Application Fee Percent	Porcentaje de cuota aplicada.
Coupon	Identificador de cupón de descuento.
Tax Percent	Porcentaje de tasas.
Canceled At (UTC)	Fecha de cancelación.
Cancel Period End At	Indicador de cancelación al final del periodo.
Ended At (UTC)	Fecha de finalización.
Period (metadata)	Periodo.

II.- Script de carga.

En este apartado se ha recogido el código del Script de carga. Está dividido en pestañas, por lo que cada apartado corresponde a la pestaña del script:

- Main:

```
SET ThousandSep='.';
SET DecimalSep=',';
SET MoneyThousandSep='.';
SET MoneyDecimalSep=',';
SET MoneyFormat='#.##0,00 €;-#.##0,00 €';
SET TimeFormat='h:mm:ss';
SET DateFormat='DD/MM/YYYY';
SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff]';
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=4;
SET FirstMonthOfYear=1;
SET CollationLocale='es-ES';
SET CreateSearchIndexOnReload=1;
SET
MonthNames='ene.;feb.;mar.;abr.;may.;jun.;jul.;ago.;sep.;oct.;nov.;dic
.';
SET
LongMonthNames='enero;febrero;marzo;abril;mayo;junio;julio;agosto;sep-
tiembre;octubre;noviembre;diciembre';
SET DayNames='lu.;ma.;mi.;ju.;vi.;sá.;do.';
SET LongDayNames='lunes;martes;miércoles;jueves;viernes;sábado;domin-
go';

//Conexion a la base de datos de desarrollo
LIB CONNECT TO 'Development';

//Conexion a la base de datos de desarrollo
// LIB CONNECT TO 'Production';

let today = Date(Today(),'YYYY-MM-DD');
let urlLocal = 'lib://QVD/';
```

- Lat/Long

```
//Solo se ha de pasar una vez, asteriscar en cargas posteriores

// Tabla con las latitudes y longitudes de los pueblos de españa
LatLongEspania:
LOAD
    Población & '|' & Provincia & '|' & 'España' as idZone,
    Comunidad,
    Latitud,
    Longitud,
    Habitantes
FROM [lib://Otras Fuentes/listado-longitud-latitud-municipios-
espana.xls]
(biff, embedded labels, header is 2 lines, table is Hoja1$);

Store LatLongEspania into $(urlLocal)LatLong.qvd (qvd);

Drop Table LatLongEspania;
```

- Paiments

```

//**** Load Paiments from Stripe
stripePayments:
Load *,
    "Converted Amount" - Fee - Tax - InterchangeCosts -
MerchantServiceCharge as margin;
LOAD
    id as paymentsID,
    date(Floor(num(date("Created (UTC)")), 'YYYY-MM-DD')) as
dataPayment,
    "Converted Amount",
//    "Converted Amount Refunded",
    Fee,
    Tax,
//    "Converted Currency",
    "Mode",
//    Status as paymentStatus,
//    "Customer ID" as customerID,
    "Invoice ID" as invoiceID,
    "Payment Source Type",
    if("Interchange Costs" = '', 0, "Interchange Costs") as
InterchangeCosts,
    if("Merchant Service Charge" = '', 0, "Merchant Service Charge") as
MerchantServiceCharge,
    1 AS PaymentRecordCounter
FROM [lib://Stripe/payments.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

// **** Load Invoices from Stripe
join (stripePayments)
LOAD
    id as invoiceID,
//    "Amount Due",
//    "Application Fee",
//    Closed,
//    Currency as invoiceCurrency,
//Customer as customerID,
    "Date (UTC)",
    Discount,
    Paid,
//"Period Start (UTC)",
    Subscription as subscriptionID,
    Subtotal,
    if(Tax = '', 0, Tax) as taxInvoice,
    if("Tax Percent" = '', 0, "Tax Percent") as taxInvoicePercent,
//Tax as taxInvoice,
//"Tax Percent" as taxPercentInvoice,
    "Total" as totalInvoice
FROM [lib://Stripe/invoices.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

// Find Min and Max order dates from orders table
MinMaxTemp:
LOAD
    MIN(dataPayment) AS MinDate,
    Today() as MaxDate

```

```

RESIDENT stripePayments;

LET vMinDate = Floor(NUM(PEEK('MinDate', 0, 'MinMaxTemp')));
LET vMaxDate =Floor(NUM(PEEK('MaxDate', 0, 'MinMaxTemp')));
LET vToday = $(vMaxDate);

Drop Table MinMaxTemp;

//***** Temporary Calendar *****
CalTemp:
LOAD
    date($(vMinDate) + rowno() - 1) AS TempDate
AUTOGENERATE
    $(vMaxDate) - $(vMinDate) + 1;

//Create the Master Calendar
//MasterCalendar:
join (stripePayments)
LOAD
    TempDate AS dataPayment,
    Week(TempDate) as Week,
    Year(TempDate) as Year,
    Month(TempDate) as Month,
    Day(TempDate) as Day,
    WeekDay(TempDate) as WeekDay,
    'Q' & Ceil(Month(TempDate) / 3) as Quarter,
    Date(MonthStart(TempDate), 'MMM-YYYY') as MonthYear,
//    WeekName(TempDate) AS WeekYear,
    Week(TempDate) & '-' & If(Week(TempDate)=1 AND
        Month(TempDate)=12, Year(TempDate)+1,
        If(Match( Week(TempDate), 52, 53) AND
            Month(TempDate)=1, Year(TempDate)-1,
            Year(TempDate))) AS WeekYear,
    if(year(TempDate) = year($(vToday)),
    if(Ceil(Month(TempDate)/3) = Ceil(Month($(vToday))/3),1,0),0) as
CQFlag,

    if(year(TempDate) = year($(vToday)),
    if(Ceil(Month(TempDate)/3) + 1 = Ceil(Month($(vToday))/3),1,0),
    if(year(TempDate)+1 = year($(vToday)),
        if(Ceil(Month(TempDate)/3) = 4 and Ceil(Month($
(vToday))/3) = 1 ,1,0),0)) as LQFlag,

    if(year(TempDate) = year($(vToday)),1,0) as CYTDFlag,
    if(year(TempDate) = (year($(vToday)) -1),1,0) as LYTDFlag,
    If(DayNumberOfYear(TempDate) <= DayNumberOfYear($(vToday)), 1,
0 ) as IsInYTD,
    If(DayNumberOfQuarter(TempDate) <= DayNumberOfQuarter($
(vToday)), 1, 0) as IsInQTD,
    If(Day(TempDate) <= Day($(vToday)), 1, 0) as IsInMTD,
    If(Month(TempDate) = Month($(vToday)), 1, 0) as IsCurrentMonth,
    If(Month(AddMonths(TempDate,1)) = Month($(vToday)), 1, 0) as
IsLastMonth
RESIDENT CalTemp
ORDER BY TempDate ASC;

DROP TABLE CalTemp;

```

- **Costes**

```
LOAD
    date(A) as dataPayment,
    "B" as costPayment
FROM [lib://Otras Fuentes/Costes.xlsx]
(ooxml, no labels, table is Sheet1);
```

- **Zones Request**

```
Qualify *;
Unqualify idZone;
//**** Carga de la tabla de zone Requests de Mongo DB ****
ZoneRequest:
Load *,
    city & '|' & state & '|' & country as idZone;
LOAD "_id",
    email,
    country,
    state,
    city,
    1 as countRequest;
SQL select *
FROM admin.zoneRequests;
Unqualify *;

Left join(ZoneRequest)
LOAD
    idZone,
    Comunidad,
    Latitud,
    Longitud,
    GeoMakePoint(Latitud, Longitud) AS CityMapPoint
FROM [$(urlLocal)LatLong.qvd] (qvd);
```

- **Seller**

```
//**** Load customers from Stripe
Customers_Map:
Mapping LOAD
    id as customerID,
    //    Email,
    DATE("Created (UTC)", 'YYYY-MM-DD') as dateStartCustomer
    //Mid("Created (UTC)",10) as HourStartCustomer
    //    Plan as planID,
    //    Status as customerStatus,
    //    "Cancel At Period End",
    //    "Account Balance",
    //    Currency
FROM [lib://Stripe/customers.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

//**** Carga de la tabla de Sellers de Mongo DB ****
Seller:
LOAD "_id" as SellerID,
    email as SellerEmail,
    "stripe_customerId" as customerID,
```

```

    "products_backdoor_total",
    "products_total",
    "products_count",
    "shops_backdoor_total",
    "shops_backdoor_count",
    "shops_markForDelete",
    1 AS CustomerRecordCounter,
    ApplyMap('Customers_Map', "stripe_customerId", 'MISSING') AS
dateStartCustomer;
SQL SELECT *
FROM admin.sellers;

```

- **Plans**

```

//**** Load Plans from Stripe ****
Plans:
LOAD id as planID,
    Name as plansName,
//    "Created (UTC)" as dateStartPlan,
    "Interval",
    Amount as planAmount,
    Currency as planCurrency,
    "Interval Count",
    "Trial Period Days"
FROM [lib://Stripe/plans.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

//**** Carga de la tabla Plans de Mongo DB ****
// mongoPlans:
Join(Plans)
LOAD "_id" as planID,
//    name as plansName,
    cost as planCost,
    "modifiers_products",
    "modifiers_shops",
    "tempLimit_period",
    "tempLimit_count",
    complement,
    "type" as mongoPlansType;
SQL SELECT *
FROM admin.plans;

```

- **UserPlans**

```

// **** mapping table for customer_ID
Customer_id_map:
Mapping Load
    customerID,
    SellerID
Resident Seller;

// //**** Carga de la tabla de UsersPlans de Mongo DB ****
UserPlansMask:
LOAD "_id" as mongoUserPlansID,
    finishDate as mongoUserPlansFDate,
    plan as planID,
    sellerId as SellerID,
    startDate as mongoPlansSDate;
SQL SELECT *
FROM admin.userPlans;

```

```

// //**** Load subscriptions from Stripe
//stripeSubscriptions:
join(UserPlansMask)
LOAD
  id as subscriptionID,
  //"Customer ID" as customerID,
  ApplyMap('Customer_id_map', "Customer ID", 'MISSING') AS SellerID,
  Plan as planID,
  Quantity,
  //"Interval",
  //Amount as SubscriptionAmount,
  Status as subscriptionStatus,
  "Created (UTC)" as dateStartSubscription,
  if(Coupon = '', 'Sin cupon', Coupon) as coupon,
  "Canceled At (UTC)",
  "Cancel At Period End",
  "Ended At (UTC)",
  Date(MonthStart("Ended At (UTC)"),'MMM-YYYY') as MonthYearEnd,
  1 AS subscriptionRecordCounter
FROM [lib://Stripe/subscriptions.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

UserPlans:
load *,
  if (planID = '5a3171010fcaaf2adb61a478', 1, Quantity) as
numPlans
Resident UserPlansMask;

drop Table UserPlansMask;

```

- **Products**

```

//**** Carga de la tabla de Products de Mongo DB ****
mongoProducts:
LOAD "_id" as mongoProductsID,
  //_sellerId as mongoSellerID,
  name,
  cost,
  previousCost,
  category,
  "shop_id" as mongoShopsID,
  backdoor,
//  startDate as mongoProductsSDate,
  date(Floor(num(date(startDate))), 'YYYY-MM-DD') as
mongoProductsSDate,
  Date(MonthStart(startDate), 'MMM-YYYY') as MonthYearProducts,
  "show_count" as productShowCount,
  isOffer,
  1 as productCount;
SQL SELECT *
FROM admin.products;

```

- Shops

```

//**** Carga de la tabla de Shops de Mongo DB ****
mongoShops:
Load *,
    city & '|' & state & '|' & pais as idZoneShops;
LOAD "_id" as mongoShopsID,
    name as mongoShopsName,
    phone ,
    email,
    street,
    number,
    if(country = 'Spain', 'España', country) as pais,
    state,
    city,
    sellerId as SellerID,
    "show_count" as shopShowCount,
    countProducts,
    "loc_coords_lat" as mongoShopsLat,
    "loc_coords_lng" as mongoShopsLong,
    startDate as mongoShopsStartDate,
    1 as shopsRecordCount;
SQL SELECT *
FROM admin.shops;

//**** Carga de la tabla de zonas de Mongo DB ****
//Left join(mongoShops)
openZones:
Load *,
    zoneCity & '|' & zoneState & '|' & paisZone as idZoneShops;
LOAD
    "_id" as zonesID,
    state as zoneState,
    city as zoneCity,
    if(country = 'Spain', 'España', country) as paisZone,
//    "coords_lat" as zoneLat,
//    "coords_lng" as zoneLong,
    GeoMakePoint("coords_lat", "coords_lng") AS CityMapPointShops;
SQL SELECT *
FROM admin.zones;

Left join(mongoShops)
LOAD
    idZone as idZoneShops,
    Comunidad as ComunidadShops
//    //Latitud,
//    //Longitud,
//    GeoMakePoint(Latitud, Longitud) AS CityMapPointShops
FROM [$(urlLocal)LatLong.qvd] (qvd);

```