



Implantación de Qlik Sense® en una e-commerce

Òscar Muñoz Álvarez

Grau en Enginyeria Informàtica

Àrea de Business Intelligence

Humberto Andrés Sanz

Atanasi Daradoumis Haralabus

15/06/2016



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Implantación de Qlik©Sense en una e-commerce</i>
Nom de l'autor:	<i>Òscar Muñoz Álvarez</i>
Nom del consultor/a:	<i>Humberto Andrés Sanz</i>
Nom del PRA:	<i>Atanasi Daradoumis Haralabus</i>
Data de lliurament (mm/aaaa):	<i>06/2016</i>
Titulació o programa:	<i>Grau en enginyeria informàtica</i>
Àrea del Treball Final:	<i>Business Intelligence</i>
Idioma del treball:	<i>Castellà</i>
Paraules clau	<i>Qlik, BI, ecommerce</i>
Resum del Treball (màxim 250 paraules):	
<p>El trabajo ha consistido en el desarrollo de una solución de <i>business intelligence</i> en el lugar mi actual lugar de trabajo. Pese a que queda fuera del área de trabajo de este trabajo de final de grado, la plataforma de business intelligence elegida ha sido Qlik©Sense Enterprise Edition debido a su alta versatilidad en el uso transversal de la empresa.</p> <p>Motobuykers es una e-commerce de reciente nacimiento que, ya entrado 2010, empieza su andadura en pleno apogeo de las tiendas en línea del tipo <i>outlet</i>. En esos primeros pasos la empresa desarrolla informes a petición y sin ningún control ni información homogénea. En 2015, debido a la actual concentración del negocio online, la empresa empieza a plantearse la adquisición de una solución de BI. A inicios de 2016 la empresa decide contratar la solución Qlik para dar un salto competitivo para poder ser más eficientes y estar por encima de sus rivales.</p> <p>Esta implantación ha consistido en una mejora del sistema actual de hojas de texto planas y soluciones externas aisladas, dando como resultado una única solución integrada dentro de Qlik©Sense Enterprise Edition. El proceso ha consistido en tres fases: análisis de la situación inicial, implantación y revisión final con los responsables implicados.</p> <p>Realizando encuestas a los responsables de área han valorado muy positivamente la herramienta consiguiendo que cada uno de ellos disponga de una visión de 360º y que sea la misma en toda la empresa.</p>	

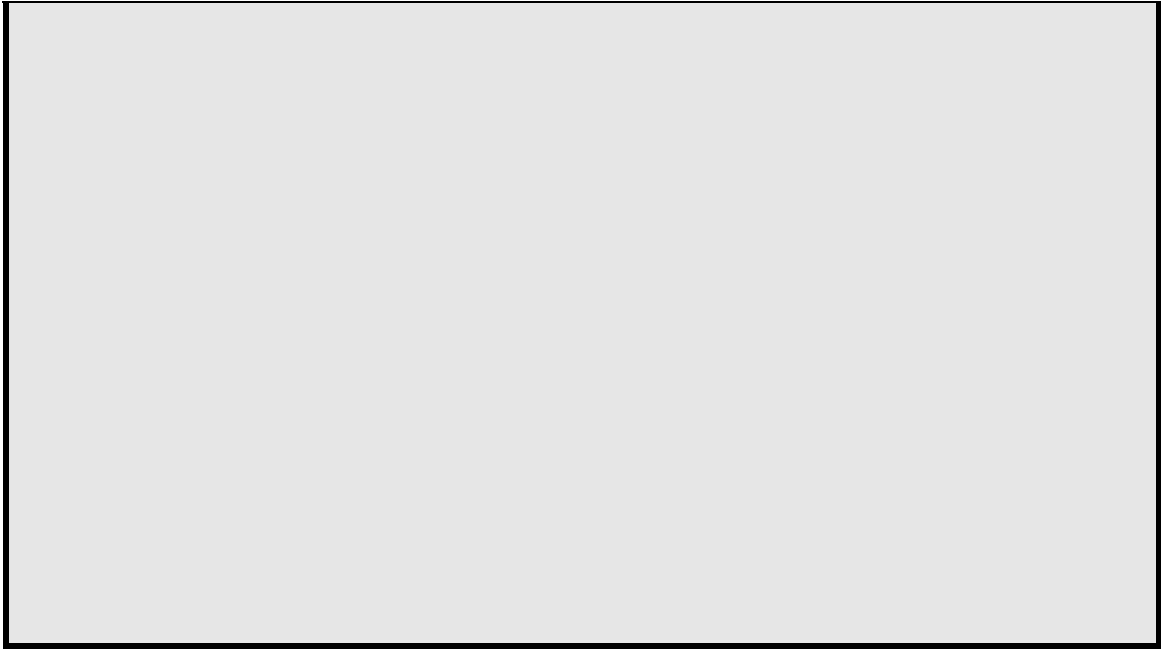
Abstract (in English, 250 words or less):

The work has involved the development of a business intelligence solution in my current workplace. Although it is outside the working area of this final degree work, the business intelligence platform has been chosen Qlik © Sense Enterprise Edition because of its high versatility in cross-company use.

Motobuyers is an e-commerce birth recently, into 2010, began its career in full online outlet stores boom. In its first steps the company developed reports on demand without any control or homogeneous information. In 2015, due to the current concentration of online business, the company begins to consider the acquisition of a BI solution. In early 2016, the company decided to contract Qlik solution to give a competitive in order to be more efficient and be above their main rivals.

This implementation has been to an improvement of the current system of plain sheets of text and isolated external solutions, resulting in a single integrated within Qlik © Sense Enterprise Edition solution. The process consisted of three phases: analysis of the initial situation, implementation and final review with all stakeholders.

Conducting surveys to area managers have valued highly the tool getting a 360-degree view and a single point of truth.



Agradecimientos

A todas aquellas personas que han estado a mi lado durante estos últimos siete años. En especial agradecer a mis padres y mi hermana por darme la oportunidad de llegar hasta aquí e inculcarme valores de superación y auto desafío. Además agradecer su comprensión durante las largas sesiones de estudio.

También agradecer a Motobuykers las facilidades para poder realizar el trabajo de fin de grado sobre la implantación del software en la empresa. Sin ellos hubiese sido más difícil.

Y, para acabar, agradecer a mi tutor de trabajo, Humberto Andrés, por su apoyo y seguimiento continuado.

Agradecimientos.....	pág 1
1. Introducción.....	pág 5
1.1. Contexto y justificación del trabajo.....	pág 5
1.2. Objetivos del trabajo.....	pág 6
1.3. Enfoque y método de trabajo.....	pág 7
1.4. Entregables.....	pág 8
1.5. Temporización.....	pág 8
2. Planificación.....	pág 11
2.1. Estudio preliminar y sistemas existentes.....	pág 12
2.1.1. Modelo de datos para Qlik.....	pág 14
2.2. Preparación del entorno e instalación.....	pág 15
3. Desarrollo.....	pág 16
3.1. Construcción del data warehouse – ETL's.....	pág 16
3.1.1. WeRide.....	pág 17
3.1.2. Motobuykers.....	pág 21
3.1.2.1. ERP.....	pág 22
3.1.2.2. Google Analytics.....	pág 23
3.1.2.3. Google Adwords.....	pág 24
3.1.2.4. Netrivals.....	pág 25
3.1.2.5. Zendesk.....	pág 25
3.1.2.6. Mailing.....	pág 26
3.1.2.7. Retargeting.....	pág 27
3.1.3. Enriquecimiento de datos.....	pág 28
3.1.4. Modelo de datos final.....	pág 31
3.2. Visualización de información.....	pág 33
3.2.1. Motobuykers.....	pág 33
3.2.1.1. Estado general.....	pág 34
3.2.1.1.1. Dashboard.....	pág 34
3.2.1.1.2. Ventas y Abonos por Zona.....	pág 35
3.2.1.1.3. Business plan.....	pág 36

3.2.1.2.	Departamento Comercial.....	pág 36
3.2.1.2.1.	Ventas, CMg: proveedores.....	pág 37
3.2.1.2.2.	Ventas, CMg: listado productos.....	pág 37
3.2.1.2.3.	Ventas, CMg: campañas.....	pág 38
3.2.1.2.4.	Ventas, CMg: marca-producto.....	pág 39
3.2.1.2.5.	Campañas: ventas año-mes-día.....	pág 39
3.2.1.2.6.	Marca: ventas año-mes-día.....	pág 40
3.2.1.2.7.	Heatmap Campaña-día.....	pág 40
3.2.1.2.8.	Feed productos.....	pág 41
3.2.1.2.9.	Netrivals.....	pág 42
3.2.1.2.10.	Detalle pedidos.....	pág 42
3.2.1.3.	Departamento Operaciones.....	pág 43
3.2.1.3.1.	Anulaciones.....	pág 43
3.2.1.3.2.	Entregas.....	pág 44
3.2.1.3.3.	Stock.....	pág 44
3.2.1.3.4.	Devoluciones.....	pág 45
3.2.1.3.5.	Stock Almacén.....	pág 45
3.2.1.3.6.	Vozitel.....	pág 46
3.2.1.3.7.	Zendesk.....	pág 47
3.2.1.3.8.	Expediciones.....	pág 47
3.2.1.4.	Departamento Marketing.....	pág 48
3.2.1.4.1.	Business plan – canal.....	pág 48
3.2.1.4.2.	Newsletter.....	pág 49
3.2.1.4.3.	Newsletter facturación.....	pág 49
3.2.1.4.4.	Newsletter detalle.....	pág 50
3.2.1.4.5.	Fidelización.....	pág 50
3.2.1.4.6.	Rowdata.....	pág 51
3.2.1.4.7.	Rowdata Exportar.....	pág 52
3.2.1.4.8.	Motos.....	pág 52
3.2.1.4.9.	Estilos.....	pág 53
3.2.1.4.10.	RFM.....	pág 53
3.2.1.4.11.	Analytics.....	pág 54
3.2.1.4.12.	Analytics – Revisión Canal.....	pág 55
3.2.1.4.13.	Analytics – Check Revenue.....	pág 55

3.2.1.4.14.	Adwords.....	pág 55
3.2.1.4.15.	Tokens tarjetas.....	pág 55
3.2.1.4.16.	Usuarios.....	pág 57
3.2.1.4.17.	Promociones.....	pág 58
3.2.1.4.18.	Popup.....	pág 58
3.2.1.4.19.	Leads – Buyers.....	pág 58
3.2.1.4.20.	Referral.....	pág 59
3.2.1.4.21.	Distribución geográfica.....	pág 60
3.2.1.5.	Departamento finanzas.....	pág 60
3.2.1.5.1.	Check facturas.....	pág 61
3.2.1.5.2.	Abonos.....	pág 61
3.2.2.	WeRide.....	pág 62
3.2.2.1.	Dashboard.....	pág 62
3.2.2.2.	Users.....	pág 63
3.2.2.3.	Geo Groups.....	pág 64
3.2.2.4.	Geo Users.....	pág 64
3.3.	Revisión de dashboards.....	pág 65
4.	Conclusiones.....	pág 66
4.1.	Lecciones aprendidas.....	pág 66
4.2.	Reflexión sobre el cumplimiento de objetivos.....	pág 67
4.3.	Seguimiento de la planificación y metodología.....	pág 68
4.4.	Líneas de trabajo futuro.....	pág 69
5.	Glosario.....	pág 70
6.	Bibliografía.....	pág 73
7.	Anexos.....	pág 74

1. Introducción

1.1. Contexto y justificación del trabajo

En el 2010 surgió la expansión de las tiendas en línea dedicadas a vender producto de fuera de temporada de marcas punteras, también conocidas por *e-commerce* de venta *outlet*. Habitualmente, cada una de estas tiendas nació especializada en una tipología de productos, por ejemplo nos podíamos encontrar con especialistas en ropa de esquí. Es en este momento cuando un grupo de compañeros decidieron crear la empresa Motobuykers¹ para acontecer especialistas en la distribución al por menor de productos relacionados con los motoristas. Des de su nacimiento dos rondas de financiamiento han permitido que la empresa se consolidase como una plataforma líder en la venta de productos relacionados con las motocicletas. Al mismo tiempo la empresa desarrolló una red social para motoristas con nombre WeRide² que busca, primero de todo, ser útil para al motorista pero al mismo tiempo aprender del usuario.

Tal como estábamos diciendo, la empresa basa sus movimientos en dos ramas:

- Motobuykers: portal web para la venta al detalle
- WeRide: aplicación móvil de geolocalización de rutas

El mundo online ha sufrido muchos cambios des del 2010 teniendo que afrontar cambios muy importantes en las vías de comunicación y en la vida del usuario. Este aspecto pasa, sobretodo, por conocer mucho mejor al usuario centralizando la información y generando un dato de única verdad. En los últimos meses, Motobuykers, afronta etapas de cambios que les permitan afrontar los futuros retos con garantías de éxito. Esto significa traspasar el modelo de datos actual a una de las plataformas de *business intelligence* punteras del sector como es Qlik³ dentro del paquete Qlik Sense Enterprise

¹ Equipamiento moto y accesorios – www.motobuykers.es

² WeRide App – werideapp.com/es

³ Qlik: Software de Business Intelligence y Visualización – www.qlik.com

Edition⁴ en el cual se buscará integrar toda la información de la empresa en un único repositorio.

1.2. Objetivos del trabajo

La dirección de Motobuykers ha entendido la importancia de traspasar los informes actuales de tablas planas a una solución de *business intelligence* consolidada que permita integrar y visualizar las diferentes fuentes de datos.

Aprovechando los conocimientos previos de la base de datos comercial así como los diversos flujos de integración de datos existentes se buscará la creación de *dashboards* que permitan disponer de una visión general y al detalle del negocio. Por tanto, los objetivos del trabajado de final de grado se basan en:

- Integrar las diferentes fuentes de datos realizando tareas de ETL haciendo uso del programa Pentaho Data Integration⁵, en su versión *community*.
- Creación de un *dashboard* para ofrecer información actualizada a los responsables de la empresa para ayudarles en la toma de decisiones y en las tareas diarias de control.

El objetivo final de la integración con la plataforma Qlik Sense es conseguir prever las futuras vendas que puedan realizar los clientes basándonos en el análisis de datos pasados. En el mismo sentido, como objetivo general, se buscar hacer visibles el conjunto de datos que la empresa dispone del cliente e interconectarlos con otros datos intrínsecos de la empresa.

⁴ Qlik Sense Enterprise Edition - <http://global.qlik.com/es/explore/products/sense/enterprise>

⁵ Pentaho Data Integration – <http://community.pentaho.com>

Queda fuera de objetivo del TFG el análisis de soluciones de *BI* que están actualmente en el mercado. Esta tarea fue realizada en los meses anteriores al trabajo y, por este motivo, no dispondremos del detalle de esta información.

Cumpliendo estos objetivos la empresa conseguirá fortalecer su posición en el mercado respecto a sus competidores. Además, como gestión interna de la empresa, se evitará la generación de informes diarios que resultaban costosos de producir y que contenían información aislada.

1.3. Enfoque y método de trabajo

Actualmente la empresa dispone de un sistema ERP propietario desarrollado a medida por la misma. En esta plataforma se gestionan todos los datos que controla la empresa sobre los productos de venta: productos, atributos, proveedores, clientes, etc. Resta fuera del alcance del proyecto la integración financiera con *SAGE* sobre movimientos bancarios y salarios de los trabajadores.

Todos los procesos de *ETL* que enriquecerán la plataforma de BI serán desarrollados por mi mismo. Debido a la flexibilidad de Qlik Sense, algunos de los procesos ETL existentes se continuaran usando. En cambio, otros, se tendrán que crear de nuevos para tener datos en el formato correcto. En la mayoría de los casos las *ETL's* serán desarrolladas usando la herramienta *Pentaho Data Integration*, en su versión *community*, que nos permitirá conectar con las diferentes fuentes de datos. En el mismo sentido, los datos de *Google Analytics* pueden ser consumidos directamente des de la plataforma Qlik Sense pero se ha decidido crear un desarrollo propio con el objetivo de adaptar los datos y disponer de ellos en el ERP.

La implantación del software acontece un pacto en la empresa para cumplir los plazos marcados por la UOC para el desarrollo del trabajo de final de grado. En

añadido, en el TFG se incluirán hitos empresariales que buscarán la obtención de requisitos con los implicados, la presentación y entrega del trabajo.

Por tanto, el trabajo de final de grado planteado basará su actuación en un proyecto de implantación del software *Qlik Sense Enterprise Edition* migrando los sistemas actuales a la nueva plataforma, siempre con especial cuidado en la gestión del proyecto y la gestión del cambio.

1.4. Entregables

La realización del proyecto de final de grado anteriormente descrito conllevará la obtención de un *dashboard*, personalizado para cada departamento, el cual lo compondrán los diferentes KPI's que se tendrán que visualizar.

Conjuntamente con la realización de los *dashboards* devendrá la realización de las diferentes *ETL's* que nutrirán la plataforma Qlik Sense.

Al ser un trabajo desarrollado que tiene como destino un entorno real la empresa pide expresamente no tener que publicar el contenido del trabajo. En cualquier caso se realizarán muestras de la plataforma con información ficticia.

1.5. Temporización de las tareas

A continuación presento un diagrama de Gantt con la planificación de las tareas:

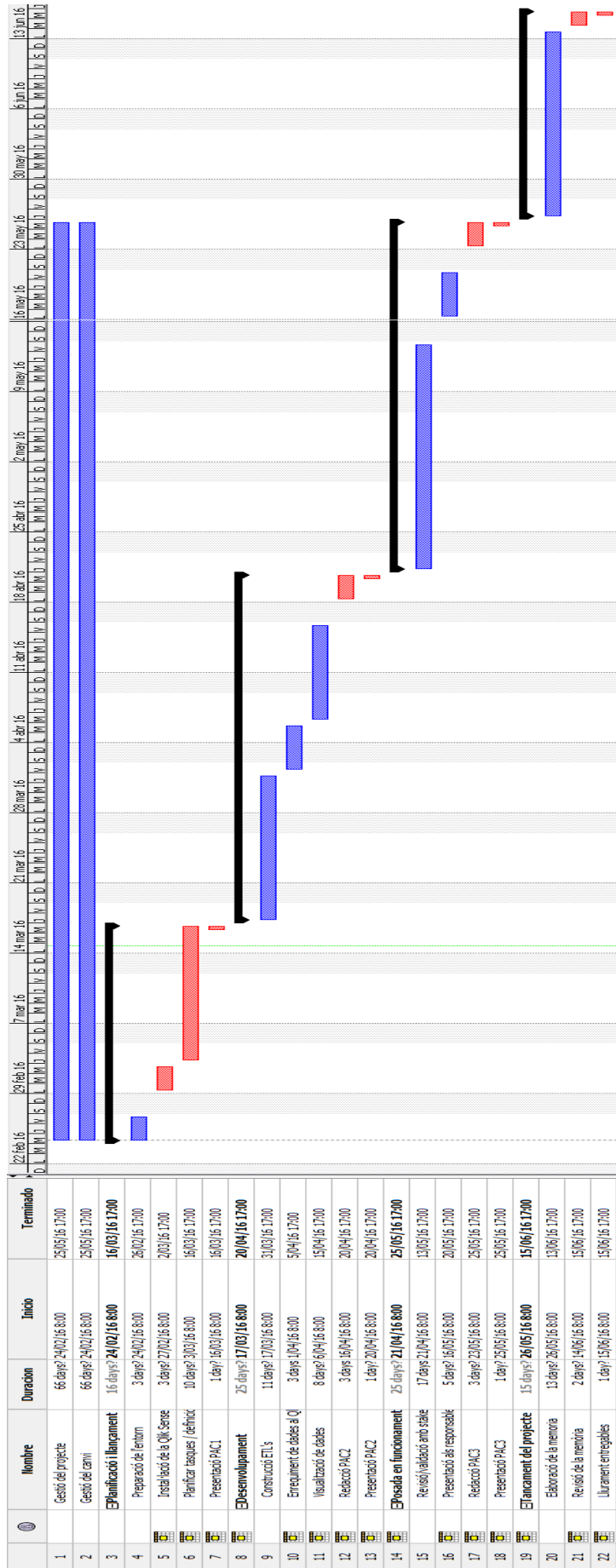


Imagen 1 – Temporización

En la planificación del proyecto se han tomado como líneas de referencia las entregas de las PAC's. Además, el proyecto se puede desglosar en cuatro grandes etapas:

- Planificación y lanzamiento: en este punto se realizarán las acciones oportunas para determinar el alcance del proyecto y su planificación. Esta etapa culminará con la entrega de la PAC1.
- Desarrollo: una vez el proyecto esté definido será en esta etapa cuando se crearán los procesos para enriquecer el sistema. Esta etapa desembocará en la entrega de la PAC2.
- Puesta en marcha: esta etapa servirá para revisar el correcto funcionamiento con los diversos stakeholders, presentación a la dirección de la empresa así como su publicación. Esta etapa originará la entrega de la PAC3.
- Cierre del proyecto: punto final del proyecto donde se recapitulará el trabajo desarrollado para obtener un entregable final del trabajo de final de grado.



Imagen 2 - Ciclo de vida de un proyecto PMBOK

De manera transversal, al mismo tiempo, también se llevarán a cabo las etapas de 'Seguimiento y Control' y Realimentación que serán desarrolladas durante toda la duración del proyecto. A pesar de contar con una etapa de 'Revisión y Presentación' se considera importante que durante todo el proyecto se pueda trabajar conjuntamente con stakeholders para avanzar parte del proyecto.

Cabe a destacar que las fases del proyecto han sido construidas para coincidir con los entregables de la evaluación continuada y poder realizar un seguimiento exhaustivo del trabajo realizado.

2. Planificación y lanzamiento

La implantación de un sistema de *business intelligence* comporta la adquisición de datos históricos para su posterior uso en la plataforma. En los siguientes puntos veremos los datos disponibles y cuál ha sido su uso en el nuevo sistema.

La gestión de las tareas y la planificación en el tiempo ha sido gestionada mediante la plataforma interna **Redmine**⁶ que dispone la empresa. Esta herramienta ha permitido la creación de un proyecto específico dentro de ella para la inclusión de todas las tareas a desarrollar.

De manera complementaria la gestión de las tareas ha sido organizada siguiendo la metodología **Scrum**⁷ sobre sprints de una semana. En el inicio de cada sprint he visualizado las tareas que se deberían desarrollar según la combinación de prioridad y puntuación. Realizar este momento previo a la actuación ha permitido mejorar la eficacia en el desarrollo.

Además, de manera paralela, disponer de una plataforma de registro de tareas y de sus actuaciones ha permitido que los stakeholders pudiesen conocer en todo momento el estado de la tarea así como una estimación de la fecha de entrega.



⁶ Redmine - <http://www.redmine.org>

⁷ Scrum - [https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))

Vale la pena destacar mi desconocimiento de la plataforma Qlik Sense anteriormente al inicio del proyecto de final de grado. Simultáneamente a la primera fase del proyecto llevé a cabo una pequeña formación en la plataforma que me ha permitido ser totalmente autónomo y llegar a situaciones que no hubiesen sido posibles siguiendo las plataformas tradicionales, como Excel.

2.1. Estudio preliminar y sistemas existentes

La estructura informática de la empresa se basa en la información relativa a los usuarios y, así como, las compras realizadas. Debido a esta situación, como podemos pensar la información a tratar está estructurada sobre un modelo de base de datos en un esquema de estrella. Tal como resulta normal, la estructura es compartida con las consultas de la web (aunque atacando base de datos replicadas distintas).

Paralelamente, en diferentes periodos de tiempo, la base de datos es enriquecida con datos proveniente de otros sistemas anexos. Esta información está siendo incorporada actualmente, por ejemplo, para adquirir información relativa a la analítica web como puede ser trabajar con Google Analytics.

Todos los datos de la empresa están siendo incorporados a un sistema gestor de base de datos del tipo **MySQL** debido a su bajo coste y flexibilidad. Durante todo el desarrollo del trabajo de final de grado se continuará usando este sistema gestor.

En el mismo sentido, los procesos de adquisición de datos, Motobuykers, los tiene desarrollados usando la herramienta de integración de datos: Pentaho Data Integration en su versión Community. Se considera continuar usando esta herramienta debido al conocimiento actual de ella y de su gran adaptabilidad con los datos. En concreto, estamos hablando de un software que puede interactuar con multitud de fuentes.

Actualmente la empresa dispone de multitud de procesos de integración. Motobuykers basa su actuación en dos tipologías distintas: por una parte las

integraciones de datos como pueden ser de Google Analytics y de Google Adwords. En una segunda actuación, la misma herramienta permite la generación de informes en Excel que contienen la información incorporada y procesada para su posterior estudio.

La empresa no dispone de información histórica anterior a abril de 2014. En ese momento se desarrolló un cambio en la plataforma web, y en consecuencia en su base de datos, y alguna información no fue imposible traspasar. Por tanto, en los datos de antes de abril de 2014 podremos identificar KPI's de facturación pero no ir al detalle del producto vendido.

En el mismo sentido, en el cambio de base de datos realizado en el 2014 la empresa decidió no gestionar la información como un data warehouse (dwh) estricto. Por ejemplo, los cambios de estado de pedidos se actualizan el valor sin que quede constancia de tal cambio.

Tal como hemos introducido, las fuentes de datos que nos servirán para enriquecer los datos existentes son:

- Todos aquellos ya existentes en el **SGBD** que hacen referencia a la vida de la página web y, por supuesto, a la de sus usuarios.
- Google **Analytics** nos servirá para la integración de datos analíticos de la página web. Por ejemplo, saber las sesiones que hemos obtenido en un momento dado.
- **Google Adwords** facilitará información sobre las ventas que provienen del sistema de referidos.
- **Netrivals** es la plataforma que nos servirá para la captación de precios de venta de la competencia
- **Zendesk** y **Vozitel** servirán para la integración de datos de atención al cliente.
- Información relativa a la plataforma de envío de emails (**Webpower**)
- Datos de asociados (**retargeting**) sobre Criteo y Webgains

Por todo esto marcamos tres soluciones a desarrollar para obtener resultados satisfactorios:

- Diseño e implementación de un modelo de datos consistente y eficaz que permita rescatar los datos web para ser complementados con datos de proveedores para acabar siendo una fuente rica en información.
- Creación, o actualización, de procesos ETL que permitan el enriquecimiento de los datos anteriormente descritos.
- Implantación del software contratado con el objetivo de desarrollar en él dashboards que reflejen la salud de la empresa.

En todo el proceso de definición los datos de WeRide son tratados en la misma definición de los objetivos, aunque se omita por recurrencia. En este sentido cabe destacar que se van a desarrollar dos dashboards totalmente independientes.

2.1.1. Modelo de datos para Qlik

La herramienta de business intelligence Qlik se caracteriza por dos aspectos que consiguen mejorar el rendimiento y permite integrar las diferentes bases de datos. Por un lado, la estructura de datos permite un esquema mixto que combina el modelo de estrella con un modelo plano (concatenación). En otras palabras, a elección del desarrollador, los datos pueden ser cargados en el modelo estrella pero, al mismo tiempo, se pueden realizar uniones de tablas. Este método de trabajo es muy habitual debido a que permite la combinación de datos de diferente índole.

En nuestro caso se planifica hacer uso del sistema mixto para poder combinar los datos de las diferentes fuentes. El objetivo, por ejemplo, es unir en una misma tabla los datos de los pedidos con los de atención al cliente. Una vez los datos estén unidos las dos tablas comparten la información de las columnas y los filtros actúan sobre las dos tablas unidas. En contraposición, continuar con el esquema de estrella no es una opción ya que no permitiría trabajar con las diferentes fuentes, sino solo con aquellas que dispongan de nexos de unión.

En conclusión, tal como se ha descrito, usar este nuevo modelo de datos en ocasiones puede parecer volver a tablas planas, consideradas anticuadas. Es

una opinión errónea que, como veremos en las siguientes páginas, nos ayudará mucho a la combinación de datos de las diferentes fuentes.

2.2. Preparación del entorno e instalación

Para determinar la capacidad del servidor nos servimos de la documentación del fabricante. Los requisitos generales aconsejados son:

- Windows Server 2012
- 8GB memoria ram
- Multi-core CPU

El servidor finalmente elegido ha sido:

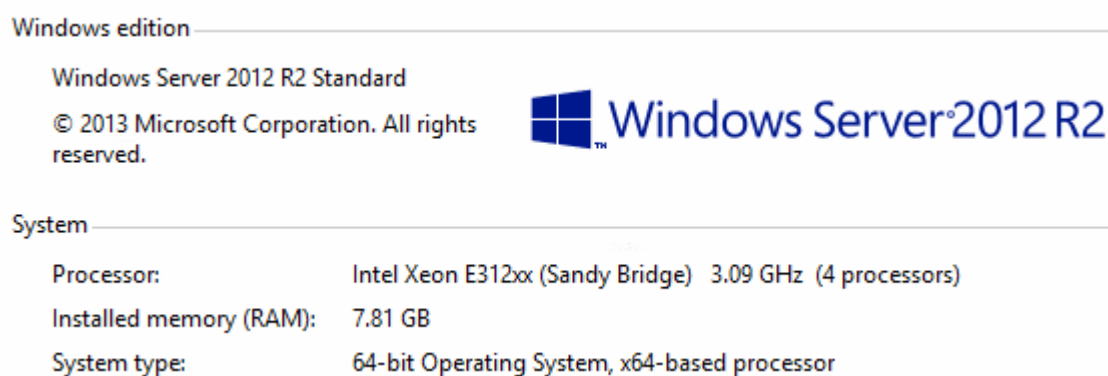


Imagen 3. Captura pantalla servidor Qlik Sense

El servidor contratado consiste en una VPS en OVH⁸ que ofrece las características anteriormente descritas, suficientes para el uso de la plataforma.

Una vez se realizó el pago del software a Qlik este facilitó un número de licencia que comportará la activación de cinco usuarios simultáneos. El proceso de instalación se realiza sobre un sistema paso a paso de rápida instalación. En nuestro caso no sucedió ningún problema destacado. Es importante destacar que Qlik Sense Enterprise Edition basa su rango de actuación en la plataforma web y que, por tanto, no puede convivir con IIS.

⁸ OVH <http://www.ovh.es>

Una vez la instalación ha concluido se facilitó el acceso a los usuarios con roles generales de usuario.

El gestión prevista de para los dashboards es la creación de 'apps' privadas donde se desarrollará todo el trabajo de creación, tanto interno como visual. Una vez los stakeholders aprueben el contenido se hará una publicación de la app para que todos los usuarios puedan disponer de los datos.

3. Desarrollo

Tan buen punto se ha considerado que se ha adquirido una buena planificación de las dimensiones del proyecto, estando todas las partes de acuerdo, se efectuó el inicio de la implantación del software con el traspaso de la información.

Previamente a esta última acción hizo falta crear diferentes ETL's para la extracción de datos para su futura carga en la plataforma. En los siguientes puntos veremos como se realizó y los problemas con que nos encontramos.

Acto seguido se produjo la construcción de los QVD que son la fuente que carga la información en Qlik Sense. Una vez la información estaba cargada en memoria se procedió a la creación de las visualizaciones.

Acto seguido, para acabar, se realizó la revisión de los diferentes dashboards con los stakeholders para determinar el cierre o los cambios a efectuar para cumplir con sus pretensiones.

3.1. Construcción del data warehouse – ETL's

Todo sistema de business intelligence basa su labor en los datos que contiene. Por tanto, contra más datos recojamos más valor contendrán. En nuestro caso, tal como hemos dicho, la construcción del datawarehouse se sustenta en los datos facilitados por la cotidianidad de la página web de Motobuykers como de la aplicación de WeRide. La información contenida en estos almacenes ha sido

complementada de diferentes maneras para que realmente aporte valor añadido al dato en sí.

Tal como decimos la primera estrategia para la implantación de Qlik Sense es la construcción de las ETL's que serán integradas:



Imagen 4 - Fuente: <http://businessintelligence.es>

3.1.1.1. WeRide

Los datos de la aplicación de WeRide restaban escondidos a la empresa. No se tenía ningún tipo de reporte sobre la misma. En esta ocasión se consideró oportuno, como objetivo, la visualización de los datos básicos de la app. Por tanto, los datos que se visualizaron pasarían a ser, casi por completo, de la misma aplicación. Exceptuando la ETL que se generó sobre los datos de Google Analytics para la aplicación. Debido a problemas con el plugin de *Kettle* para Google Analytics se tuvo que programar una aplicación que obtuviese los datos usando la API facilitada por Google.

Las métricas que fueron cargadas:

- Fecha
- Categoría del dispositivo
- Sistema operativo
- Versión del sistema operativo
- País

Al mismo tiempo las medidas:

- Sesiones
- Usuarios
- Páginas vistas
- Duración de las sesiones

Debido a que los datos de Google Analytics no son fiables hasta pasadas 48h se decide que cada día se incorpore la información del día anterior y la información de tres días atrás:

Step name **Table Input**
 Connection mtbks_stats_mo

SQL

```

SELECT
  date - 3 as fecha,
  deviceCategory as deviceCategory,
  operatingSystem as operatingSystem,
  operatingSystemVersion as operatingSystemVersion,
  country as country,
  sessions as sessions,
  users as users,
  screenviews as screenviews,
  sessionDuration as sessionDuration
FROM
  BI_WeRide.GA_RowData_Qlik_WeRide_tmp
  
```

Imagen 5 - Proceso WeRide Analytics

Pasando las fechas como parámetro Kettle ejecuta la librería de java que acaba incorporando la información a una tabla temporal antes de la incorporación al histórico:

BI_WeRide.GA_RowData_Qlik_WeRide_tmp: 255 filas en total (aproximadamente)

fecha	deviceCategory	operatingSystem	operatingSystemVersion	country	sessions	users	screenviews	sessionDuration
2016-05-31	mobile	Android	5.0.2	IT	11	9	63	1877
2016-05-31	mobile	Android	5.0	ES	26	22	332	5400
2016-05-31	mobile	Android	5.0	DE	3	3	6	50
2016-05-31	mobile	Android	5.0	CH	1	1	1	0
2016-05-31	mobile	Android	5.0	FR	1	1	1	0

Imagen 6 - Tabla GA_RowData_Qlik_WeRide_tmp

Esta tabla nos servirá para almacenar temporalmente los datos adquiridos. Una vez la API nos devuelva los datos pasamos a realizar una inserción completa de la información adquirida. Práviamente, para eliminar duplicidades, se elimina de la tabla histórica todos los datos de las fechas que coincidan con los recién adquiridos:

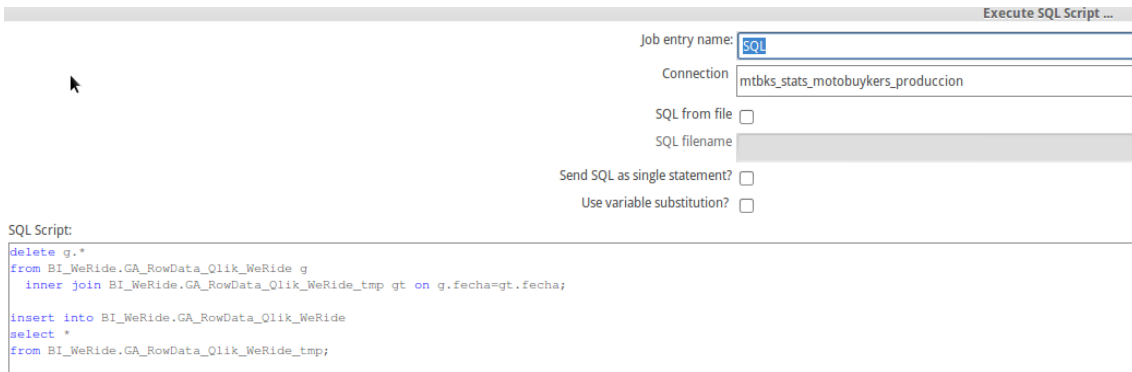


Imagen 7 - Proceso inserción datos adquiridos WeRide Analytics

Dando como resultado la inserción histórica en la tabla 'GA_RowData_Qlik_WeRide':

BI_WeRide.GA_RowData_Qlik_WeRide: 22.330 filas en total (aproximadamente), limitado a 1.000

fecha	deviceCategory	operatingSystem	operatingSystemVersion	country	sessions	users	screenviews	sessionDuration
2016-05-30	media	Android	4.1.2	ES	12	5	99	2824
2016-05-30	media	Android	4.4.2	FR	4	4	43	1850
2016-05-30	media	Android	6.0	DE	5	5	40	622
2016-05-30	media	Android	5.1.1	AR	2	1	30	827
2016-05-30	media	Android	6.0	FI	2	1	38	5898
2016-05-30	media	Android	5.1.1	ES	46	34	630	15302
2016-05-30	tablet	Android	4.4.2	ES	1	1	19	1207
2016-05-30	media	Android	5.0.1	DE	3	3	3	0
2016-05-30	media	Android	4.3	FR	4	3	8	27
2016-05-30	media	Android	5.0.1	FR	3	2	9	37
2016-05-30	media	Android	5.0.1	US	3	3	27	456
2016-05-30	media	Android	5.0.2	GB	1	1	50	1173
2016-05-30	media	Android	5.0	CH	1	1	2	6
2016-05-30	media	Android	6.0	CO	1	1	1	0
2016-05-30	media	Android	6.0.1

Imagen 8 - Tabla histórica de visitas de WeRide

Quedando el proceso final de ETL sobre Kettle de la siguiente forma:

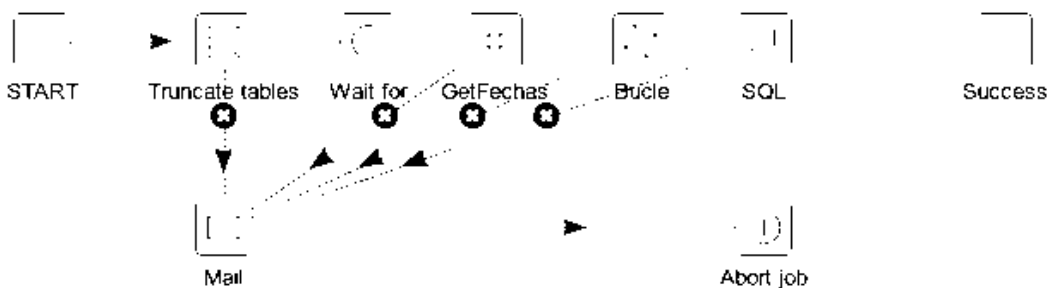


Imagen 9 - ETL Google Analytics WeRide

Otro objetivo definido en la sección de WeRide es la categorización de los usuarios. Se considera importante saber que usuarios son aquellos que confían más en la marca. Para determinar esta situación se realiza una tabla con puntuaciones según los datos que nos ha facilitado el usuario:

	Puntuación			Máximo	Mínimo
	Visible	No visible	No info		
Perfil completado					
Foto	4		0	4	0
Nick	4		4	4	4
telefono		3	0	3	0
Moto	3		0	3	0
Descripción	1		0	1	0
Nombre	2	1	0	2	0
Sexo	2	1	0	2	0
Fecha nacimiento	2	1	0	2	0
Ciudad	2	1	0	2	0
País	2	1	0	2	0
CP	2	1	0	2	0
Piloto/Copiloto	2		0	2	0
Total			4	29	4
Privacidad					
	Visible	No visible		Máximo	Mínimo
Amigos	1			1	0
Grupos	1			1	0
Rutas	1			1	0
Total				3	0
Notificaciones					
	Activas	No activadas		Máximo	Mínimo
Solicitud amistad	1	0		1	0
WeChat amigos	1	0		1	0
WeChat grupos	1	0		2	0
Notif. Weforum	1	0		2	0
Notif. Wespoteo	1	0		1	0
Total				7	0

Tabla 1: Ponderaciones WeRide

Tal como se puede ver en la tabla, algunos de los campos son obligatorios. Los voluntarios pueden ser visibles o estar ocultos al resto de usuarios. Una vez identificados los campos en cuestión se procedió a la puntuación haciendo uso de una función programada en SQL para determinar la puntuación final. Este dato, como veremos más adelante, además de poder tener una visión más

clara de los datos que disponemos de los usuarios nos permitirá combinarlo con las ventas para determinar si los usuarios que más sabemos son los que más compran.

3.1.2. Motobuykers

Anteriormente se ha descrito una serie objetivos previstos para la consecución del proyecto final. A continuación veremos el detalle de como se ha integrado la información de los diferentes proveedores en el sistema de base de datos actual de Motobuykers.

Tal como hemos visto la integración de información se realiza sobre 7 grandes ejes:

1. Datos internos de la web provenientes del ERP/WEB de la empresa.
Tanto a nivel operacional como logístico, pasando por información del usuario.
2. Google Analytics
3. Google Adwords
4. Netrivals
5. Zendesk, Zopim y Vozitel
6. Proveedor de emailing: Webpower
7. Retargeting

Como podemos observar las fuentes de datos son extensas y muy variadas. Tal como marcan los objetivos del proyecto nos podemos hacer una idea que una vez tengamos los datos integrados dispondremos de información muy completa y conseguiremos una visión de 360º del negocio.

3.1.2.1. ERP

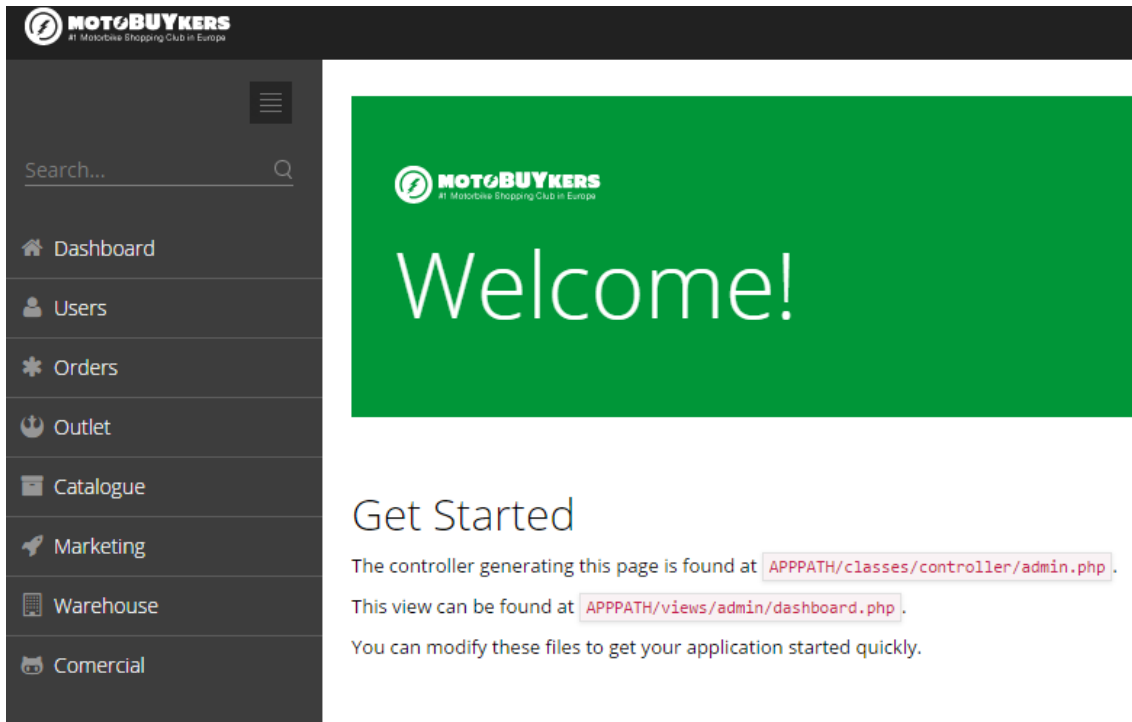


Imagen 10 - ERP propio Motobuyers

Tal como hemos ido introduciendo los datos del ERP serán cargados en Qlik Sense sin ser previamente tratados. Tal como comentamos la información, a grandes rasgos, que nos facilitará el ERP es:

- Usuarios
- Objetivos de empresa e inversión
- Actividades promocionales (cheques regalo)
- Pedidos
- Expediciones
- Productos
- Proveedores
- Motocicletas
- Facturas
- Devoluciones

Toda esta información será cargada en la nueva plataforma. Cabe destacar que la información anteriormente descrita será complementada con los datos de los siguientes puntos.

3.1.2.2. Google Analytics

El objetivo de la integración con Google Analytics es la descarga de datos relativos a indicadores de la web y datos de las transacciones. Se ha creado un proceso que combina Kettle con Java (Google API) que cada día descarga/actualiza la información de la web y de las transacciones.

Los indicadores de la web que se descargan:

- Fecha
- Zona
- Categoría del dispositivo
- Agrupación de canal
- Fuente
- Medio
- Sesiones

Juntamente con las medidas:

- Usuarios
- Nuevos usuarios
- Páginas vistas
- Transacciones
- Importe de las transacciones
- Rebotes
- Duración de la sesión
- Sesiones de búsqueda

Además se busca conocer tener más detalle de la fuente de las transacciones del sistema. Por tanto, nos descargaremos la siguiente información:

- Fecha
- Zona

- Categoría del dispositivo
- Agrupación de canal
- Fuente
- Medio
- Id de transacción
- Importe de la transacción
- Id de adwords (si aplica)
- Gastos de envío

Un tercer proceso a efectuar es la descarga de información de los indicadores de la web a nivel de cada url de la web, en concreto sobre productos. En este caso estaríamos exportando información similar a las anteriores. Nos serviría para, más adelante, poder visualizar cuál de los productos de la web tiene mejor ratio de conversión.

Una muestra del resultado:

fecha	id_zona	deviceCategory	channelGrouping	source	medium	pageviews	uniquePageviews	timeOnPage	entrances	bounces	exits	pagePath
2016-06-03	3	desktop	Referral	criteo	referral	2	2	41	0	0	0	adventure-low-brun
2016-06-03	3	desktop	Organic Search	google	organic	1	1	81	0	0	0	adventure-low-brun
2016-06-03	3	desktop	Email	facebook	email	1	1	6	0	0	0	adventure-low-brun
2016-06-03	1	desktop	(Other)	direct	direct	1	1	0	1	1	1	adventure-low-brun
2016-06-03	1	mobile	Organic Search	google	organic	1	1	119	0	0	0	adventure-low-brun
2016-06-03	1	mobile	Direct	direct	direct	1	1	53	0	0	0	adventure-low-brun
2016-06-03	1	desktop	Referral	facebook	referral	1	1	19	1	0	0	adventure-low-brun
2016-06-03	1	desktop	Referral	criteo	referral	1	1	0	1	1	1	adventure-low-brun
2016-06-03	1	desktop	Paid Search	google	paid	3	3	245	0	0	0	adventure-low-brun
2016-06-03	1	desktop	Organic Search	google	organic	2	2	1265	1	0	0	adventure-low-brun

Imagen 11 - Tabla GA_RowData_Qlik_Inf_Productos

3.1.2.3. Google Adwords

La adquisición de información con Adwords tiene como objetivo conocer los ratos de coste de cada campaña Usando la API de Google Adwords conseguimos adquirir la siguiente información:

- Id de campaña
- Campaña
- Clicks
- Impresiones
- Coste
- Fecha

- Zona
- Cuenta

A partir de esta información, tal como explicábamos, podemos saber el importe invertido y sacar ratios de conversión con el objetivo de determinar que tipología de productos es más favorable para ser promocionada.

3.1.2.4. Netrivals

Netrivals es una plataforma que nos permite conectar nuestros productos con los mismos en la competencia. Saber este dato permitirá al departamento comercial conocer mejor el motivo de las ventas o, en caso contrario, porque no se están vendiendo determinados productos.

La integración con Netrivals se realiza a través de una petición de url que devuelve un archivo xml con la información de precios del resto de competidores. Tal como previamente ya avanzamos, Netrivals no facilita información histórica, simplemente la información cuando se pide. Una muestra de la información descargada:

Fecha	Code	StoreID	Name	Price	Stock	SellerName	Position
2016-06-04	10902318	27153	nosodamoto.com	362.81	yes		1
2016-06-04	10902318	11076	febrequesmotos.es	362.95	yes		2
2016-06-04	10902318	11608		413.00	yes	(NULL)	3
2016-06-04	10902318	27083	Motobuykerns	417.69			4
2016-06-04	10902318	27082	motobouzes	436.00	yes		5
2016-06-02	10902318	27153	nosodamoto.com	362.81	yes		1

Imagen 12 - Tabla netrivals_child

3.1.2.5. Zendesk

La integración con Zendesk ha sido la más complicada. A pesar de disponer de API los datos eran devueltos de manera parcial y en formato json. Se añadía la situación comentada que se debían realizar una importante cantidad de peticiones a las diferentes APIs hasta llegar a conocer todos los datos del ticket. Con objetivo de solucionar este tema se consideró oportuno usar las librerías java, facilitadas por Zendesk, para la descarga de datos incrementales.

Una vez integrados con la plataforma disponemos de gran cantidad de información que deberá ser tratada en Qlik. Un ejemplo de los principales KPI's:

- Id de ticket
- Canal de resolución
- Trabajador que atendió al cliente
- Datos del cliente
- Tiempos de resolución

En complemento a Zendesk, Motobuykers desvía las llamadas no contestadas, como pueden ser las llamadas fuera de horario, a una centralita externa (Vozitel). Esta plataforma no dispone de API para la integración, así que cada día se realiza la exportación manual de los datos. El archivo resultando es enviado por email a una dirección corporativa con el objetivo que Kettle la integre en la base de datos:

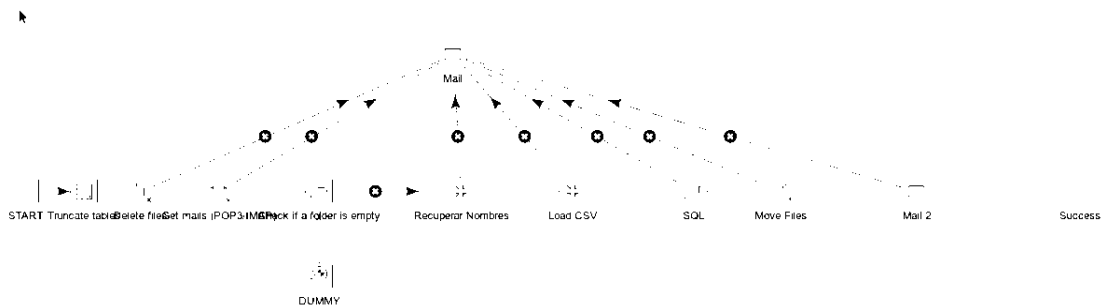


Imagen 13 - Kettle integración Vozitel

3.1.2.6. Mailing

Motobuykers basa su modelo de marketing en el envío de emails a los clientes. Tanto es así que la presión comercial sobre los clientes supone dos envíos de email para cada usuario y día. Esta información viene facilitada, principalmente, a través de dos procesos: datos agrupados y datos de facturación. Estos dos procesos fueron implementados anteriormente al trabajo de final de grado y,

por tanto, queda fuera de estudio. En añadido, los datos que se integrarán en Qlik son:

id_campana	id_mailing	name_mailing	send_date_mailing	weblink	total_sent	total_accepted	hardbounces	softbounces
5	4795	20160531_seg_moc_s_m_4	2016-05-31 20:26:19	mailto:res.moc@ovps.com?subject=...	43237	43223	0	14

Imagen 14 - Tabla listado_mailings_summary

A modo de resumen los datos del proveedor de email son indicadores como la cantidad de usuarios seleccionados, enviados, filtrados, abiertos, etc.

3.1.2.7. Retargeting

En cuanto al sistema de retargeting se busca disponer de un control de las compras reales provenientes de los diferentes afiliados. En el sector e-commerce este tema siempre es problemático debido a que cada plataforma solo tiene información parcial sobre la última interacción del usuario.

Los dos partners a los cuales se realiza seguimiento son: Criteo y Webgains. En cada uno de los casos se ha creado un proceso que revisa un buzón de correo corporativo en el cual las plataformas envían los resultados de las campañas:

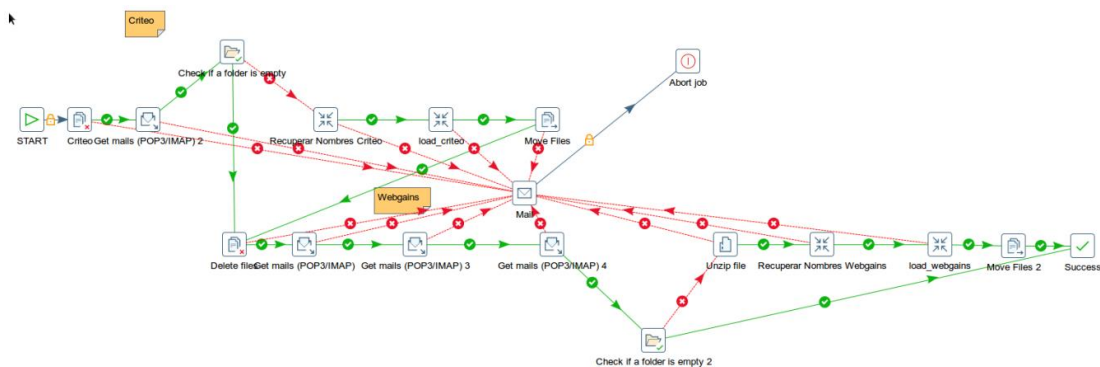


Imagen 15 - Integración retargeting en Kettle

En concreto, los datos facilitados por la plataforma de Webgains son:

- Id de pedido
- Afiliado de conversión
- Importe

- Fecha de transacción
- Fecha de click

Sobre Criteo los datos a integrar son:

- Id de pedido
- Fecha de transacción
- Fecha de click
- Importe

3.1.3. Enriquecimiento de datos

Una vez se ha construido los procesos de integración de datos significa que a partir de ese momento empezaremos a guardar la información relativa a las actuaciones de cada plataforma.

Durante los primeros días de cada proceso se inicio una fase de Q&A para determinar si el funcionamiento de la integración era adecuada o si se habían producido errores.

Una vez determinados los resultados positivos de la carga de datos se paso a la fase de adquisición de datos históricos. Como es normal, este proceso de realizó por cada uno de ellos desde que se dispone de información.

Es importante comentar que de algunos de su pudo acceder al histórico debido a no ser facilitado por la misma plataforma. Es el caso de NetRivals, la API que facilitan solo facilita de la información actual, en el momento de la consulta. Por tanto, la información relativa a cambios de precios solo está disponible des del 16 de febrero de 2016.

En lo relativo a Qlik Sense se ha optado por una solución mixta. Qlik Sense es capaz de cargar datos parciales de la base de datos. Esta situación es ideal para los casos de empresa que dispongan de muchos años de dato historificados que se sabe que del cierto que no van a cambiar. En el caso de Motobuykers es posible que datos de hace más de un año sean cambiados,

por ejemplo por devoluciones por garantía de productos, por tanto se considera que la opción ideal es la anteriormente descrita.

Al detalle podemos decir que se ha configurado una relación de horas de actualización de datos en las que se recarga la información de los archivos de datos. En contraposición, activando una opción de Qlik en el momento de hacer cambios en memoria, nos permite recargar solo estos datos sin tener acceso a la base de datos.

Horas de actualización:

- 7:00h
- 9:00h
- 11:00h
- 13:00h
- 15:00h
- 17:00h
- 19:00h
- 21:00h
- 23:00h

Para no iniciados, el sistema Qlik Sense permite atacar directamente a base de datos o también hacerlo a archivos propietarios de Qlik que son comprimidos y de rápido acceso. En nuestro caso, como decimos, los archivos son generados en determinados momentos del día para mejorar la eficiencia del servidor.

En a lo que respecta a WeRide el proceso de carga de datos se realiza de la misma manera, cargas de datos completas que se pueden activar o desactivar a gusto del programador. Eso sí, la aplicación de WeRide no requiere de una actualización tan frecuente de datos y que, por tanto, solo se actualiza una vez cada día: 09:00h.

La recarga de los datos se realiza desde la misma plataforma de Qlik Sense sobre su back-end para desarrolladores. Tal como vemos en la siguiente captura este es el aspecto que presenta actualmente:

Start ▾

Tasks

Tasks Total: 5 Showing: 5 Selected: 1

Name	Associated resource	Type	Enabled	Status	Last execution	Next execution
Actualizar Dashboard W..	WeRide	Reload	Yes	✓ Success	2016-06-01 10:00	2016-06-02 10:...
Manually triggered reloa..	Dashboard Trabajando	Reload	Yes	✗ Aborted	2016-05-10 10:30	
Reload License Monitor	License Monitor	Reload	Yes	✓ Success	2016-06-01 19:40	2016-06-01 20:...
Reload Operations Mon...	Operations Monitor	Reload	Yes	✓ Success	2016-06-01 19:40	2016-06-01 20:...
Task Dashboard	Dashboard Temporal Pu...	Reload	Yes	🔄 Started	2016-06-01 20:00	2016-06-01 22:...

Imagen 16 - Back-end de tareas de Qlik Sense

En la siguiente imagen podemos observar las diferentes iteraciones que han sido creadas para Motobuykers y WeRide:

Reload task edit

IDENTIFICATION

Name: Actualizar Dashboard WeRide

App: WeRide

EXECUTION

Enabled:

Task session timeout (minutes): 1440

Max retries: 0

Triggers ⏪ **Actions** ▾

Name	Type	Enabled
Daily 09:00h	Schedule	Yes

Imagen 17 - Actualización de WeRide

Reload task edit

Name

App

EXECUTION

Enabled

Task session timeout (minutes)

Max retries

Triggers Actions ▾

Name ▾ ⌵	Type ⌵	Enabled ⌵
Daily 07:00	Schedule	Yes
Daily 09:00	Schedule	Yes
Daily 11:00	Schedule	Yes
Daily 13:00	Schedule	Yes
Daily 15:00	Schedule	Yes
Daily 17:00	Schedule	Yes
Daily 19:00	Schedule	Yes

Imagen 18 - Actualización de Motobuykers

3.1.4. Modelo de datos final

Tal como avanzábamos anteriormente el modelo de datos es una combinación de una gran tabla plana con el modelo estrella. El resultado de integrar las diferentes fuentes de datos es:

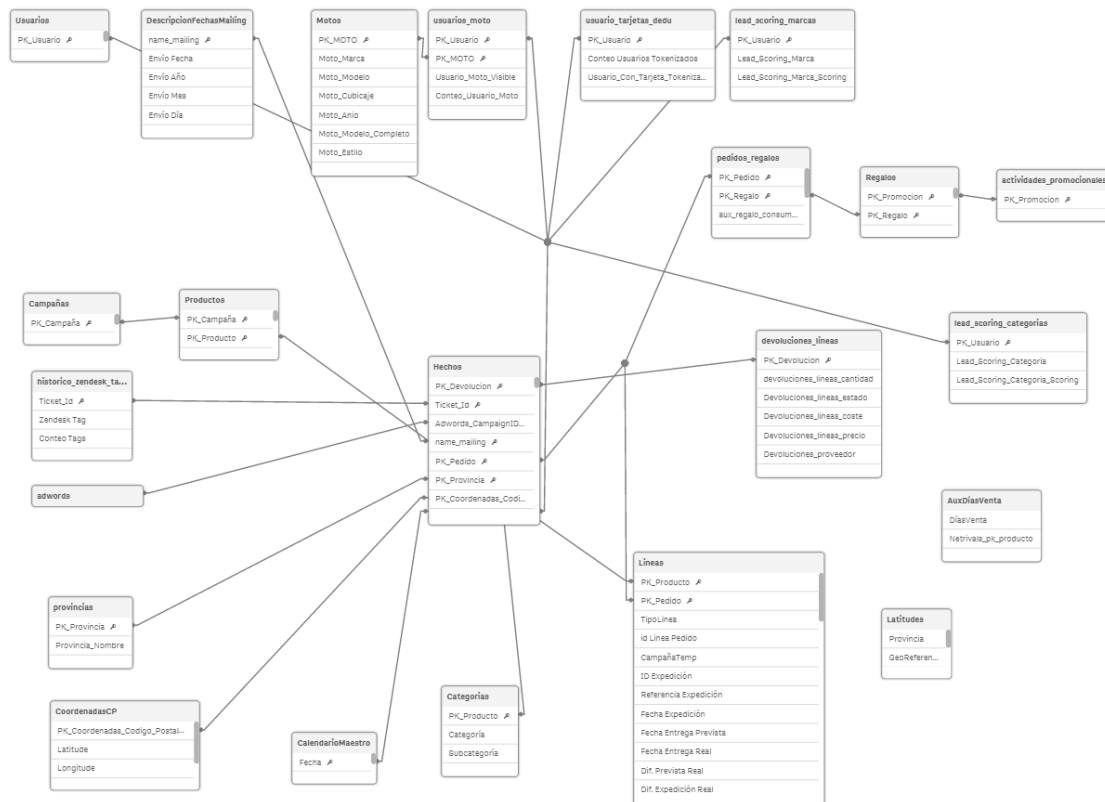


Imagen 20 – Esquema de datos

Aunque pueda parecer el típico modelo de estrella si nos fijamos veremos que hay multitud de tablas que hacen referencia a una tabla central, hecho no frecuente en ese modelo.

La tabla ‘Hechos’ es el eje central de nuestra solución de modelo de datos. En efecto, en esta tabla están integrados (unidas) las diferentes tablas de todas las fuentes de datos. Tal como hemos dicho, esta situación permite que varias fuentes compartan información en columnas y que, por tanto, puedan ser relacionadas.

Tal como es visible en la imagen, la tabla ‘Hechos’ es complementada por tablas auxiliares que nos sirvan para añadir información complementaria o información organizada en 3FN. Cabe destacar que para que un filtro de Qlik actúe sobre varios tipos de información ésta debe ser compartida entre todos los grupos de información. Por tanto, por norma general, se debería colocar la información a filtrar en la tabla de ‘Hechos’.

3.2. Visualización de información

En las siguientes páginas pasaremos al detalle de las hojas que se han creado para entender su significado y objetivo. Como veremos, el negocio de Motobuykers se basa en su plataforma web de venta y que, por tanto, la solución propuesta para este es más completa y compleja que para WeRide.

3.2.1. Motobuykers

El dashboard creado para Motobuykers consiste en todo un seguido de hojas separadas por departamentos, diferenciadas por color. Su objetivo es llegar a tener una visión de toda la empresa segmentando los objetivos para cada departamento. Por todo esto, el aspecto final del dashboard de Motobuykers es:



Imagen 21 - Dashboard Motobuykers

Tal como se puede ver en la imagen el dashboard está compuesto por cuatro hojas de interés general, diez hojas creadas para el departamento Comercial, ocho hojas para el departamento de operaciones, veinte hojas para el departamento de marketing y dos hojas para el departamento de Finanzas.

Vale la pena remarcar que los datos visualizados forman parte de una batería de pruebas, por tanto, no son reales aunque sí fidedignos.

Acto seguido pasaremos al detalle de cada una de las hojas

3.2.1.1. Estado general de la empresa

Esta agrupación de hojas se ha desarrollado con el objetivo de tener una visión a nivel de pájaro de la empresa. De esta forma se pretende conseguir que en un primer vistazo el usuario diponga de información simplificada de lo que está sucediendo en la página web en relación a las ventas.

3.2.1.1.1. Dashboard

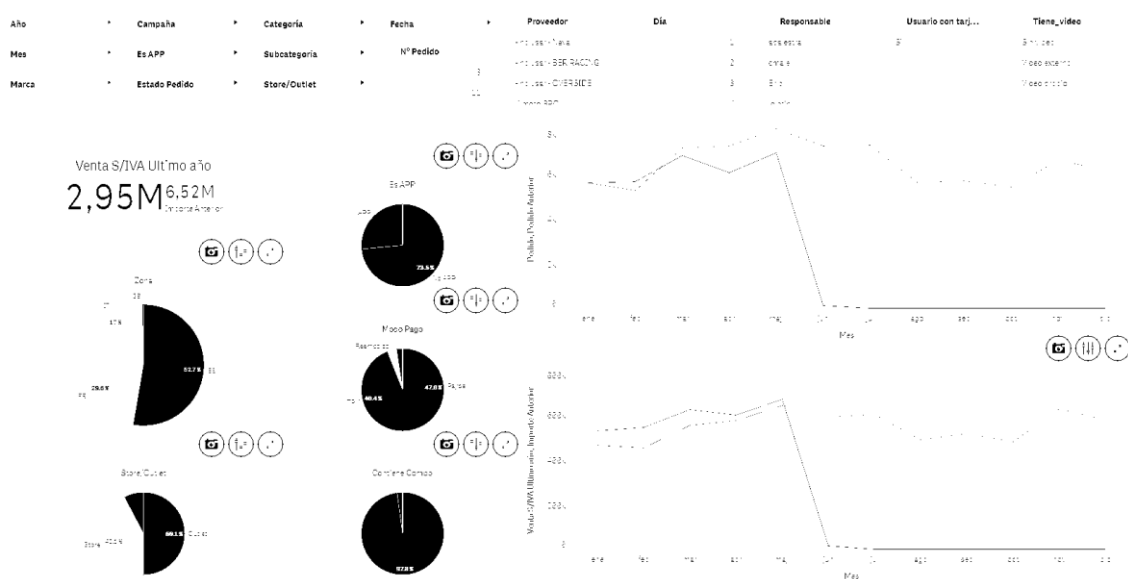


Imagen 22 - Dashboard Motobuyers - Dashboard

Esta hoja representa una visión general de la situación de la empresa. A la derecha podemos ver la comparación de los datos de compras e importe de este año (azul) respecto al año anterior (rojo).

Además, podemos ver como se han añadido distintos indicadores que nos sirven para tener una visión clara y concisa del estado de las compras, por

ejemplo en el gráfico de tarta segmentado por la tipología de compra: Store, Outlet o Mixta.

Algunas de los gráficos mostrados en esta hoja utilizan las propiedades de profundidad. Es el caso del gráfico de tarta de modo de pago. Presionando el sector de PayPal podemos visualizar el detalla sobre la tipología del pago de PayPal:

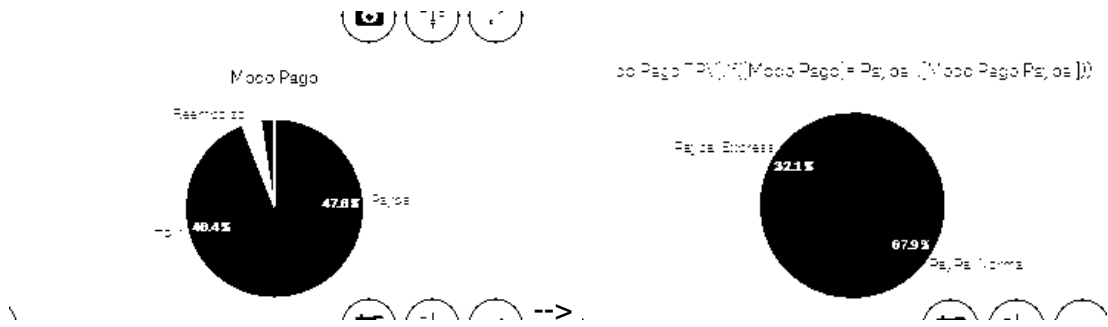


Imagen 23 - Tipología PayPal

También podremos observar, como en la mayoría de hojas, la gran cantidad de filtros que pueden ser aplicados en la hoja. Activando cualquiera de ellos el sistema actualizará todas las gráficas del sistema.

3.2.1.1.2. Ventas y Abonos por Zona



Imagen 24 - Dashboard Motobuyers – Ventas y Abonos por Zona

Igual que en la hoja anterior, la hoja actual nos permite tener una visión completa de las ventas sobre la cantidad de devoluciones que se han generado.

3.2.1.1.3. Business plan

Año	Zona	Mea																								
	Ente	Ene-Obj	Feb	Feb-Obj	Mar	Mar-Obj	Abr	Abr-Obj	May	May-Obj	Jun	Jun-Obj	Jul	Jul-Obj	Ago	Ago-Obj	Sep	Sep-Obj	Oct	Oct-Obj	Nov	Nov-Obj	Dic	Dic-Obj	Actual	
Memoria	323.0	798.05	828.0	807.456	857.2	816.85	846.0	876.25	876.0	835.65	858.0	845.05	836.0	854.45	836.0	883.85	836.0	873.25	836.0	882.85	836.0	901.45	836.0	901.45	836.0	876.00
Compras-Miembros	0.0%	0.82%	0.82%	0.76%	0.71%	0.73%	0.83%	0.82%	0.74%	0.81%	0.81%	0.80%	0.82%	0.79%	0.82%	0.81%	0.82%	0.75%	0.82%	0.75%	0.82%	0.75%	0.82%	0.75%	0.82%	
Total Venta	1.18%	1.18%	1.18%	1.14%	1.13%	1.20%	1.13%	1.23%	1.16%	1.24%	0	1.24%	0	1.24%	0	1.86%	0	1.23%	0	1.24%	0	1.23%	0	1.23%	0	
Revoluciones	0.32%	0.56%	0.55%	0.54%	0.55%	0.58%	0.52%	0.60%	0.55%	0.60%	0	0.60%	0	0.60%	0	0.56%	0	0.59%	0	0.59%	0	0.60%	0	0.60%	0	
Investment	14.36	33.316	16.86	28.083	16.99	29.083	19.63	29.883	17.62	30.383	0.89	29.250	0.89	29.250	0.89	29.950	0.89	29.250	0.89	28.850	0.89	28.650	0.89	28.650	0.89	28.650
Share-Improvement	18.88	15.65	18.87	19.48	19.37	22.58	18.69	22.94	21.43	22.51	0	23.40	0	23.40	0	25.72	0	23.01	0	23.53	0	23.79	0	23.79	0	23.79
Orders-Member	1.11	1.10	1.11	1.10	1.10	1.11	1.10	1.11	1.10	1.11	0	1.10	0	1.10	0	1.10	0	1.10	0	1.10	0	1.10	0	1.10	0	1.10
Price-Member	94.10	95.89	95.87	96.16	97.54	97.20	98.16	97.23	96.71	97.22	131.0	97.22	0	97.22	0	97.14	0	97.22	0	97.11	0	97.24	0	97.24	0	97.24
Conversion	3.62	4.82	3.86	5.20	3.13	5.20	3.92	5.20	3.67	5.20	1.73	5.20	0	5.20	0	5.20	0	5.20	0	5.20	0	5.20	0	5.20	0	5.20
Buyers	3.01	4.98	3.01	5.69	2.89	6.47	3.10	6.75	6.07	6.74	0	6.75	0	6.75	0	6.75	0	6.63	0	6.64	0	6.71	0	6.71	0	6.71
Orders	3.33	5.43	3.33	6.11	2.77	7.13	3.20	7.42	6.01	7.47	0	7.42	0	7.42	0	7.42	0	7.28	0	7.51	0	7.51	0	7.51	0	7.51
Total Sales	323.0	821.30	828.0	858.77	828.0	858.74	893.0	884.86	866.0	883.05	1.17	884.53	0.89	886.17	0.89	938.10	0.89	872.88	0.89	874.12	0.89	881.58	0.89	881.58	0.89	881.58
Returns	83.42	87.39	87.69	79.57	89.81	84.77	88.73	85.47	88.63	83.95	3.91	73.10	0.89	87.55	0.89	75.61	0.89	75.92	0.89	74.59	0.89	78.85	0.89	78.85	0.89	78.85
Results	12.11%	11.97%	12.03%	13.17%	12.85%	12.91%	11.55%	12.45%	12.82%	12.27%	17.81%	10.58%	0	12.75%	0	14.18%	0	11.28%	0	11.87%	0	11.57%	0	11.57%	0	11.57%
Net Sales	239.2	488.91	483.0	485.28	368.0	571.96	324.0	569.18	307.3	600.08	0.27	611.42	0.89	598.61	0.89	488.59	0.89	597.05	0.89	599.52	0.89	602.83	0.89	602.83	0.89	602.83

Imagen 25 - Dashboard Motobuyers – Business Plan

Esta hoja muestra una relación de diversos indicadores, como puede ser la facturación y los compara con el dato objetivo mes a mes. Por una parte, la tarea ha sido realizado en tablas para así poder exportar los datos a excel. Por otro lado, la creación de la hoja se ha realizado en dos tablas diferentes debido a limitaciones de Qlik en la longitud de fórmulas.

3.2.1.2. Departamento Comercial

El objetivo de esta agrupación de hojas se basa en dos pilares por un lado disponer de información agrupada sobre diversos indicadores y, por otro lado, visualizar información al detalle sobre diferentes valores de las transacciones.

3.2.1.2.1. Ventas, CMg: proveedores

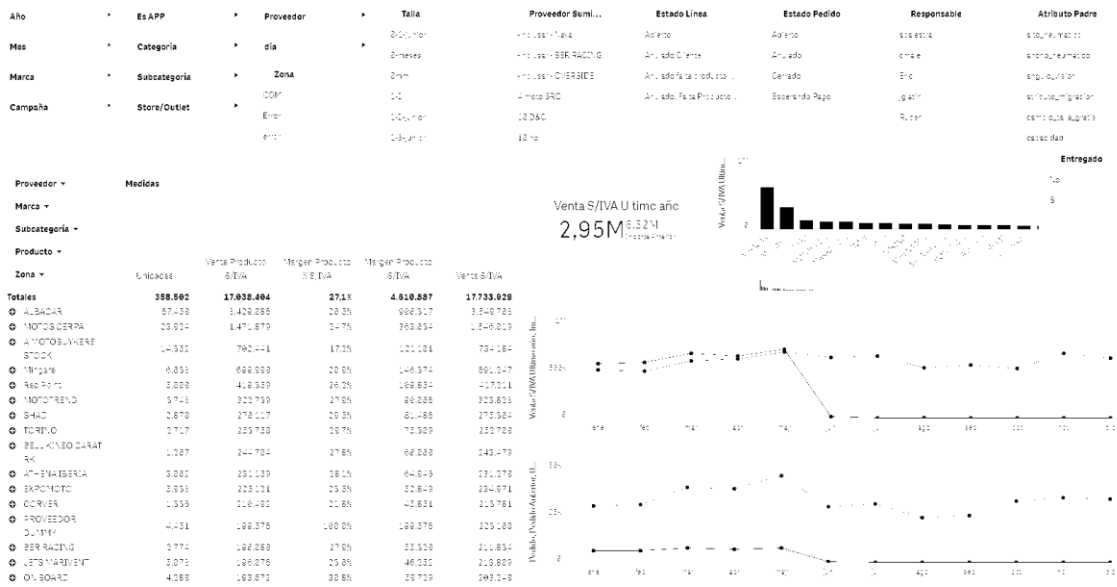


Imagen 27 - Dashboard Motobuyers – Ventas, CMg: proveedores

Empezamos la sección de comercial. Tenemos delante la hoja que permite al equipo comercial conocer de primera mano el detalle de venta por proveedor y la evolución respecto al periodo anterior.

3.2.1.2.2. Ventas, CMg: listado productos

Año	Es APP	Proveedor	Zona	Talla	Proveedor Semi...	Estado Línea	Estado Pedido	Responsable	Atributo Padre
Mes	Categoría	día	OSM	8/10/11	Proveedor Semi...	Estado Línea	Estado Pedido	Responsable	Atributo Padre
Marca	Subcategoría		EMR	8mm	Proveedor Semi...	Estado Línea	Estado Pedido	Responsable	Atributo Padre
Campaña	Store/Outlet		FR	10	Proveedor Semi...	Estado Línea	Estado Pedido	Responsable	Atributo Padre

Producto	Jrda des	Venta Producto S/C	Margen Producto S/C	Margen Producto S/C	Venta S/IVA	Margen S/IVA	Margen S/IVA	Precio Medio C/IVA	Coste Producto S/C	Marca	Subcategoría	Referencia Mto	Proveedor
Totales	358.502	17.035.404	27,1%	4.810.887	29	5.307.104	29,0%	59,81	12.427.517				
PRODUCTO DUMNY	489	10.889	37,5%	3.089	13.250	6,34%	49,4%	20,51	6.005	MARCA DUMNY		1102155	ALBACAR
PRODUCTO DUMNY	230	16.838	23,5%	4.181	16.001	3,72%	81,3%	60,35	12.479	MARCA DUMNY		108026201	ALBACAR
PRODUCTO DUMNY	359	17.541	32,2%	3.594	17.085	3,74%	50,7%	69,51	11.750	MARCA DUMNY		PERFORMANCEVALROLD	MOTOS CERPA
PRODUCTO DUMNY	352	3.488	39,3%	1.139	6.005	3,09%	49,6%	38,79	3.507	MARCA DUMNY		POSHADE	TELEKINEO CARAT 541
PRODUCTO DUMNY	216	10.833	24,6%	4.029	26.031	6,04%	28,5%	111,01	14.804	MARCA DUMNY		SHAD02M	BER RACING
PRODUCTO DUMNY	209	12.579	18,3%	2.850	16.330	4,01%	24,3%	68,00	12.340	MARCA DUMNY		GRUPOPERFORMANCE	MOTOS CERPA
PRODUCTO DUMNY	176	4.016	27,2%	1.146	3.040	1,97%	39,2%	25,91	3.070	MARCA DUMNY		ACREPUL00934	ALBACAR
PRODUCTO DUMNY	169	44.114	5,1%	1.004	44.780	2,03%	8,5%	119,03	41.830	MARCA DUMNY		GRUPOPERFORMANCE	MOTOS UNIVERSE STOCK
PRODUCTO DUMNY	130	1.110	20,4%	227	1.070	3,87%	30,3%	8,49	881	MARCA DUMNY		MOTON	ALBACAR
PRODUCTO DUMNY	149	3.200	23,3%	817	4.101	1,71%	41,5%	23,86	2.381	MARCA DUMNY		ACREPUL00934	ALBACAR
PRODUCTO DUMNY	141	5.366	12,3%	440	4.193	2,03%	23,2%	90,60	5.150	MARCA DUMNY		ACREPUL00934	ALBACAR
PRODUCTO DUMNY	134	7.850	21,4%	1.804	8.000	2,49%	30,2%	66,91	9.765	MARCA DUMNY		FR020	TELEKINEO CARAT 541
PRODUCTO DUMNY	130	2.130	20,6%	374	2.842	1,36%	45,2%	16,82	1.335	MARCA DUMNY		ACREPUL00934	ALBACAR
PRODUCTO DUMNY	130	2.039	30,1%	759	2.000	1,50%	51,2%	18,02	1.300	MARCA DUMNY		TELEKINEO CARAT 541	TELEKINEO CARAT 541
PRODUCTO DUMNY	128	9.641	31,8%	1.049	4.374	1,87%	42,0%	37,60	2.055	MARCA DUMNY		POSHADE	TELEKINEO CARAT 541
PRODUCTO DUMNY	108	1.481	30,6%	457	1.739	7,70%	43,0%	19,31	947	MARCA DUMNY		GRUPOPERFORMANCE	GRUPOPERFORMANCE
PRODUCTO DUMNY	139	2.139	40,3%	669	3.787	1,84%	39,1%	20,39	1.140	MARCA DUMNY		SHAD02M	MOTOS UNIVERSE STOCK

Imagen 28 - Dashboard Motobuyers – Ventas, CMg: listado productos

Hoja auxiliar que, una vez marcados los filtros en las otras hojas, nos servirá para exportar los datos de rendimiento en formato Excel.

3.2.1.2.3. Ventas, CMg: campañas



Imagen 29 - Dashboard Motobuyers – Ventas, CMg: campañas

En el mismo sentido que la hoja de proveedores, en este caso estamos delante de una creación para realizar el seguimiento de las diferentes campañas outlet que realiza la web. Tal como muestra la gráfica de barras apilada de la derecha podemos ver como se repartía la cesta años anteriores y como se reparte ahora.

3.2.1.2.4. Ventas, CMg: marca-producto

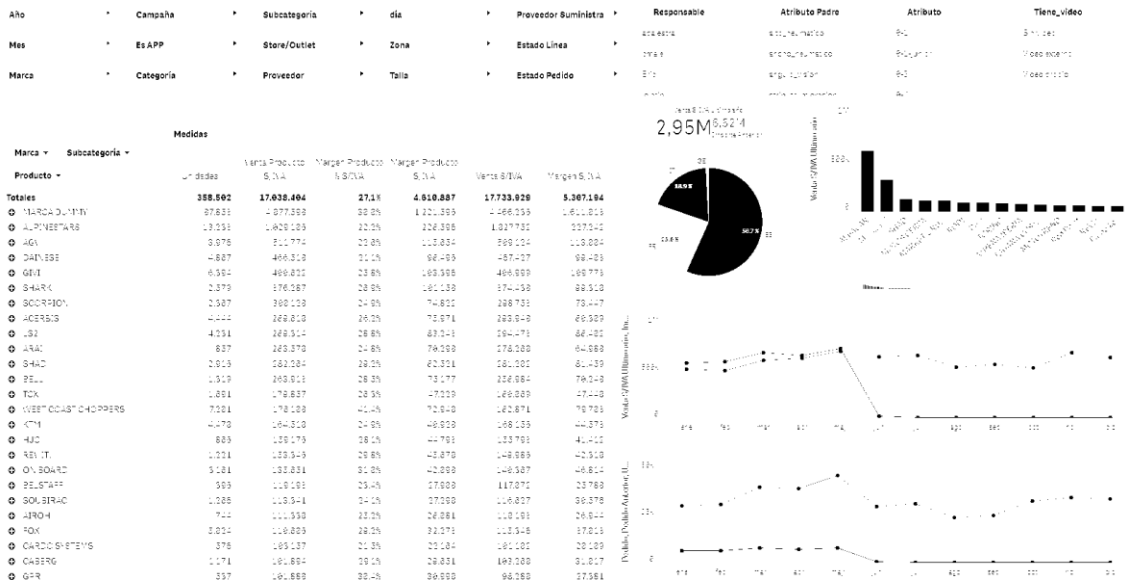


Imagen 30 - Dashboard Motobuyers – Ventas, CMg: marca-producto

Muestra información similar a las anteriores pero enfocado a la visualización del seguimiento por marca y producto.

3.2.1.2.5. Campañas: ventas año-mes-día

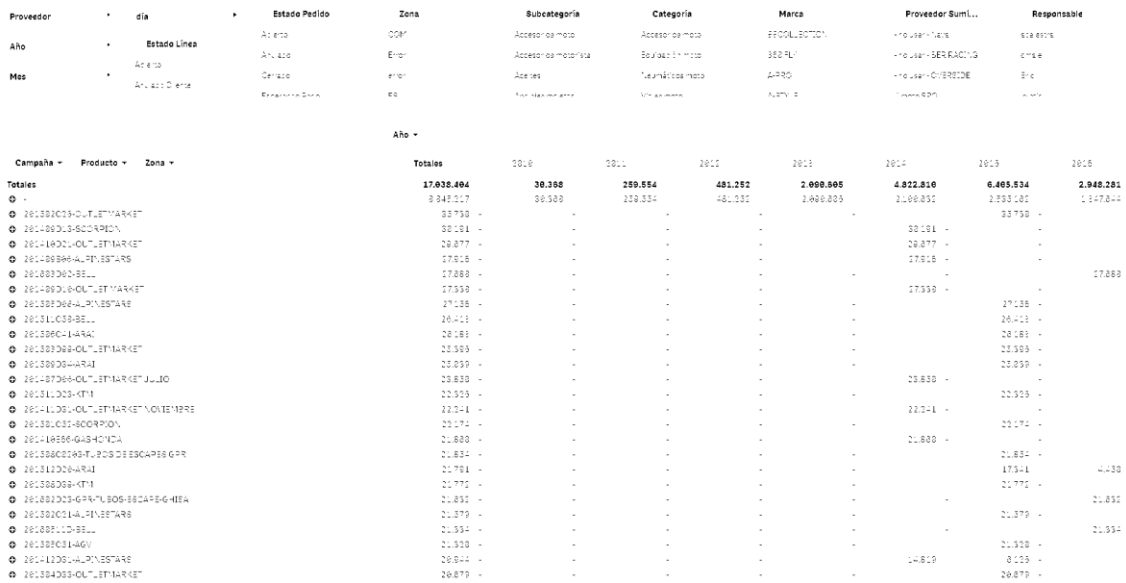


Imagen 31 - Dashboard Motobuyers – Campañas: ventas año-mes-día

En este caso mostramos la información de seguimiento sobre las campañas pero indicando en las columnas la facturación.

3.2.1.2.6. Marca: ventas año-mes-día

Proveedor	día	Estado Pedido	Zona	Subcategoría	Categoría	Marca	Proveedor Sum...	Responsable
Año	Estado Línea	Año	Zona	Subcategoría	Categoría	Marca	Proveedor Sum...	Responsable
Mes	Año	Estado	Zona	Subcategoría	Categoría	Marca	Proveedor Sum...	Responsable

Marca	Subcategoría	Año	Zona						
Producto	Campaña	Totales	O 2010	O 2011	O 2012	O 2013	O 2014	O 2015	O 2016

Totales		17038.404	30.350	259.554	481.252	2.060.905	4.822.810	6.405.534	2.948.281
ALPACALFATIN		4.977.592	30.350	259.554	481.252	2.060.905	4.822.810	6.405.534	2.948.281
ALPACALFATIN		1.402.103	-	-	-	-	227.500	221.702	378.000
ALPACALFATIN		211.772	-	-	-	-	149.189	369.134	69.439
ALPACALFATIN		496.518	-	-	-	-	129.174	223.826	119.319
ALPACALFATIN		496.518	-	-	-	-	114.692	159.687	125.889
ALPACALFATIN		376.387	-	-	-	-	87.311	211.697	78.777
ALPACALFATIN		380.109	-	-	-	-	181.031	142.789	372.091
ALPACALFATIN		288.514	-	-	-	-	182.639	196.581	170.011
ALPACALFATIN		288.514	-	-	-	-	87.879	144.584	372.091
ALPACALFATIN		288.514	-	-	-	-	12.551	144.584	121.802
ALPACALFATIN		288.514	-	-	-	-	86.817	144.584	121.802
ALPACALFATIN		288.514	-	-	-	-	46.842	116.889	169.379
ALPACALFATIN		1.781.209	-	-	-	-	32.309	111.937	321.217
ALPACALFATIN		1.781.209	-	-	-	-	78.018	411.389	281.089
ALPACALFATIN		184.510	-	-	-	-	98.187	83.546	46.677
ALPACALFATIN		129.174	-	-	-	-	173.54	69.319	45.110
ALPACALFATIN		129.174	-	-	-	-	9.174	68.081	33.778
ALPACALFATIN		129.174	-	-	-	-	23.221	62.789	19.881
ALPACALFATIN		119.511	-	-	-	-	1.719	69.270	21.788
ALPACALFATIN		119.511	-	-	-	-	61.837	377.011	14.389
ALPACALFATIN		119.511	-	-	-	-	8.029	69.823	17.039
ALPACALFATIN		111.530	-	-	-	-	23.781	21.710	36.827
ALPACALFATIN		98.002	-	-	-	-	29.020	61.020	6.814
ALPACALFATIN		98.002	-	-	-	-	16.800	49.044	39.439
ALPACALFATIN		181.894	-	-	-	-	26.037	31.020	23.788
ALPACALFATIN		181.894	-	-	-	-	14.871	32.551	33.752

Imagen 32 - Dashbard Motobuykers – Marca: ventas año-mes-día

Igual que la hoja anterior, a nivel de marca, disponemos de la información de facturación respecto a todos los años de los que se posee información.

3.2.1.2.7. Heatmap Campaña-Día

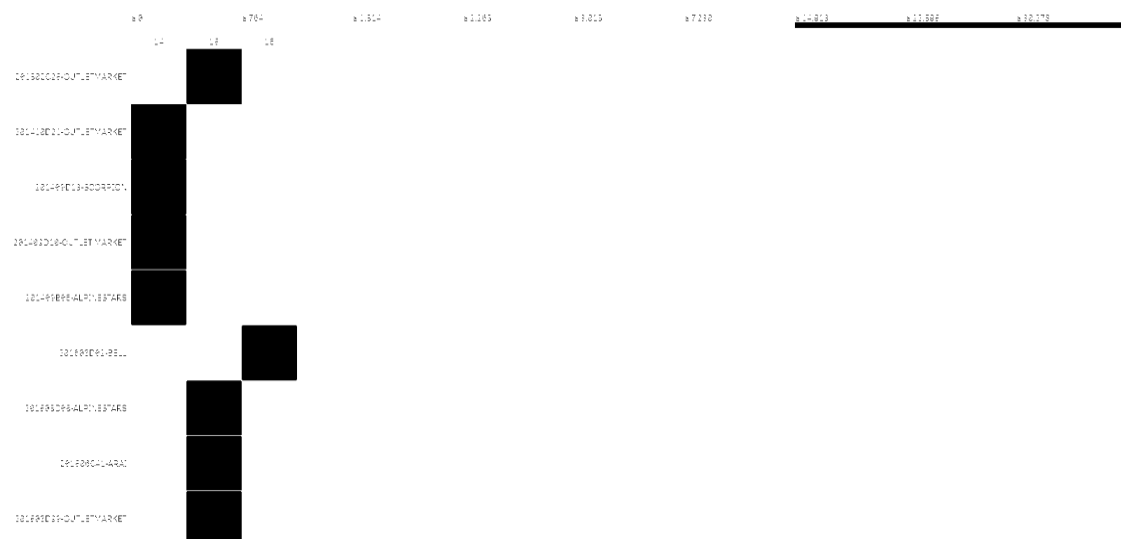


Imagen 33 - Dashboard Motobuykers – Heatmap campañas

Mapa de calor según la facturación de la campaña. La hoja dispone de dimensión de profundidad sobre la fecha (año-mes-día). Por tanto, el usuario puede ir al detalle hasta acabar a nivel de día. Es muy útil poder ver visualmente como las campañas se inician con una buena factura, los días intermedios decae hasta llegar a volver obtener una buena facturación el último día. Este hecho se produce debido a la presión comercial que se realiza al inicio y al final de la campaña. Como decimos, es muy interesante verlo gráficamente.

3.2.1.2.8. Feed productos

Día/Venta	source	Zona	channel/Grouping	Estado producto	Campaña	Producto Store/Ou...	Prod Jotic	Marc e	Subc ateg or
Marca	Venta no stock	COM	Home	Active	281441111-UNAFIS	Store			
Proveedor	10	Empl	Direct	Inactiv	281451142ERIE				
	5	empl	Direct		281338895-REPUBCALCA				
		ca	Dir...		141121111423261411111				

Medidas												Medida Precio
Marca	Producto	# de pedidos	Venta Bruta	Stock Actual	Unidades	Ventas	CR	Margen Bruto	Margen Medio	Valor	Order	Completos
Totales		7965	428.604 €	4760763	358502	17.733.029 €	0,50%	26,00%	31,1%	1573184	10.493	1.51 €
ALPINESTAR		508	33.140 €	126879	12231	1.027.772 €	0,2%	28,2%	22,2%	88284	10481	12,02 €
BSL		108	23.281 €	14878	1319	239.886 €	0,3%	28,2%	28,2%	21914	7346	19,06 €
SACD		171	19.823 €	10231	2013	281.240 €	0,3%	28,2%	28,2%	31108	10812	11,12 €
GRU		131	17.041 €	5372	6562	428.992 €	0,4%	28,2%	27,4%	19784	10233	11,12 €
SHARK		84	12.982 €	22811	2379	371.238 €	0,3%	28,2%	28,2%	18879	8213	17,06 €
DARLSE		148	10.248 €	24511	4887	467.427 €	0,3%	28,2%	21,3%	46320	7719	18,76 €
ASAI		59	10.224 €	2480	837	378.269 €	0,1%	28,2%	28,2%	26487	8733	14,03 €
AGA		77	9.321 €	32388	3978	339.121 €	0,1%	28,2%	22,2%	49136	6594	14,06 €
JS2		139	8.982 €	38872	4231	281.273 €	0,3%	28,2%	28,2%	37487	9718	12,48 €
SCORPIO		71	8.877 €	132713	2387	239.773 €	0,2%	24,3%	21,3%	42121	5883	28,06 €
PUMA		85	8.862 €	27847	1148	678.448 €	0,4%	38,2%	33,3%	21321	17381	
WESTCAST CHOPPERS		440	7.982 €	82381	7281	185.371 €	1,8%	44,2%	43,3%	26824	14214	16,82 €
HOTOCOLR		119	6.518 €	4231	1274	213.448 €	0,3%	34,2%	33,3%	24127	17281	7,69 €
AGERSIS		123	6.514 €	22330	4224	281.948 €	0,4%	27,2%	27,7%	27439	12380	11,87 €
RELT		35	6.211 €	22839	1221	179.882 €	0,1%	27,8%	27,8%	28189	12326	7,62 €
CARDONETTES		34	6.167 €	830	378	181.182 €	0,8%	18,8%	18,3%	3811	12327	12,06 €
BILWELL		120	5.999 €	2123	933	41771 €	0,8%	31,2%	33,3%	13327	12829	18,06 €
AIRON		47	5.948 €	18878	742	110.192 €	0,2%	21,3%	24,2%	18406	7786	21,02 €
FOX		917	5.811 €	42338	3824	128.148 €	1,2%	34,2%	33,3%	23330	8330	
KTM		179	5.562 €	3849	4278	289.123 €	0,8%	24,2%	27,2%	21231	8330	21,14 €
BERK		33	5.089 €	2333	101	39.787 €	0,2%	28,8%	28,7%	12372	8421	
CASBERG		44	4.788 €	28839	1171	185.269 €	0,3%	27,7%	28,2%	14786	10834	36,99 €
PHILIPPOREAD		33	4.761 €	2880	731	41.831 €	0,4%	18,8%	28,2%	42821	8613	
410		36	4.639 €	4539	883	133.799 €	0,4%	28,2%	28,2%	1814	8613	8,86 €
ARROW		19	4.517 €	533	144	82.597 €	0,1%	20,7%	18,2%	11131	9281	33,26 €

Imagen 34 - Dashboard Motobuyers – Feed Productos

Hoja preparada para el seguimiento de los productos complementando información de Google Analytics. Esta hoja permite detectar cambios en el patrón de compras de los usuarios.

3.2.1.2.9. Netrivals

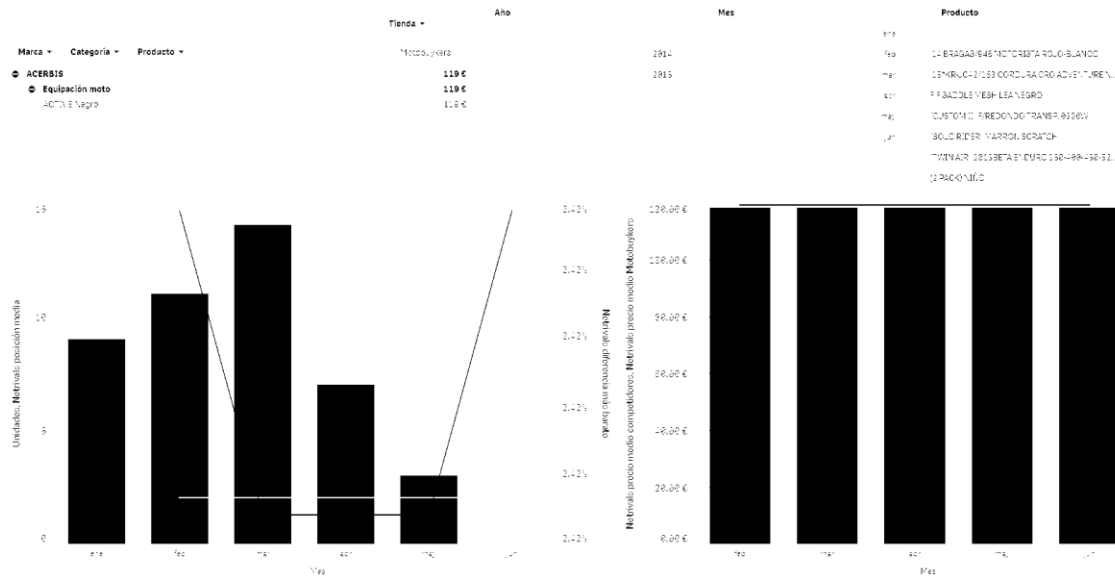


Imagen 35 - Dashboard Motobuykers - Netrivals

Haciendo uso de la información integrada en la base de datos sobre la relación de precios nuestros respecto a la competencia, se ha producido una hoja para poder realizar un seguimiento producto a producto con el objetivo de disponer de información de los precios y unidades vendidas.

3.2.1.2.10. Detalle pedidos

Detalle pedidos

Año (1/7) 2016

Es APP, Proveedor, Zona, Talla, Proveedor Sum..., Estado Línea, Estado Pedido, Atributo Padre

Fecha	Mes	Nº Pedido	Unidades	Venta Producto S/IVA	Margen Producto % S/IVA	Margen Producto S/IVA	Venta S/IVA	Margen S/IVA	Margen % S/IVA	Precio medio C/IVA	Coste Producto S/IVA	Fecha Entrega Real	Referencia Mtbks	Producto	Subcategoría	Estado Línea
Totales			358.502	17.038.404	27,1%	4.610.887	17.733.929	5.307,194	29,9%	59,81	12.427.517					
23/05/2015	may	109307	100	131	14,6%	19	133	21	15,9%	1,60	112	04/06/2014	OUT-CASTACCAMO2700069	Castrol Power 1 Racing 2T (0,125L)	Aceites	Abierto
19/10/2015	oct	446100	90	135	25,3%	34	110	9	8,4%	1,80	191	NoEntregado	ACMOT104413	MOTUL 710 2T 120x125cc	Aceites	Anulado falta producto MTBKS
10/06/2015	jun	304769	40	33	32,8%	11	33	11	32,8%	1,00	22	07/07/2015	703410	TOALLITAS LIMPIADORAS VISERA Y CASCO	Varios	Abierto
23/03/2016	mar	576719	38	250	73,9%	185	250	185	73,9%	7,90	65	12/04/2016	BRV35918	EXS FULL SYNTHETIC 4T 15W50 4L	Aceites	Abierto
25/03/2016	mar	22938	36	446	23,3%	104	476	128	27,3%	14,99	342	NoEntregado	ACELF581	PRODUCTO DUMMY	-	Abierto
24/07/2015	jul	347165	36	263	18,6%	49	237	23	9,6%	8,92	214	07/08/2015	104034	MOTUL 710 2T 1L	Aceites	Abierto
03/03/2015	mar	555498	30	221	18,5%	41	221	41	18,5%	9,00	180	24/03/2016	104034	MOTUL 710 2T 1L	Aceites	Abierto
30/04/2015	abr	607571	30	221	18,5%	41	201	21	10,2%	9,00	180	23/05/2016	104034	MOTUL 710 2T 1L	Aceites	Abierto
23/03/2016	mar	577370	30	53	36,8%	19	59	26	43,2%	2,10	34	31/03/2016	CASTACCAMO2700069	Power 1 Racing 2T 0,125L	Aceites	Abierto
17/02/2016	feb	542255	30	42	18,9%	8	46	12	26,2%	1,70	34	19/02/2016	ACCAMO2700060	POWER 1 RACING 2T 0,125L	Aceites	Abierto
12/03/2016	mar	243266	30	40	13,8%	5	45	10	23,3%	1,60	34	16/03/2015	ACCAMO2700060	POWER 1 RACING 2T 0,125L	Aceites	Abierto
07/11/2014	nov	106134	29	1.909	24,1%	459	1.920	470	24,5%	79,00	1.450	NoEntregado	000099	19/09R17 SLICK	Neumáticos moto	Abierto

Imagen 36 - Dashboard Motobuykers – Detalle pedidos

Hoja que nos servirá para exportar los datos a Excel. Dispone de una tabla preparada para su exportación con todos los campos que se creen necesarios para el trabajo offline.

3.2.1.3. Departamento Operaciones

El departamento de operaciones basa su operativa en todo lo relativo a los hechos que suceden en la empresa. La agrupación de hojas creadas para el departamento comercial se ha desarrollado con la perspectiva de visión de águila a visión al detalle de diversos indicadores de eficiencia. No disponer de estos datos podría suponer un cliente insatisfecho.

3.2.1.3.1. Anulaciones

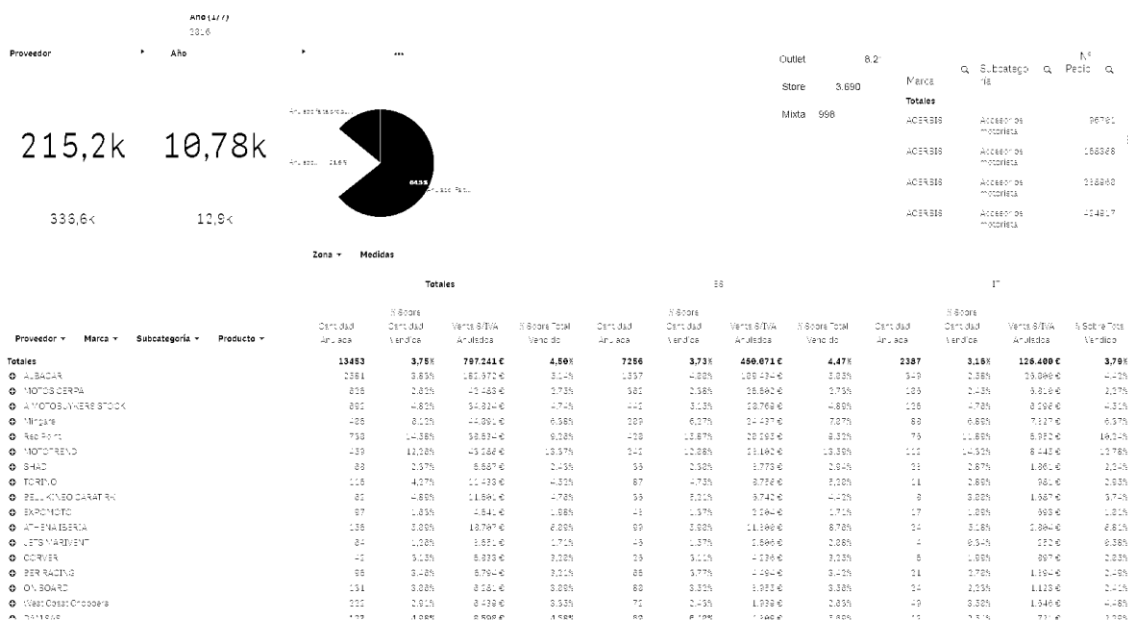


Imagen 37 - Dashboard Motobuykers - Anulaciones

Este hoja recoge la información de las anulaciones que se han efectuado. Se precisa de un gran control sobre este asunto para evitar trabajar con proveedores que habitualmente cancelan los pedidos de los clientes.

3.2.1.3.2. Entregas

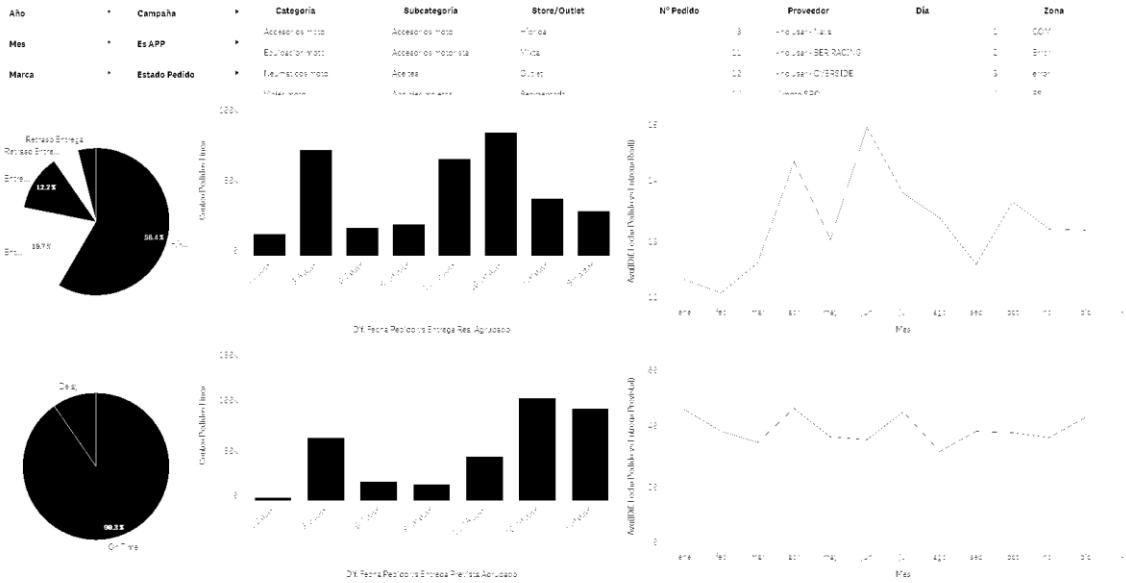


Imagen 38 - Dashboard Motobuykers – Dashboard Entregas

Continue información sobre las entregas de los pedidos. Igual que la hoja anterior, permite tener una visión general y al detalle del tiempo de acción de los pedidos. Y, en consecuencia, saber la satisfacción del cliente.

3.2.1.3.3. Stock

Marca	Campaña	Categoría	Subcategoría	Proveedor	Atributo Padre	Atributo Stock	Atributo
SELECTION	DE CALIDAD PARA	Accesorios moto	Accesorios moto	Proveedor A	Accesorios moto	10	90
SELU	DE CALIDAD PARA	Educación moto	Accesorios motorista	Proveedor B	Accesorios moto	8	90
MARK	DE CALIDAD PARA	Transportes moto	Accesor	Proveedor C	Accesorios moto	8	90
SUMME	DE CALIDAD PARA	Motociclos	Accesorios para	Proveedor D	Accesorios moto	8	90

Campaña	Proveedor	Subcategoría	Atributo	Producto	Stock Actual
-	-	-	-	-	192007
20180223D-91KV	-	-	-	-	140
20180223D-1BATT	-	-	-	-	270
20180223D-4TNY	-	-	-	-	211
20180227A-ROOF	-	-	-	-	493
20180228D-ARAI	-	-	-	-	710
20180228D-PANT-HERA	-	-	-	-	333
20180614K-EKKAJUSI	-	-	-	-	2883
20180614D-DETECT	-	-	-	-	1471
20180615D-TP-GU	-	-	-	-	710
20180615A-TOURIST-TOP-PAVOT-DORRES-AGOSTINI	-	-	-	-	2829
20180616A-88K	-	-	-	-	318
20180616A-90-2PAC	-	-	-	-	809
20180616D-DALISE	-	-	-	-	587
20180617D-ESP-441-D-816-PL-OTOS	-	-	-	-	2897
20180618D-ACERBIS	-	-	-	-	145
20180618D-FIX	-	-	-	-	103
20180618D-HERO-	-	-	-	-	273
20180618F-RELEC-4	-	-	-	-	2030
20180618E-DELROY	-	-	-	-	880
20180618D-SPECIAL-BOOTS	-	-	-	-	119

Imagen 39 - Dashboard Motobuykers - Stock

Hoja que permite el control de los stocks de la compañía. Se usa una tabla pivotante para permitir la exportación y así poder trabajar de manera offline.

3.2.1.3.4. Devoluciones

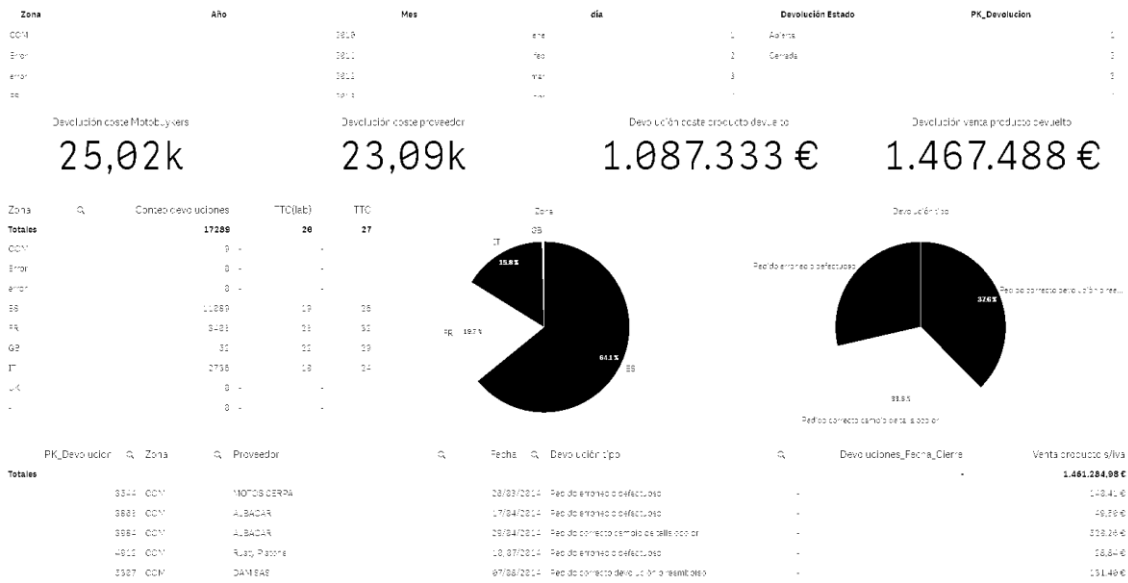


Imagen 40 - Dashboard Motobuyers – Devoluciones

Hoja relativa al control de devoluciones por parte de los usuarios. El objetivo de la hoja es disponer del detalle de devoluciones, así como datos agrupados, para determinar los motivos principales de devolución de productos para buscar una solución.

3.2.1.3.5. Stock Almacén

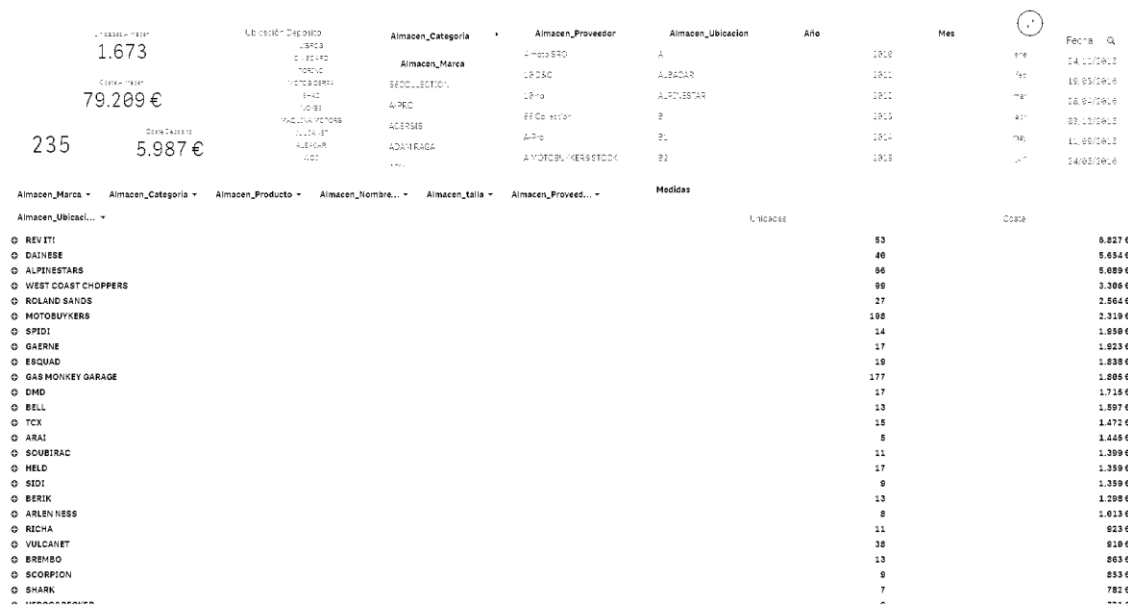


Imagen 41 - Dashboard Motobuyers – Stock Almacén

En la visualización de la hoja nos permite visualizar la cantidad de stock que dispone Motobuyers, a diferencia de la anterior que se trabaja de los stock de los proveedores. Igual que en las anteriores, se realiza sobre una tabla pivotante para permitir la exportación y el trabajo offline.

3.2.1.3.6. Vozitel

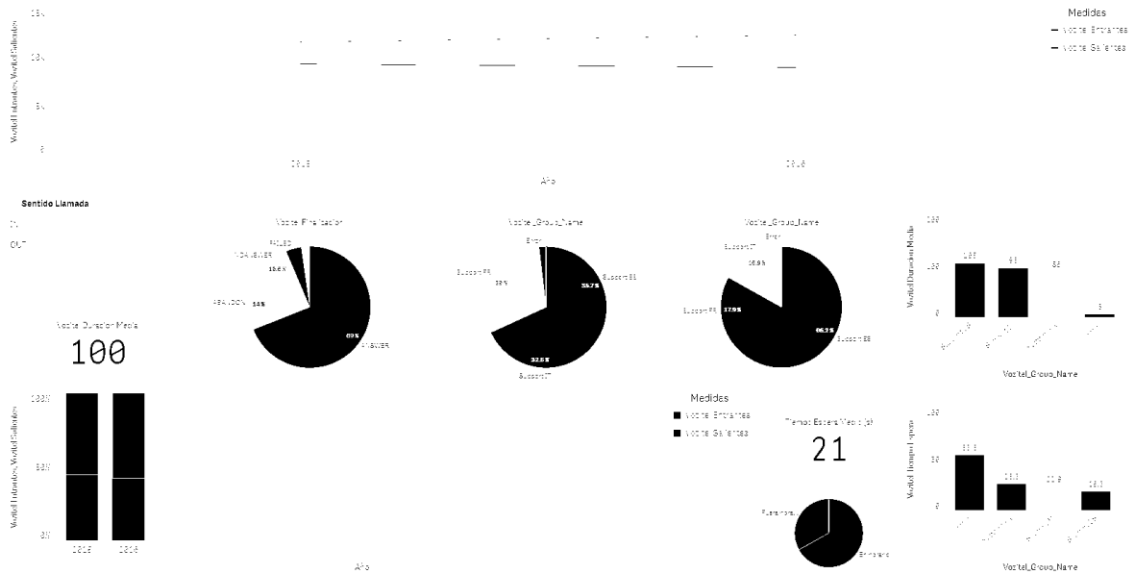


Imagen 42 - Dashboard Motobuyers – Vozitel

Permite el seguimiento de las llamadas gestionadas por el servicio de atención al cliente. Muy interesante poder visualizar los tiempos medio de respuesta por gestor de SAC. Igual que en ocasiones anteriores, algunos de los KPI's son comparados con periodos anteriores.

3.2.1.3.7. Zendesk

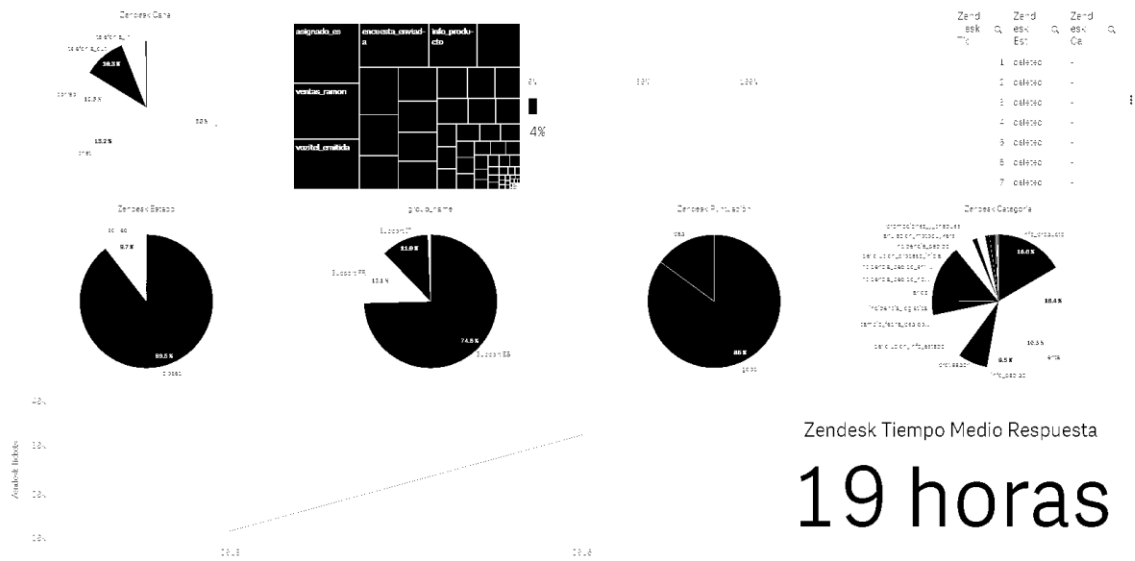


Imagen 43 - Dashboard Motobuyers – Zendesk

Contiene información relativa a la atención al cliente de los que hacen uso de la herramienta Zendesk y Zopim.

Entre otros objetivos, el principal, es permitir disponer del dato de eficiencia en la atención al cliente. Por ejemplo, podemos visualizar el tiempo medio de respuesta hacia el cliente.

3.2.1.3.8. Expediciones



Imagen 44 - Dashboard Motobuyers – Expediciones

Esta hoja muestra información sobre las expediciones que se han efectuado y, sobretodo, el TTR (cantidad de días hasta la recepción) que pasan hasta que el cliente recibe el pedido. Estas expediciones son los envíos que realiza Motobuykers sobre el proveedor logístico, tanto en sentido al cliente como de vuelta a las oficinas. El objetivo de la hoja es visualizar la eficiencia del sistema de envío.

3.2.1.4. Departamento Marketing

El objetivo de las hojas para el departamento de Marketing nace con el objetivo de disponer de una visión de los atributo del usuario.

Por otra parte, tal como se verá a continuación, se busca la agrupación de KPI's sobre la agrupación por canal facilitada por Google Analytics.

3.2.1.4.1. Business plan – Canal

Año	Mes	dia											
	2018		ene										
	2018		feb										
	2018		mar										
	2018		abr										
	2018		may										
	2018		jun										
	2018		jul										
Medidas													
Zona				Compra		Venta S.D.A		DifVenta S.D.A		CR	Objetivo CR	DifCR	
channelGrouping	566 channelGrouping	Objetivo 566 channelGrouping	Dif 566 channelGrouping	Compras	Compras	Venta S.D.A	Venta S.D.A	DifVenta S.D.A					
Totales	5.754.896	191.846	2994,48%	187199	1.095	18978,14%	17.733.829 €	181.826 €	1.7289,87%	1,88%	0,53%	256,18%	
0 ES	1.824.831	182.114	2.797,68%	58204	357	18866,97%	12.029.768 €	53.093 €	18191,04%	1,88%	0,53%	276,48%	
0 FR	1.461.187	21.847	2.756,82%	26701	515	8366,95%	4.177.223 €	12.126 €	17981,13%	1,82%	0,42%	196,38%	
0 IT	1.226.871	38.432	3.445,81%	26773	143	14188,27%	3.339.468 €	14.122 €	2.9546,54%	1,83%	0,42%	383,87%	
0 GP	12.829	2.181	458,54%	807	8	14676,81%	111.862 €	826 €	18271,78%	7,54%	0,28%	2527,72%	
0 CDY	48.789	0		318	0		54.392 €	8 €		8,47%			

Imagen 26 - Dashboard Motobuykers – Business Plan por Canal

Igual que la hoja anterior, esta muestra los KPI's importantes agrupados (a elección) por la agrupación de canal de analytics o por país. Además de comprar con el dato objetivo, el sistema realiza un cálculo, cuando se realizan filtros de meses parciales, calculando si se superará el objetivo o no.

3.2.1.4.2. Newsletter

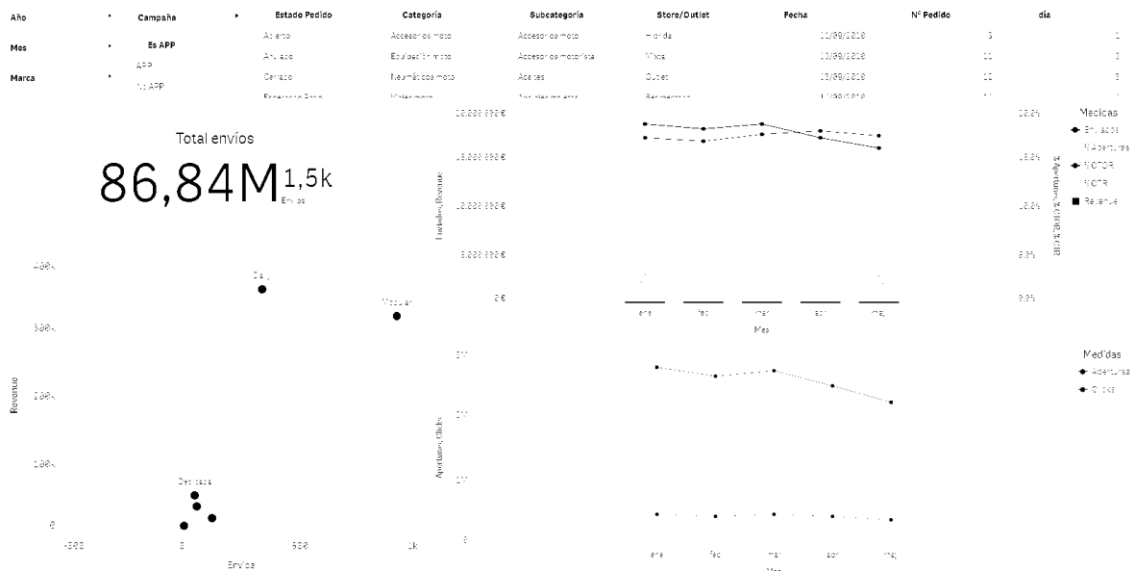


Imagen 45 - Dashboard Motobuykers – Newsletter

Entramos en la sección del departamento de Marketing. Empezamos mostrando información de envíos y su facturación. Podemos observar como disponemos de los porcentajes de aperturas y clicks.

3.2.1.4.3. Newsletter Facturación

Año	Mes	día	Fecha envío año	Fecha envío mes	Fecha envío día
2018	ene	1	2018	ene	1
2018	feb	2	2018	feb	2
2018	mar	3		mar	3
2018	abr	4		abr	4
2018	may	5		may	5
2018	jun	6		jun	6
2018	jul	7		jul	7
2018	ago	8		ago	8

Origen Email	Verde S/CVA	Enviados	Margen S/CVA	Conteo Pedidos	Precio Medio Verde S/CVA	Retorno por email
Totales	1.671.338 €	88836993	448.993 €	19.085	87,57 €	0,687 €
10131127_s_cva_cachifds_2	21.887 €	0	3.486 €	214	118,31 €	0,882 €
10130820_s_cva_academia	10.874 €	0	1.178 €	129	83,42 €	0,228 €
10131225_s_cva_borussia	9.995 €	0	1.146 €	91	108,47 €	0,277 €
10130215_s_cva_moto3denns	9.426 €	11.8137	1.072 €	88	107,82 €	0,276 €
10130328_s_cva1_6	8.764 €	186481	1.059 €	71	123,66 €	0,228 €
10130122_s_cva1_1069	8.277 €	127428	1.099 €	71	116,72 €	0,217 €
10130125_s_cva_moto4x4k1114	8.023 €	127008	1.073 €	68	98,16 €	0,217 €
10130317_s_cva1_4	8.020 €	169249	1.113 €	152	86,91 €	0,229 €
10130321_s_cva_moto3denns	8.342 €	112459	1.066 €	89	131,46 €	0,217 €
10130321_s_cva1_11	8.439 €	166573	1.062 €	89	94,76 €	0,217 €
10130321_s_cva1_3	8.197 €	117319	1.176 €	92	96,18 €	0,226 €
10130413_s_cva1_3	8.187 €	167039	1.736 €	64	88,16 €	0,217 €
10131123_s_cva_moto3denns	8.197 €	0	1.136 €	81	98,79 €	0,229 €
10131869_s_cva_moto3denns	7.992 €	0	1.242 €	84	123,31 €	0,226 €
10130323_s_cva1_11	7.861 €	116124	1.077 €	59	152,33 €	0,228 €
10130787_s_cva1_11	7.897 €	112505	1.777 €	91	86,72 €	0,226 €
10130423_s_cva_moto3denns	7.460 €	286881	1.061 €	38	132,14 €	0,226 €

Imagen 46 - Dashboard Motobuykers – Newsletter Facturación

Hoja de seguimiento de la facturación de newsletter según la fecha de envío del correo. Nos permitirá conocer la cantidad de días que pasa hasta que la newsletter pasa hasta que la newsletter deja de generar acción.

3.2.1.4.4. Newsletter detalle

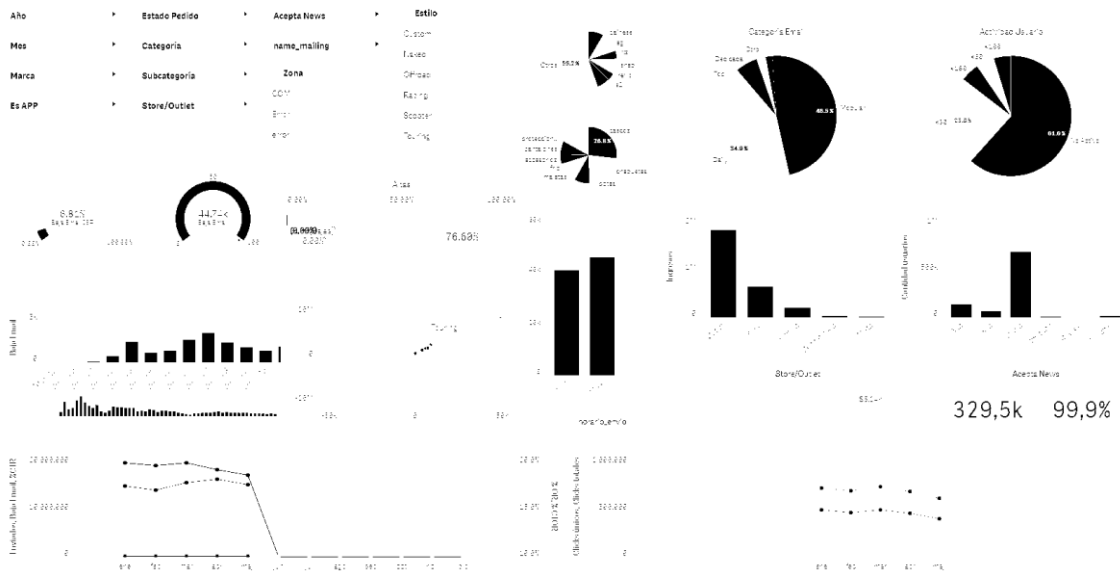


Imagen 47 - Dashboard Motobuykers – Newsletter Detalle

Aglutinación de diferentes KPI's marcados desde el departamento de marketing sobre el rendimiento de las newsletter.

3.2.1.4.5. Fidelización

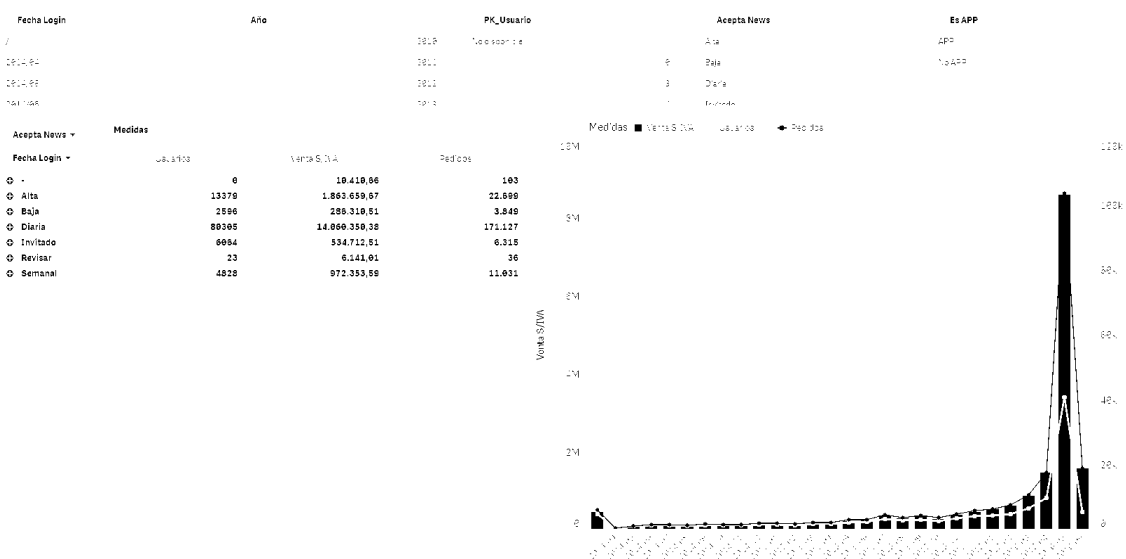


Imagen 48 - Dashboard Motobuykers – Fidelización

Hoja que nos permite visualizar la fidelización del usuario con la marca Motobuykers. En el gráfico de barras de la derecha podemos ver como los usuarios se aglutinan en el mes actual como fecha de su último login en la página web. En los meses anteriores se observa como se van desactivando los usuarios.

3.2.1.4.6. Rowdata

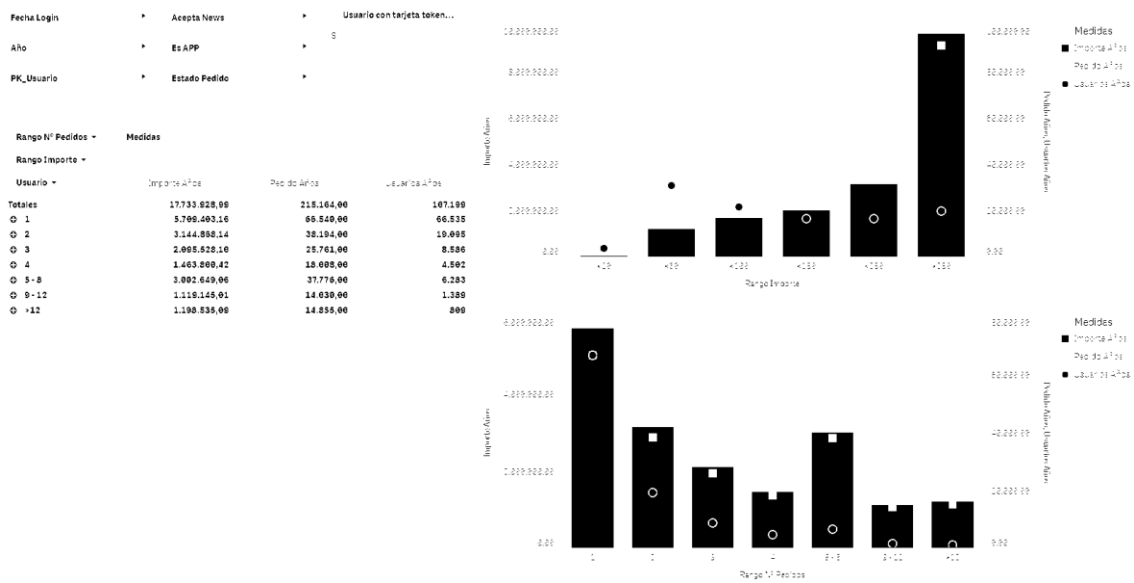


Imagen 49 - Dashboard Motobuykers – Rowdata

Siguiendo un estilo similar al anterior, esta hoja permite visualizar por cantidad de compras y la facturación de los usuarios su grado de fidelidad. El objetivo de esta presentación es analizar los usuarios por cohorts de cantidad de compras y de facturación generada.

3.2.1.4.7. Rowdata Exportar

Fecha Login	Accpta Navs	Estado Pedido	Nº Usuario	Q. Tema	Q. Leng	Q.	Importe Años	Pedido Años	Us. Años Años
Año	Es APP	Adreco	Totales				17.733.928,99	215.184,00	107.199
PK_Usuario	APP	An. app					0.210,52	40,00	0
	Es APP	Cancelado					100,50	3,00	0
		Expirado Desc.					11,76	1,00	1
							152,06	3,00	1
							2.512,66	50,00	1
							607,91	1,00	1
							500,54	10,00	1
							74,02	1,00	1
							74,06	1,00	1
							175,10	3,00	1
							27,26	3,00	1
							106,40	1,00	1
							217,13	10,00	1
							102,10	2,00	1
							200,14	0,00	1
							211,27	1,00	1
							50,37	2,00	1
							-7,37	1,00	1
							310,07	8,00	1
							48,12	1,00	1
							-9,50	1,00	1
							110,06	3,00	1
							103,00	2,00	1
							33,30	1,00	1
							550,71	7,00	1
							109,20	1,00	1

Imagen 50 - Dashboard Motobuykers – Rowdata Exportar

Hoja que permite a los usuarios exportar información personal de los usuarios, los datos han sido borrados por la LOPD.

3.2.1.4.8. Motos

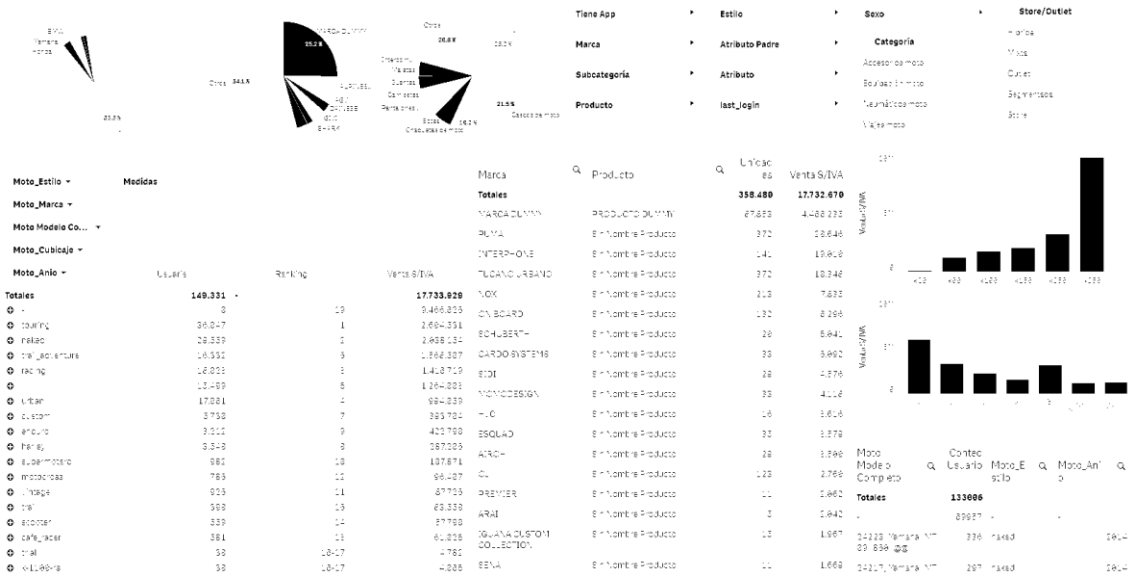


Imagen 51 - Dashboard Motobuykers – Motos

Los usuarios pueden disponer de motocicletas que han introducido en su perfil o, también, se guardan las que realizan búsquedas. En esta hoja se muestra la información de motocicletas con el estilo para visualizar cuáles son las que

tienen más tirada entre los usuarios. Por ejemplo, serviría de muestra para potenciar un tipo de productos u otros.

3.2.1.4.9. Estilos

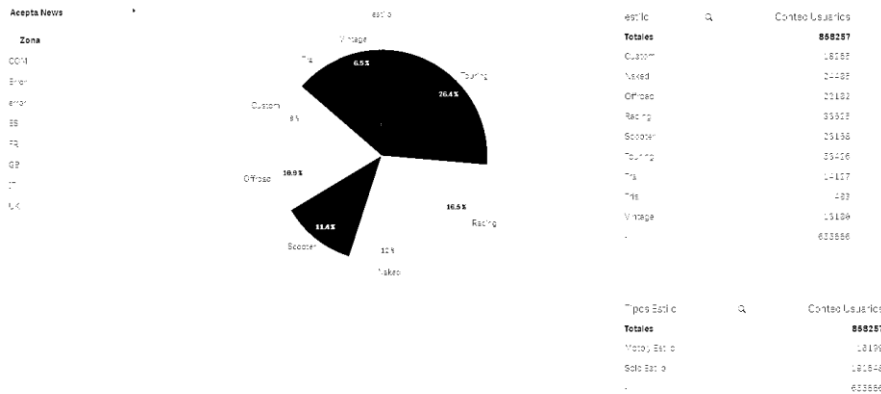


Imagen 52 - Dashboard Motobuykers – Estilos

Hoja similar a la anterior especialmente enfocada a los estilos de los usuarios. Este estilo se recoge por las visitas a páginas de la web así como de las motocicletas de los usuarios.

En análisis previos, la segmentación según el estilo de la motocicleta del usuario ha permitido aumentar la facturación. El objetivo de esta hoja es la visualización de la segmentación de los usuarios según el estilo para poder realizar filtros aplicables en otras hojas.

3.2.1.4.10. RFM

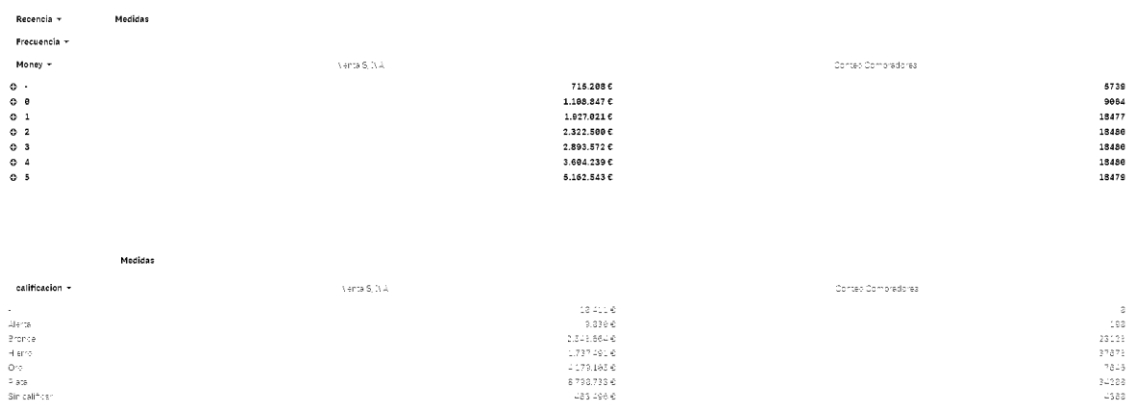


Imagen 53 - Dashboard Motobuykers – RFM

Muestra información del RFM de los usuarios así como su calificación según las compras que han realizado. La idea es aquí realizar los segmentos y exportar usando la tabla de la hoja 'Rowdata exportar'.

El sistema de RFM se basa en la segmentación de los usuarios según su recencia, frecuencia de compra y la compra total realizada. En nuestro caso, habiendo analizado los datos de los usuarios, se consideró oportuno realizar segmentos de quintiles en cada uno de los tres parámetros.

Tal como decimos, la hoja de RFM sirve al departamento de Marketing para visualizar los datos y poder filtrar en el resto de hojas.

3.2.1.4.11. Analytics

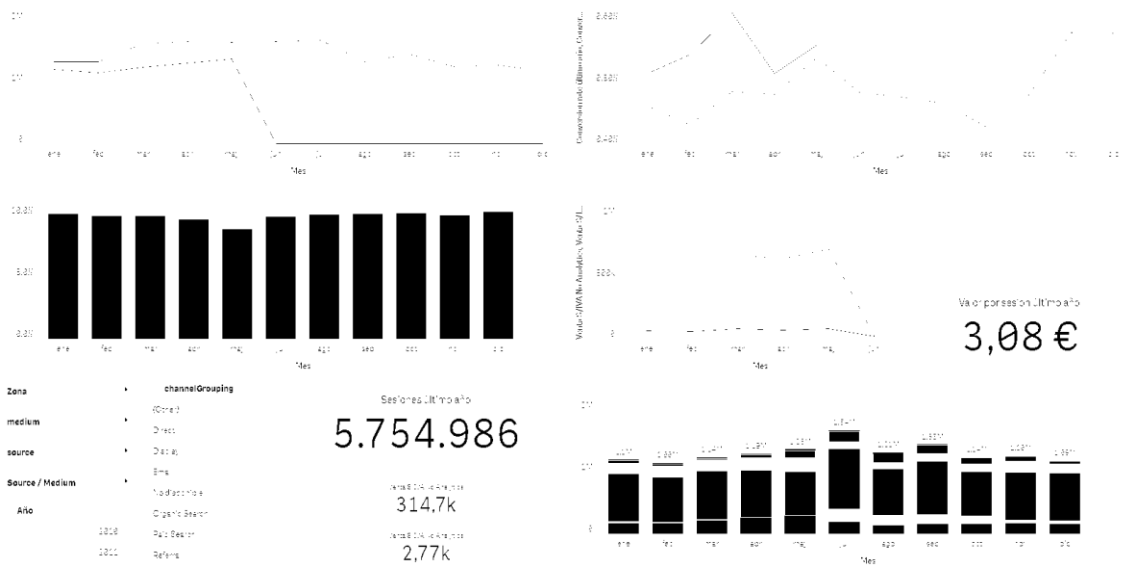


Imagen 54 - Dashboard Motobuyers – Analytics

Contiene información relativa a Google Analytics. Podemos observar los diferentes KPI's que miden la salud de una página. Además, podemos visualizar el seguimiento a doce meses vista para una idea general de la transición.

3.2.1.4.12. Analytics – Revisión Canal

Imagen 55 - Dashboard Motobuykers – Analytics Revisión Canal

Esta hoja recoge los diferentes KPI's facilitados por Google Analytics para conocer el estado de la página.

3.2.1.4.13. Analytics – Check Revenue

Año	Analytics Revenue	Verza S/CVA	Diferencia
Totales	8.708.817 €	10.917.476 €	-2.208.659 €
2010	0 €	53.344 €	-53.344 €
2011	0 €	283.312 €	-283.312 €
2012	0 €	334.341 €	-334.341 €
2013	0 €	2.193.718 €	-2.193.718 €
2014	369 €	4.837.323 €	-4.836.954 €
2015	7.687.410 €	8.192.494 €	-485.084 €
2016	1.021.407 €	2.820.227 €	-1.798.820 €

Fecha	NP Pedido	Zona	Es APP	Modo Pago	Verza S/CVA	Analytics Revenue	Verza S/CVA Analytics	Shipping Analytics	%_dificiaco
Totales					10.917.476 €	8.708.858,01 €	9.656.727,46 €	51.830,56 €	
30/06/2014	41333	ES	No APP	TFV	0,00 €	0,00 €	0,00 €	0,00 €	1
01/06/2014	21300	ES	No APP	Pa,Pa	0,00 €	0,00 €	0,00 €	0,00 €	1
11/03/2014	88870	ES	No APP	Pa,Pa	0,00 €	0,00 €	0,00 €	0,00 €	1
20/06/2014	106308	ES	No APP	988**00-20	0,41 €	0,00 €	0,00 €	0,00 €	1
27/04/2014	93824	IT	No APP	TFV	0,41 €	0,00 €	0,00 €	0,00 €	1
09/11/2015	96835	IT	No APP	TFV	0,42 €	0,21 €	0,21 €	0,00 €	1
20/06/2014	126030	ES	No APP	TFV	0,60 €	0,00 €	0,00 €	0,00 €	1
20/04/2016	062332	ES	APP	TFV	0,74 €	0,73 €	0,73 €	0,00 €	1
17/03/2015	228788	ES	No APP	TFV	0,83 €	0,00 €	0,00 €	0,00 €	1
01/11/2014	164879	FR	No APP	TFV	1,13 €	0,00 €	0,00 €	0,00 €	1
17/06/2014	135402	FR	No APP	TFV	1,37 €	0,00 €	0,00 €	0,00 €	1
11/07/2014	111914	FR	No APP	TFV	1,44 €	0,00 €	0,00 €	0,00 €	1

Imagen 56 - Dashboard Motobuykers – Analytics check revenue

Durante la realización del trabajo de final de grado se visualizó una incongruencia entre el importe de facturación facilitado por Google Analytics con el importe de facturación facilitado por la base de datos. Este hoja intenta ser un facilitador del check a realizar.

3.2.1.4.14. Adwords

Año	Mes	dia	Zona
2019	ene	ene	CO1
2019	feb	feb	EM1
2019	mar	mar	EM1
2019	abr	abr	EM1
2019	may	may	EM1
2019	jun	jun	EM1

adwords_cuenta -		Medidas												
adwords_campaña -														
Marca	Producto	Venta Producto S.N.A.	Acordos Cost	Venta Producto S.N.A.	Retorno Costo Comercio	Venta Producto S.N.A. Inversión	CPA Objetivo	Acordos CPA Res	Acordos CPA	Acordos CPA	Acordos CPA	Acordos CPA	Acordos CPA	Acordos CPA
Totales		480.390 €	484.815 €	122.660 €	-362.147	0,8	17,4 €	88,0 €	0,8%	2151183	441491893	4048	6912	
Motor, venta ESPAÑA		101.173 €	188.471 €	31.992 €	-156.479	0,8	19,2 €	138,2 €	0,3%	922804	187687601	1194	1874	
Motor, venta FRANCIA		83.062 €	110.993 €	18.282 €	-92.711	0,8	19,1 €	156,4 €	0,3%	881118	159181111	887	1149	
Motor, venta ITALIA		5227 €	81.169 €	31,6 €	-76.942	0,1	14,7 €	1.399,2 €	0,3%	282441	9712891,0	59	74	
Motor, venta venta ES		133.842 €	73.892 €	-11.972 €	-33.723	1,0	10,7 €	18,2 €	0,1%	884249	18986381	1874	2587	
Motor, venta venta FR		81.146 €	371.126 €	59.990 €	-11.928	0,4	17,5 €	36,6 €	0,1%	148771	711091,0	1011	1030	
Motor, venta venta IT		14.978 €	10.880 €	1.881 €	-6.025	1,3	10,7 €	61,8 €	1,8%	24288	5018027	184	289	

Imagen 57 - Dashboard Motobuykers – Adwords

Esta hoja recoge la información relativa a Google Adwords. Además, permite relacionar las ventas con los productos de las transacciones.

3.2.1.4.15. Tokens tarjetas

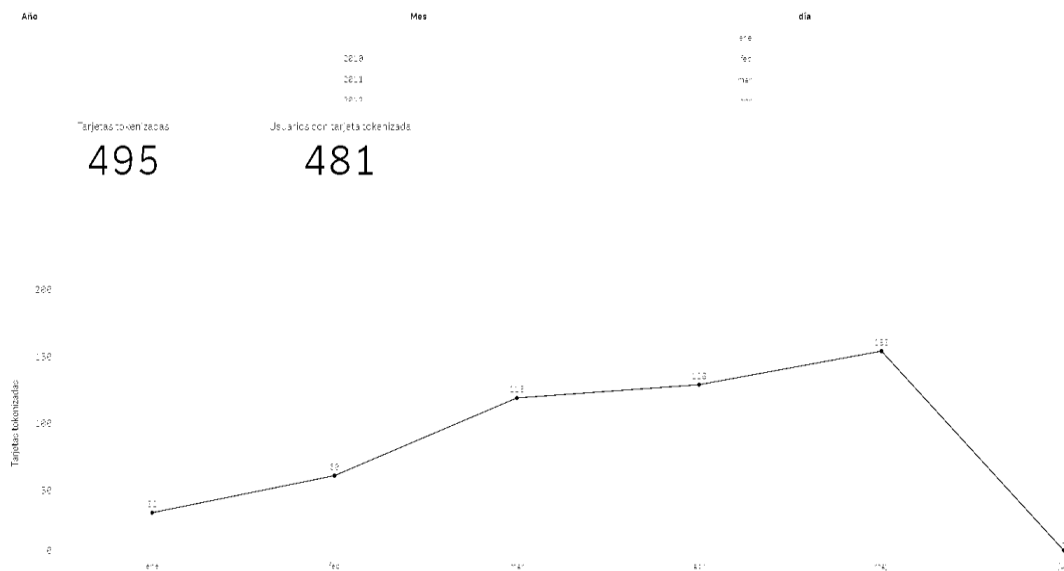


Imagen 58 - Dashboard Motobuykers – Tokens tarjetas

Los usuarios de la web pueden decidir guardar su tarjeta bancaria en el sistema de la web. Este registro se realiza sobre tarjetas tokenizadas. Esta hoja muestra el histórico de tarjetas que se han producido en el proceso.

3.2.1.4.16. Usuarios

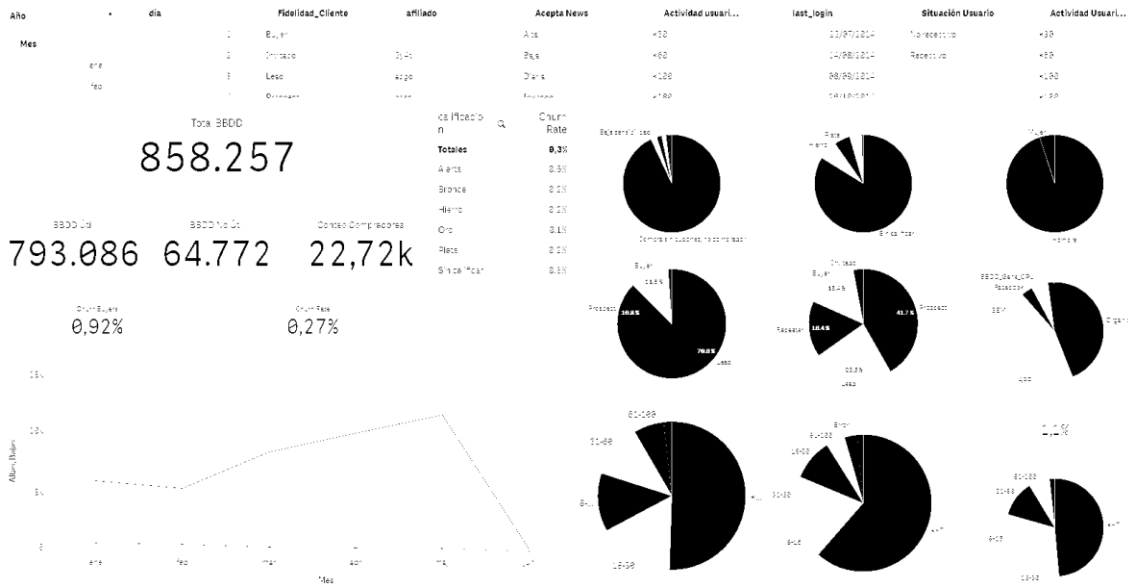


Imagen 59 - Dashboard Motobuykers – Usuarios

Hoja con indicadores relativos a la calidad de los usuarios para realizar un seguimiento de la vida útil de estos.

3.2.1.4.17. Promociones

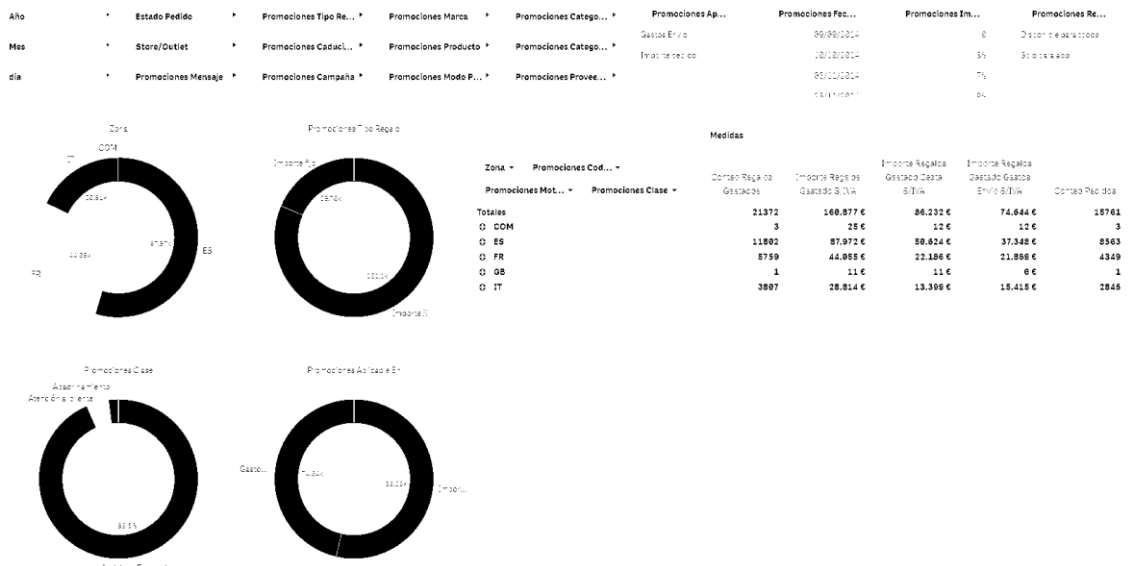


Imagen 60 - Dashboard Motobuykers – Promociones

Igual que otras e-commerce, la web permite el uso de regalos y cupones por parte de los usuarios. Dependiendo del origen de los cupones pueden ser

aplicados en combinación a otros o en solitario. Además, unos pueden ser efectivos en la cesta o en los gastos de envío. Esta hoja recurre a la visualización de los datos para poder hacer un seguimiento – provocación de la efectividad de las campañas de marketing.

3.2.1.4.18. Popup

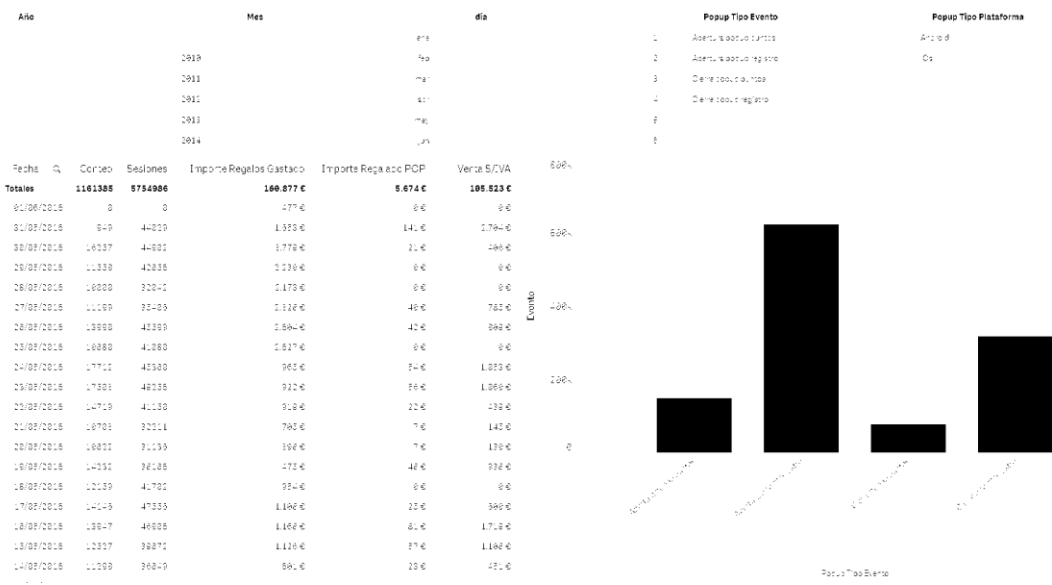


Imagen 61 - Dashboard Motobuykers – Popup

El departamento de marketing realiza una acción sobre los visitantes de la web que aún no están registrados. Este dashboard muestra la información de las acciones para hacer un seguimiento de la efectividad.

3.2.1.4.19. Leads – Buyers

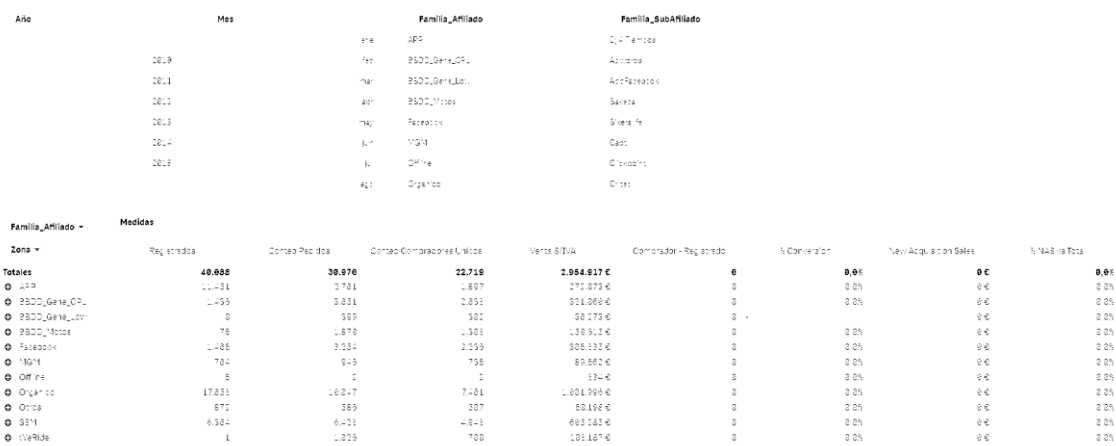


Imagen 62 - Dashboard Motobuykers – Leads Buyers

Dashboard de seguimiento de la recurrencia de los visitantes versus la cantidad de compradores.

3.2.1.4.20. Referral

Año	Mes	dia	afiliado	Familia_Afiliado	Retargeting_Deduplicate...	Fec. Creación				
		ene	1	APP	Definición	20/01/19				
	2019	feb	2	3822_Parents_CPL	Noticias	25/02/19				
	2019	mar	3	3822_Parents_CPL		29/03/19				
	2019	abr	4	3822_Mother		30/04/19				
De ay Ventas	Q	Corteo Pedidos	Venta S/DVA	Venta S/DVA Analytics	Margen % S/DVA	Margen S/DVA	Inversión	Reinversión Campaña	CPA	Venta/Inv
Totales		642	48.493,94 €	28.412,44 €	28,8%	13.173,8 €	4.211,59 €	8.962 €	7,8 €	11,7
Superior a L día		113	17.951,39 €	10.412,44 €	34,7%	8.894,4 €	2.438,84 €	5.653 €	7,8 €	8,6
Menor de L día		529	30.542,55 €	18.000,00 €	29,2%	4.279,4 €	1.772,75 €	3.309 €	7,8 €	9,1
De ay Ventas	Q	Corteo Pedidos	Venta S/DVA	Venta S/DVA Analytics	Margen % S/DVA	Margen S/DVA	Inversión	Reinversión Campaña	CPA	Venta/Inv
Totales		2394	242.431,57 €	128.717,00 €	28,7%	82.283,4 €	19.137,84 €	41.467 €	8,0 €	11,0
Menor de L día		1391	148.000,82 €	74.717,00 €	23,4%	42.891,2 €	12.714,32 €	21.965 €	8,0 €	8,1
Superior a L día		801	97.430,75 €	54.000,00 €	28,5%	39.392,2 €	6.423,52 €	19.502 €	8,0 €	11,6

Imagen 63 - Dashboard Motobuykers – Referral

Motobuykers realiza acciones conjuntamente con partners del sector. Esta hoja muestra la información facilitada por estos afiliados para así poder calcular ratios de beneficio/retorno y poder determinar cual de ellos es más conveniente para la empresa.

3.2.1.4.21. Distribución geográfica

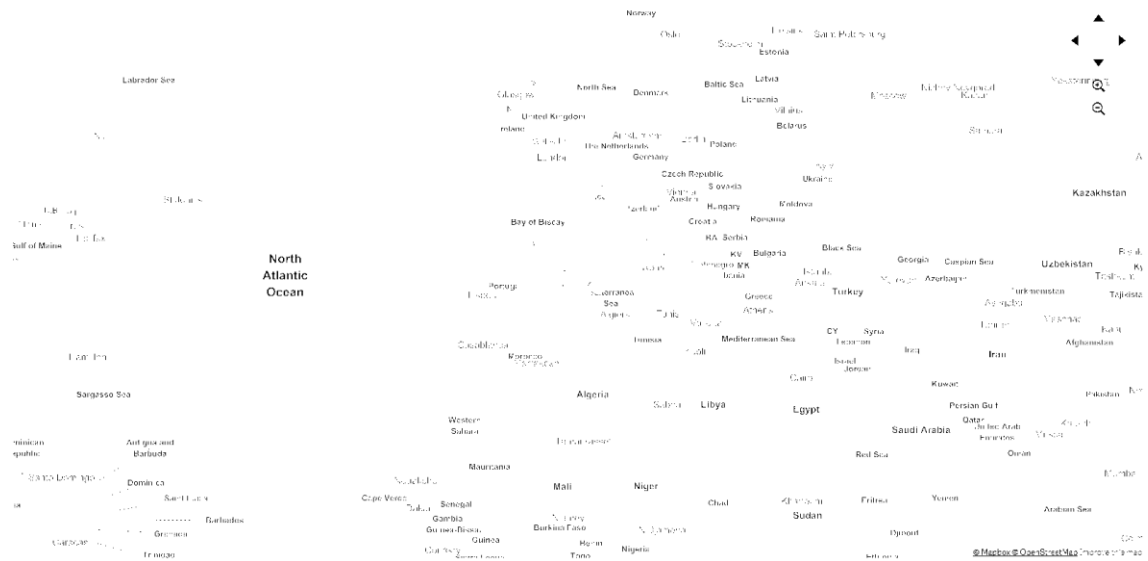


Imagen 64 - Dashboard Motobuykers – Distribución geográfica

Para el tratamiento de la información, a nivel efectivista, se realizó la construcción de un mapa de coordenadas según los códigos postales de los compradores de las zonas donde Motobuykers realiza el negocio. Acto seguido se creó la hoja de distribución geográfica para tener una mejor visión ocular de los lugares donde se realizan las compras.

3.2.1.5. Departamento finanzas

Con el fin de simplificar las tareas diarias al departamento de finanzas se crearon dos hojas con el objetivo de visualizar posibles incongruencias en la emisión de las facturas. Y, además, poder realizar un seguimiento al detalle de los abonos que se realiza a los clientes.

3.2.1.5.1. Check facturas

Año	Mes	Zona	Modo Pago
2021	ene	0011	Pagado
2021	feb	0101	Pagado
2021	mar	error	Reembolso
2021	abr	00	no

PK_Pedido	Q	Venta S/IVA	Importe Factura	Diferencia
Totales		2.854.917,451 €	2.855.184,887 €	-419,692 €
862-50		203.428 €	1.85.978 €	38.918 €
939871		51.164 €	18.978 €	11.814 €
862-50		18.199 €	20.179 €	3.792 €
911500		51.583 €	50.359 €	1.133 €
948593		51.583 €	50.359 €	1.133 €
948189		51.583 €	50.359 €	1.133 €
952759		51.583 €	50.359 €	1.133 €
969014		51.583 €	50.359 €	1.133 €
948585		51.172 €	50.128 €	0.912 €
948029		51.172 €	50.128 €	0.912 €

Imagen 65 - Dashboard Motobuyers – Check Facturas

Se llevó a cabo una hoja específica para finanzas para la comprobación que los importes de ventas coincidan con los importes visualizados en las facturas.

3.2.1.5.2. Abonos

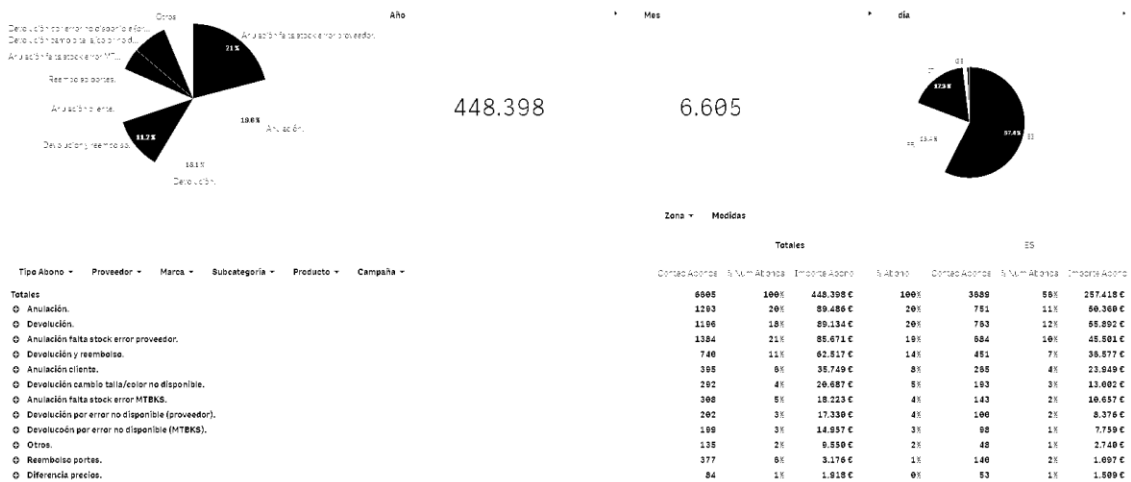


Imagen 66 - Dashboard Motobuyers – Abonos

Esta hoja también es perteneciente al departamento de finanzas. Se busca lograr tener una visión clara de los motivos de los abonos para así conseguir avanzarnos a estos y reducirlos.

3.2.2. WeRide

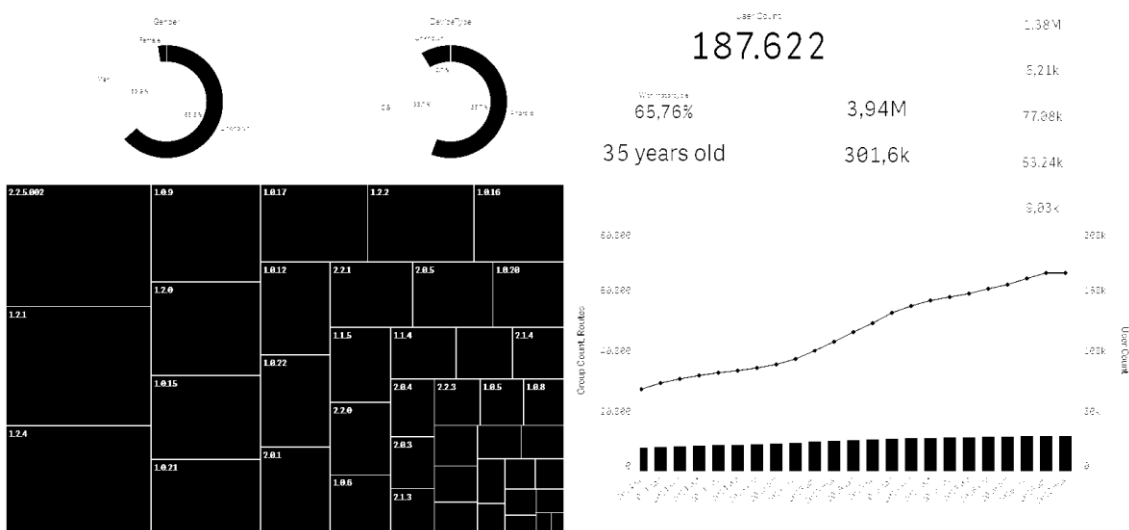
La solución propuesta para WeRide ha sido una implementación esencial y consistente sobre los datos provenientes de la base de datos y complementados con la información de Google Analytics. Este hecho fue comentado anteriormente, no existía ningún tipo de solución de reporting sobre WeRide de ahí que se planteara una visualización sencilla sobre los datos. El resultado final de la solución es tal que así:



Imagen 67 - App WeRide

Tal como se puede ver en la imagen la solución de implantación ha consistido en la creación de cinco hojas. Cada una de ellas ha sido creada sobre con una finalidad preconcebida. Si vamos al detalle:

3.2.2.1. Dashboard



Esta hoja es la más completa de las cinco. En concreto hace referencia a todo un conjunto de indicadores básicos sobre la salud de la aplicación. En otras palabras, nos segmenta a los usuarios según sus características.

Tal como podemos observar, en esta hoja toma importancia la distribución de usuarios según la versión de la app de WeRide que poseen. También podemos ver como se categoriza a los usuarios según el tipo de dispositivo y el género. Además de multitud de indicadores como la edad media, etc, podemos observar (abajo a la derecha) un gráfico de barras que muestra la cantidad de usuarios, grupos y rutas sobre una línea de tiempo.

3.2.2.2. Users

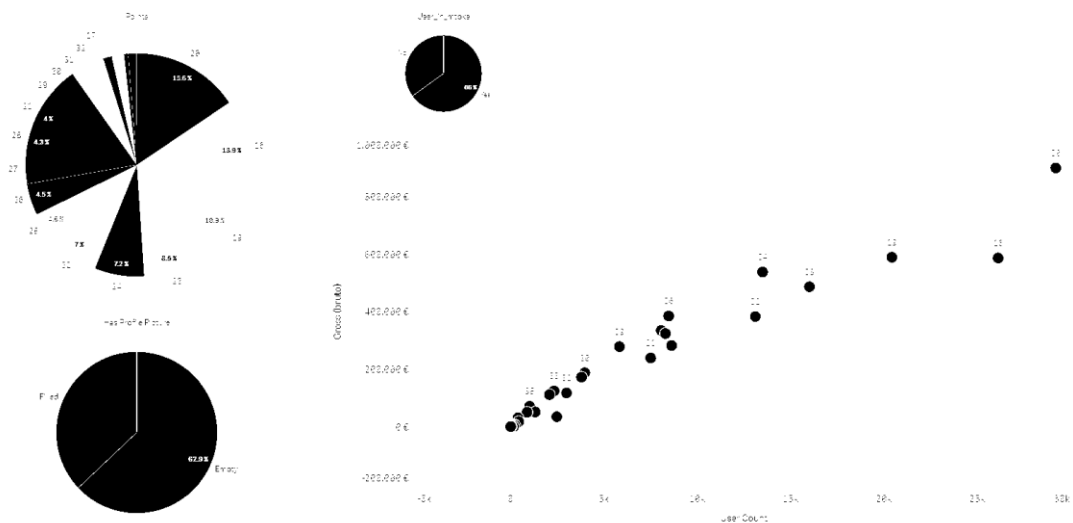


Imagen 69 - WeRide App - Users

La hoja de usuarios ha sido construida para añadir información complementaria a la anteriormente descrita. Por una parte muestra la cantidad de usuarios que tiene la foto de perfil introducida. Por otra parte, también muestra la cantidad de usuarios coincidentes en la base de datos de Motobuykers (cross-selling). Además podemos ver como aquí entra en juego la calificación de usuarios que habíamos creado anteriormente. Vemos su segmentación de usuarios según la cantidad de usuarios en cada puntuación. Y, el dato más importante, de cómo

la calificación puede llegar a ser un indicativo de ser mejor comprador en Motobuykers (gráfico de dispersión).

3.1.2.3. Geo Groups

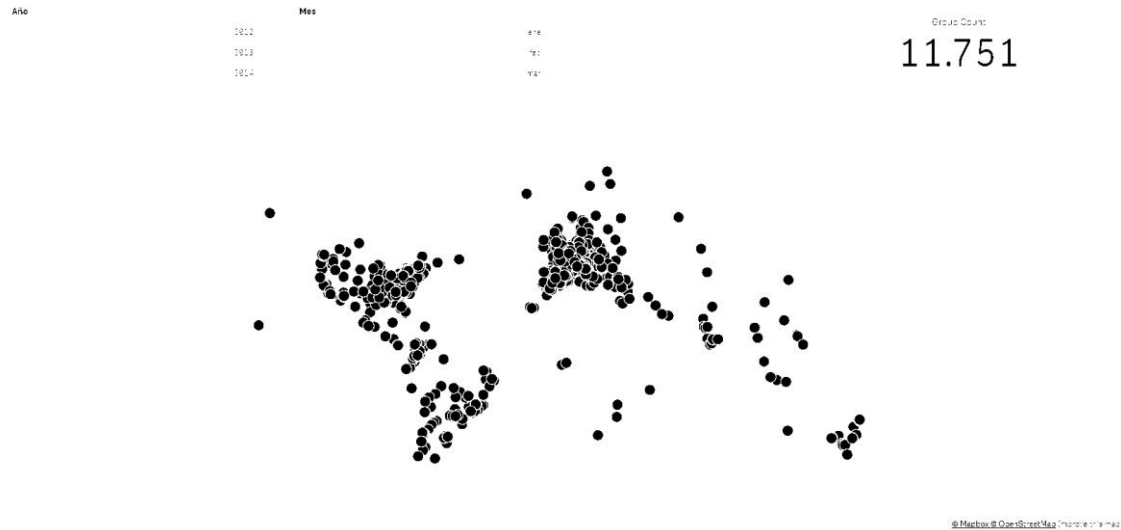


Imagen 70 - WeRide App – Groups

Hoja que muestra localización en el mapa de los diferentes grupos que componen la naturaleza de WeRide. A destacar los filtros, nos permiten ver para una fecha determinada los grupos que se crearon.

3.1.2.4. Geo Users

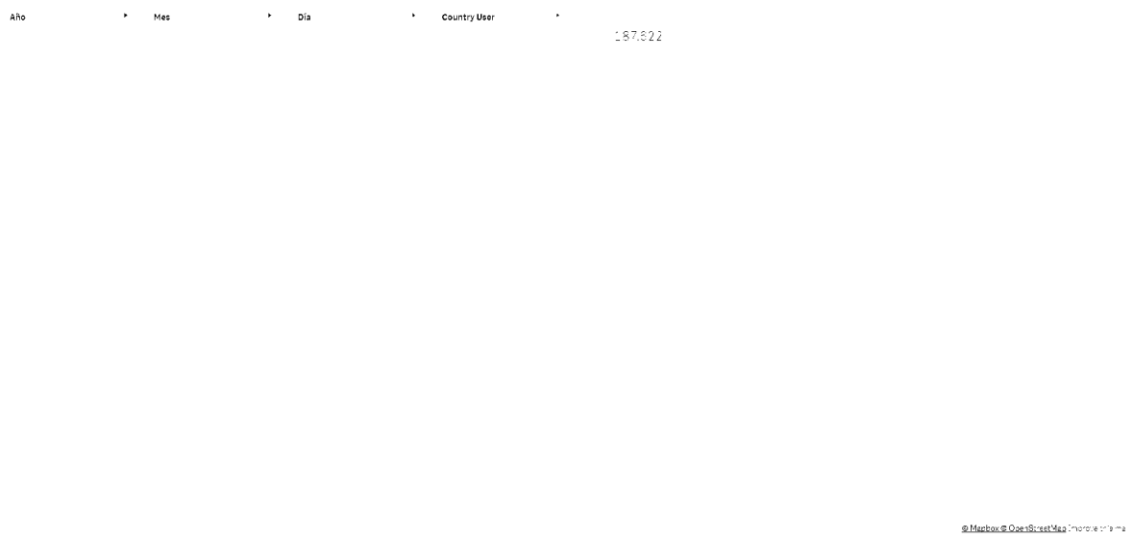


Imagen 71 - WeRide App – Geo Users

Hoja de dashboard que, igual que con grupos, representa la ubicación de los diferentes usuarios de la aplicación. Tal como podemos ver, la hoja nos llega a permitir filtrar por recencia del usuario y poder observar, por ejemplo, la ubicación de los usuarios creados en una fecha determinada.

3.2. Revisión de los dashboards

Una vez se dio por concluida la fase de desarrollo de la solución dio a lugar la fase de revisión del trabajo hecho. Para llevar a cabo esta tarea se dividió el objetivo en partes: una tarea por departamento. En otras palabras, empezamos reuniones con los jefes de departamento para realizar a revisar las hojas y aplicar las correcciones oportunas.

Las revisiones dieron a lugar a diferentes modificaciones que se apuntaban en un ticket de redmine, que pasaba a ser valorado y priorizado. Una vez hechas las modificaciones se volvía a la fase de revisión para cerrar la tarea, o en caso contrario volver a realizar un registro de los cambios a realizar.

Es importante remarcar que estas sesiones se realizaban durante largas horas durante el día y, habitualmente, las tareas de modificación se realizaban al momento. Por ejemplo, era habitual tener que cambiar el nombre del KPI al que prefiriese el stakeholder.

Una parte importante de las modificaciones vino dada según pequeñas modificaciones en las tablas dinámicas / pivotantes. Estas pueden ser ordenadas por el mismo usuario según la visión que requiera en ese momento. A nivel personal consideré que el usuario sería el que preferiría disponer de una única hoja e cambiar a petición la forma de ordenación. Esta forma de proceder no fue del agrado de todo el mundo. Tanto es así, que como se observa en las hojas 'CMg', los datos son los mismos pero en orden modificado en diferentes hojas. Este hecho viene motivado a que en Qlik Sense Enterprise Edition, al ser una versión basada en navegador web, se pueden tener diferentes hojas abiertas con el objetivo de tener datos complementarios a la vista.

Por norma general la revisión de los dashboards por cada departamento acabó de forma rápida en un primer sprint de modificación. En contraposición, el departamento de Marketing, como es normal al disponer tantas hojas, tuvimos que realizar diferentes sprints para que todo quedara a gusto del consumidor.

4. Conclusiones

4.1. Lecciones aprendidas

Ahora que el proyecto de final de grado se puede dar por concluido podemos decir que se han cumplido todos los objetivos inicialmente planteados en cada una de sus fases. Los conocimientos adquiridos en las asignaturas estudiadas, sobretodo en los últimos años de carrera, en combinación a la experiencia profesional me han permitido disfrutar de la implantación de las soluciones propuestas siendo, como no, un reto profesional determinando en mi carrera laboral.

Durante la duración del proyecto he podido comprobar como muchas de las asignaturas impartidas en la carrera han sido de gran importancia para la consecución de objetivos. Cabe destacar como la gestión del proyecto, muchas veces ninguneada, es una pieza clave desde el inicio al fin. Juntamente con la gestión del proyecto saber marcar correctamente los objetivos y requisitos también pasa a ser un factor determinante. Tanto es así que, durante la implantación, se han detectado ciertos requisitos poco o mal definidos que luego se han tenido que corregir.

Por supuesto asignaturas más técnicas como base de datos y fundamentos de sistemas de la información han sido muy importantes. Estamos hablando sobre la gestión de base de datos de millones de tuplas que tienen que ser tratadas usando técnicas de eficiencia para mejorar el rendimiento de las consultas.

En resumen, tanto a nivel educativo como laboral el proyecto me ha servido para prestar atención en determinadas maneras de hacer las cosas que deben

de ser modificadas pero, al mismo tiempo, otras que son fundamentales para una correcta finalización del proyecto.

4.2. Reflexión sobre el cumplimiento de objetivos

Los objetivos de la adquisición de una solución de business intelligence han girado en torno a disponer de una visión de 360º de los datos de la empresa. En otras palabras, traspasar la información aislada en varios informes a uno de solo que permita a todos los departamentos tener el mismo dato de verdad. Además, el otro objetivo paralelo, se marcó en la no generación de multitud de informes. Aunque de manera automática, la generación de informes no comporta un gasto de tiempo, si que necesita de tiempo de máquina para generarlos.

Los objetivos temporales, fases del proyecto, se definieron siguiendo las entregas de las PACS para que así relativamente más sencillo realizar un corte en el tiempo para valorar el trabajo hecho y el que vendría. Al ser un proyecto real sobre la empresa donde ya se trabaja permitió tener gran conocimiento de la forma de trabajar y poder marcar los intervalos de tiempo según las necesidades, o costumbres, particulares de Motobuykers.

A nivel personal estoy muy contento con los resultados obtenidos. No disponía de conocimientos de ninguna solución de business intelligence y, por tanto, durante la primera fase y antes implicó la formación en la plataforma. De manera subjetiva estoy totalmente seguro que la solución propuesta cumple con las necesidades de la empresa y, sobretodo, está dimensionada a ella. Por ejemplo, la adquisición de más usuarios hubiese sido un sobrecoste innecesario.

Durante la tercera fase, revisión y presentación, se realizaron varias encuestas a los diferentes stakeholders para conocer el grado de satisfacción y confianza en la plataforma. En cada una de ellas se valoró muy positivamente la

plataforma. En concreto, la obtención de resultados completos que pueden ser relacionados con otros datos de usuarios ha sido el beneficio mejor valorado. A nivel de la duración del proyecto todos los departamentos han considerado que ha sido satisfactorio y, cuando han dispuesto de ello, con una solución completa.

En resumen, las dos partes del proyecto hemos considerado que la consecución de objetivos ha sido la correcta y han sido temporalizados siguiendo los ritmos naturales de Motobuykers.

4.3. Seguimiento de la planificación y metodología

El proyecto ha sido desarrollado en una empresa de reciente creación que compite con otras muy similares, todas ellas en una enorme búsqueda de crecimiento rápido y descontrolado a golpe de inversión externa. Este hecho implica dos situaciones que han marcado el desarrollo en toda la vida del proyecto. Por una parte la empresa requiere que los cambios se produzcan de la manera más rápida posible para así ser los primeros en disponer de la ventaja competitiva. Este hecho ha comportado la dedicación de muchas horas en la fase 2, implantación, debido a querer una solución completa lo antes posible.

Por otro lado, tal como comentábamos, Motobuykers ha requerido una implantación vista a continuas modificaciones. No podemos decir que en la fase 3 se consiguiese una primera entrega satisfactoria. La realidad ha sido la entrega de una propuesta que ha sufrido continuos cambios hasta llegar a la versión que satisfactoria para el usuario. Se añade el hecho, en positivo, que la plataforma Qlik Sense permite la edición de las hojas por parte del usuario. Esta situación ha facilitado que se completasen con más rapidez las soluciones.

Cabe destacar, fuera de cualquier planificación, que durante el tiempo de implantación se han producido situaciones conflictivas en cuanto a la estabilidad de la plataforma. Durante la publicación de una app se produjo un solapamiento con el refresco de los datos. Esta situación dio a cabo una

corrupción de la aplicación que estaba corrompida. Pese a que se realizaban exportaciones completas de la app el sistema no realizaba exportaciones de las hojas privadas de los usuarios dando a lugar una pérdida de información. Procedí a ponerme en contacto con el soporte de la herramienta no consiguiendo una solución satisfactoria. Desde este instante se contrato una solución de backups completos sobre la vps por si volvía a suceder.

4.4. Líneas de trabajo futuro

Debido a la corta duración del presente trabajo, aún habiendo cumplidos los objetivos marcados, se han localizado ciertas tareas que hará falta que la empresa se focalice para conseguir una ventaja competitiva respecto a sus competidores.

Por una parte, durante el desarrollo de los diferentes dashboards hemos podido notar un descenso en el rendimiento de la plataforma. Las virtudes del software contratado se basan en la cantidad de datos almacenados en memoria. Por tanto, como es de suponer, a más datos peor rendimiento. A pesar de no contratar el hardware sobre los requisitos mínimos creemos que el siguiente paso a desarrollar es un cambio de servidor el cual posea más memoria ram.

Por otra parte, acto seguido del paso anteriormente descrito, haría falta que la empresa apostase por la adquisición de datos en masa. Estamos hablando de la adquisición de una solución de big data que permita a la empresa procesar información de los usuarios que hasta ahora no es posible. La decisión debería ser complementada con la contratación de un conector entre la base de datos elegida y Qlik Sense, por ejemplo Jethro.

En el mismo sentido, debido a la alta cantidad de datos procesados, sería de gran utilidad la contratación de una solución profesional de base de datos como Oracle. Este cambio aportaría un plus de rendimiento que estoy seguro será apreciado debido a la rapidez de las consultas. También se debería plantear el uso real de una arquitectura de data warehouse donde los datos sean agregados y no solo actualizados. De esta manera se podría generar imágenes de estado de los datos en una fecha determinada.

Como hemos visto, la primera de ellas es una propuesta que la empresa debería poner solución en las próximas semanas. En cambio las otras deberían formar parte de la estrategia de empresa para los próximos años.

5. Glosario

Business Intelligence (BI)

Conjunto de herramientas cuyo objetivo es la toma de decisiones en el conjunto de las organizaciones basándose en el conocimiento y la gestión de los datos.

Data warehouse

Es un repositorio de información en la cual se integran las diferentes fuentes de datos para acabar siendo usado por las herramientas de BI.

ETL

Son las siglas de 'Extracción, Transformación y Carga(load)' que definen el conjunto de procesos que realizan las tareas de integración en un data warehouse.

Dashboard

Es una hoja que representa los datos, también conocido como cuadro de mandos.

Key Performance Indicator (KPI)

Medida creada con el objetivo de poder valorar una determinada acción, su grado de satisfacción.

Enterprise Resource Planning (ERP)

Sistema de la empresa que recoge la información relativa al día a día. Puede gestionar de uno a todos los departamentos de la empresa.

Google Analytics

Plataforma de Google que facilita información analítica de la navegación de los usuarios.

Virtual Private Server (VPS)

Tal como su nombre indica es un servidor virtual que forma parte de una máquina más potente donde se ubican diversas VPS.

Google Adwords

Plataforma de Google que gestiona campañas de pago.

Q&A (Questions & Answers)

Etapa en que se inicia un periodo de comprobación de los datos y una fase de preguntas y respuestas por parte de los implicados.

6. Bibliografía

- Qlik Recursos <http://www.qlik.com>
- Pentaho Data Integration <http://community.pentaho.com>
- Adquisición de conocimiento <https://www.udemy.com>
- Google Analytics API support
<https://developers.google.com/analytics/?hl=es>
- Google Analytics cursos
https://www.google.es/intl/es_ALL/analytics/learn/index.html
- Requerimientos Qlik Sense Enterprise <https://help.qlik.com/en-US/sense/2.2/Subsystems/Installation/Content/Introduction/system-requirements.htm>
- Instalación Qlik Sense Enterprise <https://help.qlik.com/en-US/sense/2.2/Subsystems/Installation/Content/InstallationLicensing/Installing.htm>
- Metodología Scrum
<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

7. Anexos

Anexo 1 – QVD de Qlik Sense sobre Motobuykers

```
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 MonthNames='ene;feb;mar;abr;may;jun;jul;ago;sep;oct;nov;dic';
SET DayNames='lun;mar;mié;jue;vie;sáb;dom';
SET
LongMonthNames='enero;febrero;marzo;abril;mayo;junio;julio;agosto;septiemb
e;octubre;noviembre;diciembre';
SET LongDayNames='lunes;martes;miércoles;jueves;viernes;sábado;domingo';
SET FirstWeekDay=0;
SET BrokenWeeks=1;
SET ReferenceDay=0;
SET FirstMonthOfYear=1;

//Para desarrollo poner a 0
Let vGeneraQVD=1;

Let vRutaQVD='lib://DatosLocal (vps229099_administrator)';

LIB CONNECT TO 'Produccion (vps229099_administrator)';

if $(vGeneraQVD)=1 then

Latitudes:
```



```

LOAD *;
SQL SELECT *
FROM BI.`Coordenadas_Provincias`;
Store Latitudes into '$(vRutaQVD)Dashboard\Latitudes.qvd' (qvd);
drop table Latitudes;

```

usuarios:

```

LOAD *;
SQL SELECT uu.*, if(last_login=0,null,FROM_UNIXTIME(last_login)) AS
last_login_date,
Familia_Afiliado_Nueva(afiliado) AS Familia_Afiliado,
Familia_Afiliado_SubAfiliado(afiliado) AS Familia_SubAfiliado,
case
    when ifnull(pais,idioma) in ('es','pt') then 'es'
    when ifnull(pais,idioma) in ('it') then 'it'
    when ifnull(pais,idioma) in ('fr','be') then 'fr'
    else 'en'
end as lang,
ifnull((select 1 from pedidos p where p.id_usuario=uu.id and p.estado in (1,2)
group by 1),0) as comprador,
ifnull((select 1 from usuarios_moto um where uu.id=um.id_usuario group by
1),0) as tiene_moto
FROM `motobuykers_produccion`.usuarios uu;
Store usuarios into '$(vRutaQVD)Dashboard\usuarios.qvd' (qvd);
drop table usuarios;

```

Objetivos:

```

LOAD *;
SQL SELECT *
FROM BI.`BI_Envio_Datos_Empresa`;
Store Objetivos into '$(vRutaQVD)Dashboard\Objetivos.qvd' (qvd);
drop table Objetivos;

```

regalos_clases:

LOAD *;

SQL SELECT *

FROM `motobuykers_produccion`.`regalos_clases`;

Store regalos_clases into '\$(vRutaQVD)Dashboard\regalos_clases.qvd' (qvd);

drop table regalos_clases;

regalos_motivos:

LOAD *;

SQL SELECT *

FROM `motobuykers_produccion`.`regalos_motivos`;

Store regalos_motivos into '\$(vRutaQVD)Dashboard\regalos_motivos.qvd'

(qvd);

drop table regalos_motivos;

pedidos_regalos:

LOAD *;

SQL SELECT *

FROM `motobuykers_produccion`.`pedidos_regalos`;

Store pedidos_regalos into '\$(vRutaQVD)Dashboard\pedidos_regalos.qvd'

(qvd);

drop table pedidos_regalos;

actividades_promocionales:

LOAD *;

SQL SELECT *

FROM `motobuykers_produccion`.`actividades_promocionales`;

```
Store actividades_promocionales into
'$(vRutaQVD)Dashboard\actividades_promocionales.qvd' (qvd);
drop table actividades_promocionales;
```

regalos:

```
LOAD *;
SQL SELECT *
FROM `motobuykers_produccion`.regalos;
```

```
Store regalos into '$(vRutaQVD)Dashboard\regalos.qvd' (qvd);
drop table regalos;
```

Mailing:

```
LOAD *;
SQL SELECT id_campana, id_mailing, if(name_mailing like '%(%-%-%)%',
concat(date_format(send_date_mailing,'%Y%m%d_'),trim(replace(name_mailing,substring_index(name_mailing,' ',-1),''))),
name_mailing) as name_mailing, send_date_mailing, weblink, total_sent,
total_accepted, hardbounces, softbounces, unsubscribers, suspends,
spamcomplaints, unique_opens, total_renders, unique_renders,
total_clickthroughs, unique_clickthroughs, unique_link_clickthroughs,
total_conversion, unique_conversion, click2open_rate, revenue, pedidos,
case
when gr.name_mailing like '%\st\_%' then 'Store'
when gr.name_mailing like 'st\_%' then 'Store'
when gr.name_mailing like '%\ou\_%' then 'Outlet'
when gr.name_mailing like 'ou\_%' then 'Outlet'
when gr.name_mailing like '%\hb\_%' then 'Híbrida'
when gr.name_mailing like 'hb\_%' then 'Híbrida'
when gr.name_mailing like '%\sg\_%' then 'Segmentada'
when gr.name_mailing like 'sg\_%' then 'Segmentada'
end as 'Outlet/Store',
```

```

case
  when gr.name_mailing like '%\_top10\_%' then 'Top'
  when gr.name_mailing like '%\_last\_%' then 'Last'
  when gr.name_mailing like '%\_modular\_%' then 'Modular'
    when gr.name_mailing like '%\_mod\_%' then 'Modular'
  when gr.name_mailing like '%\_daily\_%' then 'Daily'
  when gr.name_mailing like '%\_tecnica\_%' then 'Tecnica'
  when gr.name_mailing like '%\_tec\_%' then 'Tecnica'
  when gr.name_mailing like '%\_dedicada\_%' then 'Dedicada'
  when gr.name_mailing like '%\_ded\_%' then 'Dedicada'
  else 'Otro'
end as 'eMailType',

```

```

case
  when gr.name_mailing like '%\_es' then 'ES'
  when gr.name_mailing like '%\_es %' then 'ES'
  when gr.name_mailing like '%\_es\_%' then 'ES'
  when gr.name_mailing like '%\_it' then 'IT'
  when gr.name_mailing like '%\_it %' then 'IT'
  when gr.name_mailing like '%\_it\_%' then 'IT'
  when gr.name_mailing like '%\_fr' then 'FR'
  when gr.name_mailing like '%\_fr %' then 'FR'
  when gr.name_mailing like '%\_fr\_%' then 'FR'
    when gr.name_mailing like '%\_uk' then 'GB'
  when gr.name_mailing like '%\_uk %' then 'GB'
  when gr.name_mailing like '%\_uk\_%' then 'UK'
  when gr.name_mailing like '%\_int' then 'COM'
  when gr.name_mailing like '%\_int %' then 'COM'
  when gr.name_mailing like '%\_int\_%' then 'COM'
  else 'error'
end as 'Pais'

```

```

FROM Webpower.`listado_mailings_summary` gr;
Store Mailing into '$(vRutaQVD)Dashboard\Mailing.qvd' (qvd);
drop table Mailing;

```

BI_GoogleAnalytics_RowData:

LOAD *;

SQL SELECT *

FROM `motobuyers_produccion`.`BI_GoogleAnalytics_RowData`;

Store BI_GoogleAnalytics_RowData into

'\$(vRutaQVD)Dashboard\BI_GoogleAnalytics_RowData.qvd' (qvd);

drop table BI_GoogleAnalytics_RowData;

FechaExpedicion:

LOAD

ID_PEDIDO_LINEA,

date(FECHA_EXPEDICION) AS FECHA_EXPEDICION;

SQL

SELECT epl.id_pedido_linea AS 'ID_PEDIDO_LINEA',

ex.id AS 'ID_EXPEDICION',

ex.referencia_motobuyers AS 'REF_EXPEDICION',

DATE(MAX(ex.fecha_expedicion)) AS 'FECHA_EXPEDICION'

FROM expediciones_pedidos_lineas epl JOIN expediciones ex ON

(epl.id_expedicion = ex.id)

WHERE ex.tipo = 1

AND ex.estado IN (1, 2, 3, 4)

AND ex.is_reexpedicion = 0

GROUP BY epl.id_pedido_linea;

Store FechaExpedicion into '\$(vRutaQVD)Dashboard\FechaExpedicion.qvd'

(qvd);

drop table FechaExpedicion;

FechaEntrega:

LOAD ID_PEDIDO_LINEA,

date(FECHA_ENTREGA) AS FECHA_ENTREGA;

SQL SELECT epl.id_pedido_linea AS 'ID_PEDIDO_LINEA',

```

ex.id AS 'ID_EXPEDICION',
ex.referencia_motobuykers AS 'REF_EXPEDICION',
DATE(MAX(ex.fecha_actualizacion)) AS 'FECHA_ENTREGA'
FROM expediciones_pedidos_lineas epl JOIN expediciones ex ON
(epl.id_expedicion = ex.id)
WHERE ex.tipo = 1
AND ex.estado = 3
AND ex.is_reexpedicion = 0
GROUP BY epl.id_pedido_linea;

```

```

Store FechaEntrega into '$(vRutaQVD)Dashboard\FechaEntrega.qvd' (qvd);
drop table FechaEntrega;

```

categorias:

```

LOAD id_producto,
Categoria,
Subcategoria;
SQL
SELECT
cp.id_producto,
cz1.nombre AS Categoria,
cz2.nombre AS Subcategoria
from (SELECT catp.id_categoria, catp.id_producto
FROM categorias_productos catp
GROUP BY catp.id_producto) cp
JOIN categorias ca ON (cp.id_categoria = ca.id)
LEFT JOIN categorias_zonas cz1 ON (ca.id_padre = cz1.id_categoria AND
cz1.id_zona = 1)
LEFT JOIN categorias_zonas cz2 ON (ca.id = cz2.id_categoria AND
cz2.id_zona = 1);

```

```

Store categorias into '$(vRutaQVD)Dashboard\categorias.qvd' (qvd);

```

drop table categorias;

marcas:

LOAD *;

SELECT *

FROM `motobuykers_produccion`.marcas;

Store marcas into '\$(vRutaQVD)Dashboard\marcas.qvd' (qvd);

drop table marcas;

proveedores:

LOAD

*;

SQL SELECT pro.*, au.username as admin_users_username

FROM `motobuykers_produccion`.proveedores pro

left join admin_users au on pro.id_responsable=au.id;

Store proveedores into '\$(vRutaQVD)Dashboard\proveedores.qvd' (qvd);

drop table proveedores;

productos_zonas:

LOAD *;

SQL SELECT

pz. *,

case

when locate('OAho4uWcm5k',pz.descripcion)

or

locate('OAho4uWcm5k',pz.video)

or
locate('tdzwyehRTxE',pz.descripcion)
or
locate('tdzwyehRTxE',pz.video)
or
locate('eNew5psRfVk',pz.descripcion)
or
locate('eNew5psRfVk',pz.video)
or
locate('xDwK2TxzI8',pz.descripcion)
or
locate('xDwK2TxzI8',pz.video)
or
locate('VxEOBMn4LQ4',pz.descripcion)
or
locate('VxEOBMn4LQ4',pz.video)
or
locate('pUwdLQ95yww',pz.descripcion)
or
locate('pUwdLQ95yww',pz.video)
or
locate('EmesBc2_3MU',pz.descripcion)
or
locate('EmesBc2_3MU',pz.video)
or
locate('Seoh0f5G3CQ',pz.descripcion)
or
locate('Seoh0f5G3CQ',pz.video)
or
locate('UwGSNtBABTc',pz.video)
or
locate('UwGSNtBABTc',pz.descripcion)
or
locate('sQiScI5QKIo',pz.video)


```

or
locate('sQiScI5QKIo',pz.descripcion)
or
locate('NLj0VzEy120',pz.video)
or
locate('NLj0VzEy120',pz.descripcion)
or
locate('m45RthfqEJ4',pz.video)
or
locate('m45RthfqEJ4',pz.descripcion)
or
locate('8gEo6n3WTBI',pz.video)
or
locate('8gEo6n3WTBI',pz.descripcion)
or
locate('GJoPdjGw2Ac',pz.video)
or
locate('GJoPdjGw2Ac',pz.descripcion)
or
locate('ncyYqF-dCzc',pz.video)
or
locate('ncyYqF-dCzc',pz.descripcion)
or
locate('CqnD8k4dCSk',pz.video)
or
locate('CqnD8k4dCSk',pz.descripcion)
then 'Video propio'
when locate('youtube',pz.descripcion)
or
locate('youtube',pz.video) then 'Video externo'
else 'Sin video'
end as Tiene_video
FROM `motobuykers_produccion`.`productos_zonas` pz;

```

```
Store productos_zonas into '$(vRutaQVD)Dashboard\productos_zonas.qvd'  
(qvd);  
drop table productos_zonas;
```

```
campanias:  
SQL SELECT *  
FROM `motobuykers_produccion`.campanias;
```

```
Store campanias into '$(vRutaQVD)Dashboard\campanias.qvd' (qvd);  
drop table campanias;
```

```
productos:  
SQL SELECT *  
FROM `motobuykers_produccion`.productos;
```

```
Store productos into '$(vRutaQVD)Dashboard\productos.qvd' (qvd);  
drop table productos;
```

```
atributos_valores:  
SQL SELECT id,  
    valor  
FROM `motobuykers_produccion`.`atributos_valores`;
```

```
Store atributos_valores into '$(vRutaQVD)Dashboard\atributos_valores.qvd'  
(qvd);  
drop table atributos_valores;
```

```
ReferenciaExpedicion:  
LOAD DISTINCT  
ID_PEDIDO_LINEA,
```

```

REF_EXPEDICION;
SQL
SELECT epl.id_pedido_linea AS 'ID_PEDIDO_LINEA',
ex.id AS 'ID_EXPEDICION',
ex.referencia_motobuykers AS 'REF_EXPEDICION',
DATE(MAX(ex.fecha_expedicion)) AS 'FECHA_EXPEDICION'
FROM expediciones_pedidos_lineas epl JOIN expediciones ex ON
(epl.id_expedicion = ex.id)
WHERE ex.tipo = 1
AND ex.estado IN (1, 2, 3, 4)
AND ex.is_reexpedicion = 0
GROUP BY epl.id_pedido_linea;

```

```

Store ReferenciaExpedicion into
'$(vRutaQVD)Dashboard\ReferenciaExpedicion.qvd' (qvd);
drop table ReferenciaExpedicion;

```

```

IDExpedicion:
LOAD DISTINCT
ID_PEDIDO_LINEA,
ID_EXPEDICION;
SQL
SELECT epl.id_pedido_linea AS 'ID_PEDIDO_LINEA',
ex.id AS 'ID_EXPEDICION',
ex.referencia_motobuykers AS 'REF_EXPEDICION',
DATE(MAX(ex.fecha_expedicion)) AS 'FECHA_EXPEDICION'
FROM expediciones_pedidos_lineas epl JOIN expediciones ex ON
(epl.id_expedicion = ex.id)
WHERE ex.tipo = 1
AND ex.estado IN (1, 2, 3, 4)
AND ex.is_reexpedicion = 0
GROUP BY epl.id_pedido_linea
;

```

```
Store IDEXpedicion into '$(vRutaQVD)Dashboard\IDExpedicion.qvd' (qvd);
drop table IDEXpedicion;
```

Pedidos:

```
LOAD *;
```

```
SQL SELECT id, id_usuario, nombre, apellido_1, apellido_2, email, tipo, nif,
destinatario, calle, numero, piso, poblacion, cp, id_provincia, id_pais,
tel_contacto, factura, f_destinatario, f_id_direccion, f_nif, f_calle, f_numero,
f_piso, f_cp, f_poblacion, f_id_provincia, f_id_pais, fecha, fecha_entrega,
subtotal, estado, plataforma, user_idioma, id_zona, id_language, ip, session_id,
id_moneda,
```

```
case
```

```
when ratio_moneda<>1 and date(fecha)>='2015-02-27' then ratio_moneda
```

```
when ratio_moneda<>1 and date(fecha) between '2015-01-01' and '2015-02-
26' then 1.33
```

```
when ratio_moneda<>1 and date(fecha)<'2015-01-01' then 1.22
```

```
else ratio_moneda
```

```
end as ratio_moneda, id_pago, id_direccion, modo_pago, gastos_envio,
costes_envio, regalos, suplementos, importe_pendiente_pagar, total, follow,
origen, codigos, migracion, frontal, enviado_mail_tras_entrega, is_app,
is_notificado, id_punto_recogida, nombre_establecimiento_punto_recogida,
tipo_envio
```

```
FROM `motobuykers_produccion`.pedidos;
```

```
Store Pedidos into '$(vRutaQVD)Dashboard\Pedidos.qvd' (qvd);
```

```
drop table Pedidos;
```

pedidos_lineas:

```
SQL SELECT *
```

```
FROM `motobuykers_produccion`.`pedidos_lineas`;
```

```
Store pedidos_lineas into '$(vRutaQVD)Dashboard\pedidos_lineas.qvd' (qvd);
```

```
drop table pedidos_lineas;
```

abonos:

```
LOAD *;
```

```
SQL SELECT a.id, a.id_pedido, a.id_movimiento_pedido, a.fecha, a.concepto,  
a.origen, a.estado, abs(f.subtotal) as importe, a.divisa FROM  
`motobuykers_produccion`.abonos a inner join facturas f on a.id=f.id_abono  
and f.origen=2 where estado=1;
```

```
Store abonos into '$(vRutaQVD)Dashboard\abonos.qvd' (qvd);
```

```
drop table abonos;
```

productos_atributos:

```
load *;
```

```
SQL select
```

```
pa.id_producto,  
replace(a.nombre,'att_',") as atributo_padre,  
replace(av.valor,'att_',") as atributo,  
if(a.tiene_stock,'Sí','No') as tiene_stock,  
pa.stock,  
pa.venta_no_stock,  
ifnull((select 'Pasada'  
from productos pro  
inner join campanias c on pro.id_campania=c.id  
where pa.id_producto=pro.id  
and c.fin<now()),'Activo') as pa_estado
```

```
from productos_atributos pa
```

```
inner join atributos_valores av on pa.id_atributo=av.id_atributo and  
pa.id_valor=av.id
```

```
inner join atributos a on pa.id_atributo=a.id;
```

```
STORE products_atributos into
```

```
'$(vRutaQVD)Dashboard\productos_atributos.qvd' (qvd);
```

```
drop table productos_atributos;
```

CoordenadasCP:

```
LOAD *;
SQL SELECT *
FROM BI.Coordenadas_Codigos_Postales;
STORE CoordinadasCP into '$(vRutaQVD)Dashboard\CoordenadasCP.qvd'
(qvd);
DROP TABLE CoordinadasCP;
```

PedidosDatosExtra:

```
LOAD *;
SQL SELECT *
FROM pedidos_datos_extra;
STORE PedidosDatosExtra into
'$(vRutaQVD)Dashboard\PedidosDatosExtra.qvd' (qvd);
DROP TABLE PedidosDatosExtra;
```

PedidosEstadisticas:

```
LOAD *;
SQL SELECT *
FROM estadisticas_pedidos;
STORE PedidosEstadisticas into
'$(vRutaQVD)Dashboard\PedidosEstadisticas.qvd' (qvd);
DROP TABLE PedidosEstadisticas;
```

SCI:

```
LOAD *;
SQL SELECT *
FROM BI.sci_usuarios_segmento;
STORE SCI into '$(vRutaQVD)Dashboard\SCI.qvd' (qvd);
DROP TABLE SCI;
```

usuarios_moto:

```
LOAD *;
SQL SELECT *
FROM usuarios_moto;
```

```
STORE usuarios_moto into '$(vRutaQVD)Dashboard\usuarios_moto.qvd' (qvd);
DROP TABLE usuarios_moto;
```

motos:

```
LOAD *;
SQL select id, marca, marca_url, modelo, modelo_url, cubicaje, denominacion,
anio, tipo, estado, concat(id,'||',
    marca,'||',
    modelo,'||',
    cubicaje,'||',
    if(denominacion is null or denominacion=",'@@',denominacion)) as
modelo_completo, productos, productos_universales, top, estilo_1, estilo_2,
estilo_3, estilo_4
from motos;
STORE motos INTO '$(vRutaQVD)Dashboard\motos.qvd' (qvd);
DROP TABLE motos;
```

lead_scoring_categorias:

```
LOAD *;
SQL SELECT *
FROM Webpower.lead_scoring_categorias_import;
STORE          lead_scoring_categorias          INTO
'$(vRutaQVD)Dashboard\lead_scoring_categorias.qvd' (qvd);
DROP TABLE lead_scoring_categorias;
```

lead_scoring_marcas:

```
LOAD *;
SQL SELECT *
FROM Webpower.lead_scoring_marcas_import;
STORE          lead_scoring_marcas          INTO
'$(vRutaQVD)Dashboard\lead_scoring_marcas.qvd' (qvd);
DROP TABLE lead_scoring_marcas;
```

facturas:

```
LOAD *;  
SQL SELECT *  
FROM facturas  
where origen=1;  
STORE facturas INTO '$(vRutaQVD)Dashboard\facturas.qvd' (qvd);  
DROP TABLE facturas;
```

provincias:

```
LOAD *;  
SQL SELECT *  
FROM provincias;  
STORE provincias INTO '$(vRutaQVD)Dashboard\provincias.qvd' (qvd);  
DROP TABLE provincias;
```

GA_RowData_Qlik:

```
LOAD *;  
SQL SELECT *  
FROM BI.GA_RowData_Qlik;  
STORE GA_RowData_Qlik INTO  
'$(vRutaQVD)Dashboard\GA_RowData_Qlik.qvd' (qvd);  
DROP TABLE GA_RowData_Qlik;
```

GA_RowData_Qlik_Transacciones:

```
LOAD *;  
SQL SELECT *  
FROM BI.GA_RowData_Qlik_Transacciones;  
STORE GA_RowData_Qlik_Transacciones INTO  
'$(vRutaQVD)Dashboard\GA_RowData_Qlik_Transacciones.qvd' (qvd);  
DROP TABLE GA_RowData_Qlik_Transacciones;
```

devoluciones:

```
LOAD *;
```



```
SQL SELECT *
FROM devoluciones;
STORE devoluciones INTO '$(vRutaQVD)Dashboard\devoluciones.qvd' (qvd);
DROP TABLE devoluciones;
```

devoluciones_lineas:

```
LOAD *;
SQL SELECT *
FROM devoluciones_lineas;
STORE devoluciones_lineas INTO
'$(vRutaQVD)Dashboard\devoluciones_lineas.qvd' (qvd);
DROP TABLE devoluciones_lineas;
```

popup_reincidente:

```
LOAD *;
SQL SELECT *
FROM popup_reincidente;
STORE popup_reincidente INTO
'$(vRutaQVD)Dashboard\popup_reincidente.qvd' (qvd);
DROP TABLE popup_reincidente;
```

GA_RowData_Qlik_Inf_Productos:

```
LOAD *;
SQL SELECT *
FROM BI.GA_RowData_Qlik_Inf_Productos
where fecha>date_sub(now(), interval 1 month);
STORE GA_RowData_Qlik_Inf_Productos INTO
'$(vRutaQVD)Dashboard\GA_RowData_Qlik_Inf_Productos.qvd' (qvd);
DROP TABLE GA_RowData_Qlik_Inf_Productos;
```

adwords:

```
LOAD *;
SQL SELECT *
FROM BI.adwords;
```

```
STORE adwords INTO '$(vRutaQVD)Dashboard\adwords.qvd' (qvd);  
DROP TABLE adwords;
```

zonas:

```
LOAD *;  
SQL SELECT *  
FROM zonas;  
STORE zonas INTO '$(vRutaQVD)Dashboard\zonas.qvd' (qvd);  
DROP TABLE zonas;
```

netrivals_producto:

```
LOAD *;  
SQL SELECT *  
FROM BI.netrivals_producto;  
STORE netrivals_producto INTO  
 '$(vRutaQVD)Dashboard\netrivals_producto.qvd' (qvd);  
DROP TABLE netrivals_producto;
```

netrivals_child:

```
LOAD *;  
SQL SELECT *  
FROM BI.netrivals_child;  
STORE netrivals_child INTO '$(vRutaQVD)Dashboard\netrivals_child.qvd'  
(qvd);  
DROP TABLE netrivals_child;
```

objetivo_empresa:

```
LOAD *;  
SQL SELECT *  
FROM BI.objetivo_empresa;  
STORE objetivo_empresa INTO  
 '$(vRutaQVD)Dashboard\objetivo_empresa.qvd' (qvd);  
DROP TABLE objetivo_empresa;
```

```

objetivo_empresa_canal:
LOAD *;
SQL SELECT *
FROM BI.objetivo_empresa_canal;
STORE                objetivo_empresa_canal                INTO
'$(vRutaQVD)Dashboard\objetivo_empresa_canal.qvd' (qvd);
DROP TABLE objetivo_empresa_canal;

```

```

tipologia_usuario:
LOAD *;
SQL SELECT
    n.id,
    case
        when estado=4 then 'Invitado'
        when cantidad>1 then 'Repeater'
        when cantidad=1 then 'Buyer'
        when tiene_moto=1 or tiene_estilo=1 then 'Prospect'
        else 'Lead'
    end as tipologia_usuario
from (
select
    u.id,
    ifnull(pe.cantidad,0) as cantidad,
    (select 1 from usuarios_moto um where u.id=um.id_usuario group by 1) as
tiene_moto,
    if(estilo_1 is null,0,1) as tiene_estilo,
    u.estado
from usuarios u
left join (select p.id_usuario, count(1) as cantidad
from pedidos p
where p.estado in (1,2)
group by 1) as pe on u.id=pe.id_usuario) as n;

```

```
STORE tipologia_usuario INTO '$(vRutaQVD)Dashboard\tipologia_usuario.qvd'  
(qvd);
```

```
DROP TABLE tipologia_usuario;
```

```
tipologia_tpv:
```

```
load *;
```

```
sql select
```

```
mp.id_pedido,
```

```
case
```

```
when respuesta_en_bruto like '{"@ATT%' then 'Tokenizado'
```

```
when respuesta_en_bruto like '%eur3ds%' then '3DSecure-Santander'
```

```
when respuesta_en_bruto like '%gbp3ds%' then '3DSecure-Santander'
```

```
when respuesta_en_bruto like '%, "Ds_Terminal": "004", %' then '3DSecure-  
Sabadell'
```

```
when respuesta_en_bruto like '%, "Ds_Terminal": "005", %' then '3DSecure-  
Sabadell'
```

```
end as Tipo_tpv
```

```
from movimientos_pedido mp
```

```
where
```

```
mp.tipo_transaccion=0
```

```
and (respuesta_en_bruto like '{"@ATT%'
```

```
or respuesta_en_bruto like '%eur3ds%'
```

```
or respuesta_en_bruto like '%gbp3ds%'
```

```
or respuesta_en_bruto like '%, "Ds_Terminal": "004", %'
```

```
or respuesta_en_bruto like '%, "Ds_Terminal": "005", %');
```

```
STORE tipologia_tpv INTO '$(vRutaQVD)Dashboard\tipologia_tpv.qvd' (qvd);
```

```
DROP TABLE tipologia_tpv;
```

```
usuario_tarjetas:
```

```
LOAD *;
```

```
SQL select
```

```
id_usuario,
```

```
1 as Tarjeta_Tokenizada,
```

```
ut.created_on
```

```

from usuario_tarjetas ut;
STORE usuario_tarjetas INTO '$(vRutaQVD)Dashboard\usuario_tarjetas.qvd'
(qvd);
DROP TABLE usuario_tarjetas;

```

usuario_tarjetas_dedu:

```

LOAD *;
SQL select
    id_usuario,
    1 as Tarjeta_Tokenizada
from usuario_tarjetas
group by 1;
STORE          usuario_tarjetas_dedu          INTO
'$(vRutaQVD)Dashboard\usuario_tarjetas_dedu.qvd' (qvd);
DROP TABLE usuario_tarjetas_dedu;

```

stock_almacen:

```

LOAD *;
SQL select
    pal.referencia_producto as Almacen_Producto,
    av.valor as Almacen_Talla,
    count(distinct un.id_unidad) as Almacen_Cantidad,
    pr.nombre as Almacen_Proveedor,
    m.nombre as Almacen_Marca,
    ca.nombre_permanente as Almacen_Categoria,
    ua.nombre as Almacen_Ubicacion,
    un.fecha as Fecha,
    un.coste as Almacen_Coste,
    count(distinct un.id_unidad)*un.coste as Almacen_Coste_Total,
    pro.id
from producto_almacen pal
    inner join unidades un on pal.id=un.id_producto
    inner join productos pro on pro.referencia=pal.referencia_producto and
pro.id_partner=pal.id_proveedor

```

```

inner join atributos_valores av on un.id_atributo_valor=av.id
inner join marcas m on pro.id_marca=m.id
inner join categorias ca on pal.id_categoria=ca.id
inner join proveedores pr on pal.id_proveedor=pr.id
inner join ubicacion_almacen ua on un.id_ubicacion=ua.id
group by 1,2,7,8,9;
STORE stock_almacen INTO '$(vRutaQVD)Dashboard\stock_almacen.qvd'
(qvd);
DROP TABLE stock_almacen;

```

historico_vozitel:

```

LOAD *;
SQL
SELECT ID,
        DIR,
        CALLID,
        CHANNEL,
        CE_EXTENSION,
        HANGUP_CAUSE,
        CLASIFICACION,
        FECHA,
        FECHA_DIA,
        REFERENCIA,
        CALLAPIID,
        CALLERIDNUM,
        WAITTIME,
        SLA,
        WHU,
        CALLERIDNAME,
        CALLEDDID,
        CALLBACK,
        FLUJO,
        CALLEDNUMBER,

```

```

HANGUP,
DURACION,
FILERECORDED,
`NUM`,
ESTADO_CLASIFICACION,
case
when CE_EXTENSION in (500,800,200,201,204,205,206,207,850) then
'Support ES'
when CE_EXTENSION in (202,501,801) then 'Support IT'
when CE_EXTENSION in (203,502,802) then 'Support FR'
else 'Error'
end as Vozitel_Group_Name,
case
when CE_EXTENSION='200' THEN 'Noel'
when CE_EXTENSION='201' THEN 'Carla'
when CE_EXTENSION='202' THEN 'Ilenia'
when CE_EXTENSION='203' THEN 'Bene'
when CE_EXTENSION='204' THEN 'Ramon'
when CE_EXTENSION='205' THEN 'Magatzem'
when CE_EXTENSION='206' THEN 'Fixe'
when CE_EXTENSION='207' THEN 'Barcelona'
when CE_EXTENSION='500' THEN 'Cola España'
when CE_EXTENSION='501' THEN 'Cola Italia'
when CE_EXTENSION='502' THEN 'Cola Francia'
when CE_EXTENSION='800' THEN 'IVR España'
when CE_EXTENSION='801' THEN 'IVR Italia'
when CE_EXTENSION='802' THEN 'IVR Francia'
when CE_EXTENSION='850' THEN 'IVR Desbordamiento'
else 'Error'
end as Vozitel_Group_Support,
case
when CE_EXTENSION in (500,800,200,201,204,205,206,207,850) and
(time(fecha) between '09:30' and '13:00' or time(fecha) between '15:00' and
'17:00') then 'En horario'

```

```

when CE_EXTENSION in (202,501,801) and time(Fecha) between '09:30' and
'13:00' then 'En horario'
when CE_EXTENSION in (203,502,802) and time(Fecha) between '09:30' and
'13:00' then 'En horario'
else 'Fuera horario'
end as Vozitel_Dentro_Horario
FROM BI.historico_vozitel;
STORE historico_vozitel INTO '$(vRutaQVD)Dashboard\historico_vozitel.qvd'
(qvd);
DROP TABLE historico_vozitel;

```

historico_zendesk_tickets:

```

LOAD *;
SQL SELECT
url,
id,
external_id,
channel,
`from`,
`to`,
rel,
FROM_UNIXTIME(unix_created_at) as Fecha,
from_unixtime(unix_updated_at) as Fecha_update,
`type`,
subject,
description,
priority,
status,
recipient,
requesterEmail,
requesterPhone,
submitterEmail,
submitterPhone,
assignee_id,

```



```

assignee_email,
organization_id,
group_name,
has_incidents,
score,
score_comment,
custom_field_1,
custom_field_1_value,
custom_field_2,
custom_field_2_value,
custom_field_3,
custom_field_3_value,
custom_field_4,
custom_field_4_value,
custom_field_5,
custom_field_5_value,
if(
(hzt.custom_field_4_value='correo'
and hzt.submitterEmail not like '%@motobuykers.com')
or
exists (select 'Ok' from BI.historico_zendesk_tags hAux where hzt.id=hAux.id
and hAux.tag='zopim_offline_message'),
(select min(hzc.unix_created_at) from BI.historico_zendesk_comments hzc
where hzc.ticket_id=hzt.id and hzc.author_email like '%@motobuykers.com'),
null
) as fecha_primera_respuesta,
unix_created_at
FROM BI.historico_zendesk_tickets hzt;
STORE          historico_zendesk_tickets          INTO
'$(vRutaQVD)Dashboard\historico_zendesk_tickets.qvd' (qvd);
DROP TABLE historico_zendesk_tickets;

historico_zendesk_tags:
LOAD *;

```

```

SQL select
  id,
  `tag`
FROM BI.historico zendesk_tags;
STORE          historico zendesk_tags          INTO
'$(vRutaQVD)Dashboard\historico zendesk_tags.qvd' (qvd);
DROP TABLE historico zendesk_tags;

```

historico zendesk_comments:

```

LOAD *;
SQL select
  id,
  ticket_id,
  body,
  public,
  author_id,
  author_email,
  author_phone,
  from_unixtime(unix_created_at) as Fecha_Comment_Created
FROM BI.historico zendesk_comments;
STORE          historico zendesk_comments          INTO
'$(vRutaQVD)Dashboard\historico zendesk_comments.qvd' (qvd);
DROP TABLE historico zendesk_comments;

```

historico_transacciones_webgains_criteo:

```

LOAD *;
SQL select
  'Criteo' as Retargeting_Partner,
  `Transaction id` as id_pedido,
  `Click date` as click_date,
  `Duplicate status` as duplicate_status,
  null as Partner_Afiliado
from BI.historico_transacciones_criteo

```

```

union all
select
  'Webgains' as Retargeting_Partner,
  Número_de_Referencia as id_pedido,
  Hora_del_click as click_date,
  null,
  Afiliado as Partner_Afiliado
from BI.historico_transacciones_webgains;
STORE          historico_transacciones_webgains_criteo          INTO
'$(vRutaQVD)Dashboard\historico_transacciones_webgains_criteo.qvd' (qvd);
DROP TABLE historico_transacciones_webgains_criteo;

```

expediciones:

```

LOAD *;
SQL select
  e.id,
  e.id_proveedor_logistico,
  pl.nombre as nombre_proveedor_logistico,
  e.contra_reembolso_importe,
  e.contra_reembolso_moneda,
  e.fecha_creacion,
  e.fecha_expedicion,
  e.fecha_actualizacion,
  e.estado,
  e.tipo,
  e.ttr,
  e.bultos,
  e.id_almacen,
  ea.almacen as nombre_almacen,
  e.courier,
  e.is_reexpedicion,
  e.peso,
  e.coste,

```

```

e.peso_real,
upper(ifnull(e.dr_pais,e.de_pais)) as pais,
referencia_motobuykers,
referencia_proveedor_logistico
from expediciones e
  inner join expediciones_almacenes ea on e.id_almacen=ea.id
  inner join proveedores_logisticos pl on e.id_proveedor_logistico=pl.id
where
  fecha_expedicion is not null;
STORE expediciones INTO '$(vRutaQVD)Dashboard\expediciones.qvd' (qvd);
DROP TABLE expediciones;
Endif

```

```

if (1=1) then

```

```

Analytics_Transacciones_Device:

```

```

Mapping LOAD

```

```

  transactionId,
  deviceCategory

```

```

FROM '$(vRutaQVD)Dashboard\GA_RowData_Qlik_Transacciones.qvd' (qvd);

```

```

Analytics_Transacciones_channelGrouping:

```

```

Mapping LOAD

```

```

  transactionId,
  channelGrouping as Transaccion_channelGrouping

```

```

FROM '$(vRutaQVD)Dashboard\GA_RowData_Qlik_Transacciones.qvd' (qvd);

```

```

Analytics_Transacciones_Source:

```

```

Mapping LOAD

```

```

  transactionId,
  source

```

```

FROM '$(vRutaQVD)Dashboard\GA_RowData_Qlik_Transacciones.qvd' (qvd);

```

Analytics_Transacciones_Medium:

Mapping LOAD

 transactionId,
 medium as Transaccion_medium

FROM '\$(vRutaQVD)Dashboard\GA_RowData_Qlik_Transacciones.qvd' (qvd);

Analytics_Transacciones_Campaign:

Mapping LOAD

 transactionId,
 adwordsCampaignID

FROM '\$(vRutaQVD)Dashboard\GA_RowData_Qlik_Transacciones.qvd' (qvd);

Analytics_Transacciones_Revenue:

Mapping LOAD

 transactionId,
 transactionRevenue

FROM '\$(vRutaQVD)Dashboard\GA_RowData_Qlik_Transacciones.qvd' (qvd);

Analytics_Transacciones_Shipping:

Mapping LOAD

 transactionId,
 transactionShipping

FROM '\$(vRutaQVD)Dashboard\GA_RowData_Qlik_Transacciones.qvd' (qvd);

AuxDíasVenta:

LOAD *

INLINE [

DíasVenta

1

2

3

4

5

6

7
8
9
10
11
12
13
14
15
16
17
18
19
20
];

MapaAdwords:

MAPPING LOAD

Campaign,

Campaign

FROM '\$(vRutaQVD)Dashboard\Adwords.qvd' (qvd);

MapaPedidoUsuario:

MAPPING LOAD

id,

id_usuario

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd) where estado>0;

MapaProductosStoreOutlet:

Mapping Load

id,

id_campania

from '\$(vRutaQVD)Dashboard\productos.qvd' (qvd);

MapaProductosNombreParaLineas:

Mapping Load

```
id_producto as Netrivals_pk_producto,
nombre
from '$(vRutaQVD)Dashboard\productos_zonas.qvd' (qvd)
where id_zona=1;
inner join
load
Code as Netrivals_pk_producto
from '$(vRutaQVD)Dashboard\netrivals_child.qvd' (qvd);
```

MapaDescCalificacion:

Mapping LOAD

```
id,
`desc`;
SQL SELECT id,
`desc`
FROM `motobuyers_produccion`.`desc_calificacion_usuarios`;
```

MapaDescInversionCanal:

Mapping LOAD

```
Afiliado & '|' & Familia_Afiliado,
Canal
FROM [lib://Sense_Dashboard
(vps229099_administrator)/historico_inversion_qlik.xlsx]
(ooxml, embedded labels, table is Descripcion_Canal);
```

MapaProductosPartner:

Mapping Load

```
id,
id_partner
from '$(vRutaQVD)Dashboard\productos.qvd' (qvd);
```

MapaTipologiaUsuario:

Mapping Load

id,

tipologia_usuario

from '\$(vRutaQVD)Dashboard\tipologia_usuario.qvd' (qvd);

MapaTipologiaTPV:

Mapping Load

id_pedido,

Tipo_tpv

from '\$(vRutaQVD)Dashboard\tipologia_tpv.qvd' (qvd);

MapaZonas:

MAPPING LOAD

id,

id_zona

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaFechas:

MAPPING LOAD

id,

fecha

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaZonasIvaGeneral:

Mapping load

id,

iva_general

from '\$(vRutaQVD)Dashboard\zonas.qvd' (qvd);

MapaNombreCampana:

Mapping LOAD

id,


```
    `nombre_unico`  
from '$(vRutaQVD)Dashboard\campanias.qvd' (qvd);
```

```
LIB CONNECT TO 'Produccion (vps229099_administrator)';
```

Latitudes:

```
LOAD
```

```
    id AS Provincia,  
    GeoMakePoint(latitud, longitud) AS GeoReferencia,  
    latitud,  
    longitud
```

```
FROM [lib://DatosLocal (vps229099_administrator)/Dashboard\Latitudes.qvd]  
(qvd);
```

CoordenadasCP:

```
LOAD
```

```
    CountryCode&'- '&PostalCode as PK_Coordenadas_Codigo_Postal,  
    Latitude,  
    Longitude,  
    PlaceName as CiudadCP
```

```
from '$(vRutaQVD)Dashboard\CoordenadasCP.qvd' (qvd);
```

Categorías:

```
LOAD DISTINCT id_producto AS PK_Producto,
```

```
Categoria AS Categoría,
```

```
Subcategoria AS Subcategoría
```

```
from '$(vRutaQVD)Dashboard\categorias.qvd' (qvd);
```

MapaMarcas:

```
MAPPING LOAD id,  
    nombre  
from '$(vRutaQVD)Dashboard\marcas.qvd' (qvd);
```

MapaProveedores:

```
MAPPING LOAD DISTINCT id,  
    nombre  
from '$(vRutaQVD)Dashboard\proveedores.qvd' (qvd);
```

MapaNombreProducto:

```
MAPPING LOAD `id_producto`,  
    nombre  
from '$(vRutaQVD)Dashboard\productos_zonas.qvd' (qvd)  
WHERE id_zona=1  
;
```

MapaTieneVideo:

```
MAPPING LOAD `id_producto`,  
    Tiene_video  
from '$(vRutaQVD)Dashboard\productos_zonas.qvd' (qvd)  
WHERE id_zona=1  
;
```

MapaResponsable:

```
MAPPING LOAD DISTINCT id,  
    admin_users_username  
from '$(vRutaQVD)Dashboard\proveedores.qvd' (qvd);
```

Campañas:

```
LOAD id AS PK_Campaña,  
    `nombre_unico` AS Campaña,  
    referencia AS [Ref. Campaña] ,
```

```

date(inicio) AS [Fecha Inicio Campaña],
date(fin) AS [Fecha Fin Campaña],
1 as [Conteo Campañas]
from '$(vRutaQVD)Dashboard\campanias.qvd' (qvd);

```

Productos:

```

LOAD id AS PK_Producto,
    //Applymap('MapaProveedores',id_partner,'Sin Proveedor') AS Proveedor
    Applymap('MapaMarcas`,`id_marca`,`Sin Marca') AS Marca,
    Applymap('MapaNombreProducto`,`id`,`Sin Nombre Producto') AS Producto,
id_campania AS PK_Campaña,
referencia AS [Referencia Producto],
`referencia_modelo` AS [Referencia Modelo],
orden,
coste as productos_coste,
estado as productos_estado,
if(len(id_campania)=0,'Store','Outlet') as [Producto Store/Outlet],
ApplyMap('MapaTieneVideo',id,'Sin video') as Tiene_video

from '$(vRutaQVD)Dashboard\productos.qvd' (qvd);

```

MapaProductosMarca:

Mapping Load

```

    id,
    Applymap('MapaMarcas`,`id_marca`,`Sin Marca') AS Marca
from '$(vRutaQVD)Dashboard\productos.qvd' (qvd);

```

MapaProductosCampaña:

Mapping Load

```

    id,
    id_campania

```

```
from '$(vRutaQVD)Dashboard\productos.qvd' (qvd);
```

MapaProductosReferencia:

Mapping Load

id,

referencia

```
from '$(vRutaQVD)Dashboard\productos.qvd' (qvd);
```

MapaProductosReferenciaModelo:

Mapping Load

id,

referencia_modelo

```
from '$(vRutaQVD)Dashboard\productos.qvd' (qvd);
```

concatenate (Productos)

Load

id_producto as PK_Producto,

atributo_padre as [Atributo Padre],

atributo as Atributo,

tiene_stock as [Atributo Stock],

stock as Stock,

if(venta_no_stock=1,'Sí','No') as [Venta no stock],

Applymap('MapaNombreProducto',id_producto,'Sin Nombre Producto') AS
Producto,

Applymap('MapaProductosMarca',id_producto,'Sin Marca') as Marca,

Applymap('MapaProductosCampaña',id_producto,'Sin Campaña') as
PK_Campaña,

Applymap('MapaProductosReferencia',id_producto,'Sin Referencia') as
[Referencia Producto],

Applymap('MapaProductosReferenciaModelo',id_producto,'Sin Referencia
Modelo') as [Referencia Modelo],

pa_estado,

```
if(len(ApplyMap('MapaProductosStoreOutlet',id_producto))=0,'Store','Outlet')
as [Producto Store/Outlet]
from '$(vRutaQVD)Dashboard\productos_atributos.qvd' (qvd);
```

MapaCampaña:

MAPPING LOAD

id,

id_campania

from '\$(vRutaQVD)Dashboard\productos.qvd' (qvd);

MapaTallas:

MAPPING LOAD id,

replace(valor, 'att_', '')

from '\$(vRutaQVD)Dashboard\atributos_valores.qvd' (qvd);

MapaUltimaFechaExpedicion:

MAPPING LOAD

ID_PEDIDO_LINEA,

date(FECHA_EXPEDICION)

from '\$(vRutaQVD)Dashboard\FechaExpedicion.qvd' (qvd);

MapaUltimaReferenciaExpedicion:

MAPPING LOAD DISTINCT

ID_PEDIDO_LINEA,

REF_EXPEDICION

from '\$(vRutaQVD)Dashboard\ReferenciaExpedicion.qvd' (qvd);

MapaUltimoIDExpedicion:

MAPPING LOAD DISTINCT

ID_PEDIDO_LINEA,

ID_EXPEDICION

from '\$(vRutaQVD)Dashboard\IDExpedicion.qvd' (qvd);

MapaUltimaFechaEntrega:

MAPPING LOAD ID_PEDIDO_LINEA,

date(FECHA_ENTREGA)

from '\$(vRutaQVD)Dashboard\FechaEntrega.qvd' (qvd);

MapaRegalos:

MAPPING LOAD id,

regalos

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaGastosEnvio:

MAPPING LOAD id,

gastos_envio

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaCostesEnvio:

MAPPING LOAD id,

costes_envio

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaSuplementos:

MAPPING LOAD id,

suplementos

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaEstado:

MAPPING LOAD

id,

estado

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaFechaPedido:

MAPPING LOAD

id,

fecha

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaRatio:

MAPPING LOAD

id,

ratio_moneda

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaImportePedidos:

MAPPING LOAD

id_pedido,

sum(precio*cantidad / (1+(iva/100)))

from '\$(vRutaQVD)Dashboard\pedidos_lineas.qvd' (qvd)

where Applymap('MapaEstado',id_pedido)>0

group by id_pedido

;

MapaProvincia:

MAPPING LOAD

id,

id_provincia

from '\$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

MapaPedidosDatosExtra:

MAPPING LOAD

```

id_pedido,
if(is_paypal_express=1,'Paypal Express','PayPal Normal')
from '$(vRutaQVD)Dashboard\PedidosDatosExtra.qvd' (qvd);

```

Lineas:

LOAD

```

    'Líneas' AS TipoLinea,
    id AS [id Linea Pedido],
    Applymap('MapaCampaña',id_producto) AS CampañaTemp,

    Applymap('MapaUltimoIDExpedicion',id,'Sin ID') AS [ID Expedición],
    Applymap('MapaUltimaReferenciaExpedicion',id,'Sin Referencia') AS
[Referencia Expedición],
    Applymap('MapaUltimaFechaExpedicion',id,'Error') AS [Fecha Expedición],
    date(`fecha_entrega`) AS [Fecha Entrega Prevista],
    Applymap('MapaUltimaFechaEntrega',id,'NoEntregado') AS [Fecha Entrega
Real],
    if( len(Applymap('MapaUltimaFechaEntrega',id,'Error'))=0,0,
    Applymap('MapaUltimaFechaEntrega',id,'Error') - date(`fecha_entrega`)) AS
[Dif. Prevista Real],
    Applymap('MapaUltimaFechaEntrega',id,'Error')-
Applymap('MapaUltimaFechaExpedicion',id,'Error') AS [Dif. Expedición Real],

    if(len(Applymap('MapaUltimaFechaEntrega',id,'Error'))='Error' and today()-
date(`fecha_entrega`)>0,
    'Pdte Entrega - Retraso',
    if(len(Applymap('MapaUltimaFechaEntrega',id,'Error'))='Error' and today()-
date(`fecha_entrega`)<=0,
    'Pendiente Entrega',
    if(date(Applymap('MapaUltimaFechaEntrega',id,'Error'))-
date(`fecha_entrega`)>3
    AND
date(Applymap('MapaUltimaFechaEntrega',id,'Error'))-date(`fecha_entrega`)<7,
    'Retraso Entrega',

```



```

        if(date(Applymap('MapaUltimaFechaEntrega',id,'Error'))-
date(`fecha_entrega`)>=7,
        'Retraso Entrega + Cliente Cabreado',
        if(date(Applymap('MapaUltimaFechaEntrega',id,'Error'))-
date(`fecha_entrega`)<-3                                AND
date(Applymap('MapaUltimaFechaEntrega',id,'Error'))-date(`fecha_entrega`)>-7,
        'Entrega Con Margen',
        if(date(Applymap('MapaUltimaFechaEntrega',id,'Error'))-
date(`fecha_entrega`)<=-7,
        'Entrega Con Margen + Fechas Mal Ajustadas',
        '+/- On Time'
        )
        )
        )
        )
        ) AS [Agrup. Tipo Entrega],

```

```

        date(Applymap('MapaUltimaFechaEntrega',id,'Error'))-date(`fecha_entrega`)
as [Test Agrupacion],

```

```

        if(Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')='NoEntregado' or
Applymap('MapaUltimaFechaExpedicion',id,'NoExpedido')='NoExpedido',
        null(),
        if(Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')-
Applymap('MapaUltimaFechaExpedicion',id,'NoExpedido')>5                                and
Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')-
Applymap('MapaUltimaFechaExpedicion',id,'NoExpedido')<10,
        'Retraso OL',
        if(Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')-
Applymap('MapaUltimaFechaExpedicion',id,'NoExpedido')>=10                                and
Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')-
Applymap('MapaUltimaFechaExpedicion',id,'NoExpedido')<15,

```

```

'Retraso OL Alto - Check',
  if(Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')-
Applymap('MapaUltimaFechaExpedicion',id,'NoExpedido')>=15,
  'Retraso OL no se pueden repetir - CHECK',
  'En Tiempo OL')
  )
  )
) as [Entrega Operador Logístico],

if(Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')='NoEntregado',
  null(),

  if(date(` fecha_entrega`)<=date(Applymap('MapaUltimaFechaEntrega',id))
,

  NetWorkDays(date(` fecha_entrega`),date(Applymap('MapaUltimaFechaE
ntrega',id))),

NetWorkDays(date(Applymap('MapaUltimaFechaEntrega',id)),date(` fecha_entr
ega`))*-1
  )) as [Dif. Entrega Prevista vs Real],

if(Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')='NoEntregado',
  'Sin datos',
  if(date(` fecha_entrega`)<=date(Applymap('MapaUltimaFechaEntrega',id))
and
NetWorkDays(date(` fecha_entrega`),date(Applymap('MapaUltimaFechaEntrega
',id)))>3,
  '+ 3 days later',
  if(date(` fecha_entrega`)<=date(Applymap('MapaUltimaFechaEntrega',id))
and

```

```

NetWorkDays(date(`fecha_entrega`),date(Applymap('MapaUltimaFechaEntrega',id))>1,
    '1-3 days later',
    if(date(`fecha_entrega`)>date(Applymap('MapaUltimaFechaEntrega',id))
and
NetWorkDays(date(Applymap('MapaUltimaFechaEntrega',id)),date(`fecha_entrega`))*-1<-3,
    '+ 3 days before',
    if(date(`fecha_entrega`)>date(Applymap('MapaUltimaFechaEntrega',id))
and
NetWorkDays(date(Applymap('MapaUltimaFechaEntrega',id)),date(`fecha_entrega`))*-1<=-1,
    '1-3 days before',
    'Ontime')))) as [Dif. Entrega Prevista vs Real Agrupado],

```

```

if(Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')='NoEntregado',
    null(),

```

```

NetWorkDays(Applymap('MapaFechaPedido',id_pedido),Applymap('MapaUltimaFechaEntrega',id))
    ) as [Dif. Fecha Pedido vs Entrega Real],

```

```

if(Applymap('MapaUltimaFechaEntrega',id,'NoEntregado')='NoEntregado',
    'Sin datos',

```

```

if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),Applymap('MapaUltimaFechaEntrega',id))<=2,
    '1-2 days',

```

```

if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),Applymap('MapaUltimaFechaEntrega',id))<=5,
    '3-5 days',

```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),Applymap('MapaUltimaFechaEntrega',id))<=7,  
    '6-7 days',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),Applymap('MapaUltimaFechaEntrega',id))<=10,  
    '8-10 days',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),Applymap('MapaUltimaFechaEntrega',id))<=15,  
    '11-15 days',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),Applymap('MapaUltimaFechaEntrega',id))<=20,  
    '16-20 days',  
    '>20 days'
```

```
)))))) as [Dif. Fecha Pedido vs Entrega Real Agrupado],
```

```
NetWorkDays(Applymap('MapaFechaPedido',id_pedido),date(`fecha_entrega`))  
as [Dif. Fecha Pedido vs Entrega Prevista],
```

```
if(date(`fecha_entrega`)=null(),  
    'Sin datos',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),date(`fecha_entrega`))<=2,  
    '1-2 days',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),date(`fecha_entrega`))<=5,  
    '3-5 days',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),date(`fecha_entrega`))<=7,  
    '6-7 days',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),date(`fecha_entrega`))<=10,  
    '8-10 days',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),date(`fecha_entrega`))<=15,  
    '11-15 days',
```

```
if(NetWorkDays(Applymap('MapaFechaPedido',id_pedido),date(`fecha_entrega`))<=20,  
    '16-20 days',  
    '>20 days'  
)))))) as [Dif. Fecha Pedido vs Entrega Prevista Agrupado],
```

```
//Para los indicadores de entrega me genero unas marcas porque solo  
querré contar
```

```
//los que hayan traído valor.
```

```
if(    len(Applymap('MapaUltimaFechaEntrega',id,'Error'))>0,1,0)    AS  
CheckEntregado,
```

```
if(    len(Applymap('MapaUltimaFechaExpedicion',id,'Error'))>0,1,0)    AS  
CheckExpedido,
```

```

        if( len(Applymap('MapaUltimaFechaEntrega',id,'Error'))=0 and today()-
date(`fecha_entrega`)>0,
        'Pdte Entrega - Retraso',
        'Pdte Entrega') AS [Pdte. Entrega],

```

```

`id_pedido` AS Pedido_Linea_Id_Pedido,
`id_pedido` AS PK_Pedido,
`id_producto` AS PK_Producto,
        ((precio*cantidad /
(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,75,76),0,iva)/100)))
+
        (Applymap('MapaGastosEnvio',id_pedido,0) /
(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,75,76),0,iva)/100)))
/
        Applymap('MapaImportePedidos',id_pedido)*
((precio*cantidad)/(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,7
5,76),0,iva)/100)))
-
        (Applymap('MapaRegalos',id_pedido,0) /
(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,75,76),0,iva)/100))) /
Applymap('MapaImportePedidos',id_pedido)*
((precio*cantidad)/(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,7
5,76),0,iva)/100)))
+
        (Applymap('MapaSuplementos',id_pedido,0)/(1+(if(match(Applymap('MapaProvi
ncia',id_pedido,0),73,74,75,76),0,iva)/100)))/
Applymap('MapaImportePedidos',id_pedido)*
((precio*cantidad)/(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,7
5,76),0,iva)/100))))*Applymap('MapaRatio',id_pedido,1)
        AS [Venta S/IVA],

```

$$\frac{((\text{Applymap}('MapaRegalos',id_pedido,0))/(1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}{\text{Applymap}('MapaImportePedidos',id_pedido)*((\text{precio}*cantidad)/(1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}) * \text{Applymap}('MapaRatio',id_pedido,1)} \text{ AS Regalo,}$$

$$\frac{((\text{Applymap}('MapaGastosEnvio',id_pedido,0))/(1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}{\text{Applymap}('MapaImportePedidos',id_pedido)*((\text{precio}*cantidad)/(1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}) * \text{Applymap}('MapaRatio',id_pedido,1)} \text{ AS [Gastos Envio],}$$

$$\frac{((\text{Applymap}('MapaCostesEnvio',id_pedido,0))/(1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}{\text{Applymap}('MapaImportePedidos',id_pedido)*((\text{precio}*cantidad)/(1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}) * \text{Applymap}('MapaRatio',id_pedido,1)} \text{ AS [Costes Envio],}$$

$$\frac{((\text{Applymap}('MapaSuplementos',id_pedido,0))/(1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}{\text{Applymap}('MapaImportePedidos',id_pedido)*((\text{precio}*cantidad)/(1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}) * \text{Applymap}('MapaRatio',id_pedido,1)} \text{ AS [Suplementos S/IVA],}$$

$$\text{coste}*cantidad \text{ AS [Coste S/IVA],}$$

$$\text{precio}*cantidad * \text{Applymap}('MapaRatio',id_pedido,1) \text{ as [Venta producto C/IVA],}$$

$$\frac{((\text{precio}*cantidad}{1+(\text{if}(\text{match}(\text{Applymap}('MapaProvincia',id_pedido,0),73,74,75,76),0,\text{iva})/100)))}{+}$$

```

        (Applymap('MapaGastosEnvio',id_pedido,0) /
(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,75,76),0,iva)/100)))
/
        Applymap('MapaImportePedidos',id_pedido)*
((precio*cantidad)/(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,7
5,76),0,iva)/100)))
-
        (Applymap('MapaRegalos',id_pedido,0) /
(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,75,76),0,iva)/100))) /
Applymap('MapaImportePedidos',id_pedido)*
((precio*cantidad)/(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,7
5,76),0,iva)/100)))
+

(Applymap('MapaSuplementos',id_pedido,0)/(1+(if(match(Applymap('MapaProvi
ncia',id_pedido,0),73,74,75,76),0,iva)/100)))/
Applymap('MapaImportePedidos',id_pedido)*
((precio*cantidad)/(1+(if(match(Applymap('MapaProvincia',id_pedido,0),73,74,7
5,76),0,iva)/100)))
) * Applymap('MapaRatio',id_pedido,1)
-
coste*cantidad as [Margen S/IVA],

precio*cantidad*Applymap('MapaRatio',id_pedido,1)/(1+(if(match(Applymap('Ma
paProvincia',id_pedido,0),73,74,75,76),0,iva)/100)) as [Venta Producto S/IVA],
if(estados=0,'Abierto',
if(estados=1,'Anulado. Falta Producto Proveedor',
if(estados=2,'Anulado falta producto MTBKS',
if(estados=3,'Anulado Cliente','Sin Estado')))) AS [Estado
Línea],
if(estados=0,'Abierto',
if(estados=1,'Anulado. Falta Producto Proveedor',
if(estados=2,'Anulado falta producto MTBKS',
if(estados=3,'Anulado Cliente','Sin Estado')))) AS [Estado
Línea Truco],

```



```

referencia AS [Referencia Mtbks],
Applymap('MapaTallas', `id_valor`, 'Sin talla') AS Talla,
cantidad,
precio*Applymap('MapaRatio',id_pedido,1) as precio,
partner,
Applymap('MapaProveedores',partner,'Sin Proveedor') AS Proveedor,
Applymap('MapaResponsable',partner,'Sin Proveedor') AS Responsable,
`ref_partner` AS [Ref. Partner],
if(entregado=1,'Sí','No') AS Entregado,

Applymap('MapaProveedores', `partner_suministra`,`Sin Proveedor') AS
[Proveedor Suministra],
if (len(`id_combinado`)=0, 'No', 'Sí') AS [Contiene Combo],
1 as [Conteo Pedido Linea],
'Motobuykers' as Tienda,
ApplyMap('MapaProductosNombreParaLineas',id_producto) as
Netrivals_Nombre_Producto
from `${vRutaQVD}Dashboard\pedidos_lineas.qvd' (qvd)
where Applymap('MapaEstado',id_pedido)>0
;

```

```

MapaNumeroLineas:
MAPPING LOAD
PK_Pedido,
count(PK_Pedido)
resident Lineas where TipoLinea='Líneas'
group by PK_Pedido
;

```

```

MapaNumeroLineasConCampaña:

```

```
MAPPING LOAD
PK_Pedido,
count(PK_Pedido)
resident Lineas where TipoLinea='Líneas' and len(CampañaTemp)>0
group by PK_Pedido
;
```

MapaNombreEmail:

```
MAPPING LOAD
name_mailing,
name_mailing
FROM '$(vRutaQVD)Dashboard\Mailing.qvd' (qvd);
```

MapaPedidosDatosExtra:

```
MAPPING LOAD
id_pedido,
if(is_paypal_express=1,'Paypal Express','PayPal Normal')
from '$(vRutaQVD)Dashboard\PedidosDatosExtra.qvd' (qvd);
```

MapaPedidosEstadisticasPlataforma:

```
Mapping Load
id_pedido,
if(tipo_plataforma=1,'Desktop',if(tipo_plataforma=2,'Mobile',if(tipo_plataforma=3,
'Responsive','No disponible')))
from '$(vRutaQVD)Dashboard\PedidosEstadisticas.qvd' (qvd);
```

MapaPedidosEstadisticasDispositivo:

```
Mapping Load
id_pedido,
if(tipo_dispositivo=1,'Android',if(tipo_dispositivo=2,'iOs','No disponible'))
from '$(vRutaQVD)Dashboard\PedidosEstadisticas.qvd' (qvd);
```

Hechos:

LOAD

```
if(Applymap('MapaNumeroLineas',
id,0)=Applymap('MapaNumeroLineasConCampaña',id,0),'Outlet',
    if(Applymap('MapaNumeroLineasConCampaña',id,0)=0,'Store', 'Mixta'))
AS [Store/Outlet],
    'Pedido' AS TIPO,
    id AS PK_Pedido,
    id AS [Nº Pedido],
    `id_usuario` AS PK_Usuario,
    id_usuario AS Usuario_Truco1,
    id_usuario AS Usuario_Truco2,
    1 AS [Conteo Pedidos],
    `id_usuario` AS Usuario,
    cp,
    `id_provincia` AS PK_Provincia,
    `id_pais` AS País,

    `id_pais` & '-' & `cp` as PK_Coordenadas_Codigo_Postal,

    if( wildmatch(id_pais,'PT')>0,'ES',
        if( wildmatch(id_pais,'BE')>0,'FR',
            if( wildmatch(id_pais, 'IT','FR','ES','GB')>0,
                id_pais, 'COM')))) AS Zona,

    date(floor(fecha)) AS Fecha,
    hour(fecha) AS Hora,
    time(fecha) as HoraMinuto,
    datE(fecha) as Fecha_Larga,
    /^fecha_entrega`,
    //subtotal,
    if (estado=0,'Esperando Pago', if (estado=1, 'Abierto', if (estado=2, 'Cerrado',
    if (estado=3, 'Anulado','Sin Estado')))) AS [Estado Pedido],
```

```

if (estado=0,'Esperando Pago', if (estado=1, 'Abierto', if (estado=2, 'Cerrado',
if (estado=3, 'Anulado','Sin Estado')))) AS [Estado Pedido Set],
`ratio_moneda`,
if (`modo_pago`=1,'TPV',
    if (modo_pago=2,
        'Paypal',
        if (modo_pago=3,
            'Reembolso',
            if(modo_pago=4,
                'Paga+t',
                'Sin Modo Pago')))) AS [Modo Pago],
if(`modo_pago`=2,ApplyMap('MapaPedidosDatosExtra',id,'Paypal
Express'),null()) as [Modo Pago Paypal],
if(`modo_pago`=1,ApplyMap('MapaTipologiaTPV',id,'TPV'),null()) as [Modo
Pago TPV],

```

```

ApplyMap('MapaPedidosEstadisticasPlataforma',id,'No Disponible') as
[Plataforma Venta],
ApplyMap('MapaPedidosEstadisticasDispositivo',id,'No Disponible') as
[Dispositivo Venta],
suplementos as [Suplementos pedido],
origen AS Origen,
ApplyMap('MapaNombreEmail',origen,null()) AS name_mailing,
if(`is_app`=1,'APP', 'No APP') AS [Es APP],
if(`tipo_envio`=1,'Envío a domicilio',if(`tipo_envio`=2,'Punto Pack','Sin Tipo
Envío')) AS [Tipo Envío],
ApplyMap('Analytics_Transacciones_Device',id,'No disponible') as
deviceCategory,
ApplyMap('Analytics_Transacciones_channelGrouping',id,'No disponible') as
channelGrouping,
ApplyMap('Analytics_Transacciones_Source',id,'No disponible') as source,

```

```

/*if(lower(ApplyMap('Analytics_Transacciones_Source',id,'No
disponible'))='criteo','criteo',if(lower(ApplyMap('Analytics_Transacciones_Sourc
e',id,'No disponible'))='webgains','webgains')) as source_retargeting,*/
if(wildmatch(lower(ApplyMap('Analytics_Transacciones_Source',id,'No
disponible')), '*criteo*')>0,'criteo') as source_retargeting,
ApplyMap('Analytics_Transacciones_Medium',id,'No disponible') as medium,
ApplyMap('Analytics_Transacciones_Campaign',id,null()) as
Adwords_CampaignID,
ApplyMap('Analytics_Transacciones_Revenue',id,null()) as
Analytics_Revenue,
ApplyMap('Analytics_Transacciones_Shipping',id,null()) as
Analytics_Shipping,
is_notificado,
`total` as pedido_total
from '$(vRutaQVD)Dashboard\pedidos.qvd' (qvd) where estado>0;

```

MapaPaisPedido:

```

Mapping LOAD id,
`id_pais` from '$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);

```

MapaSCISugerencia1:

```

Mapping load ID,
`Sugerencia.1`
from '$(vRutaQVD)Dashboard\SCI.qvd' (qvd);

```

MapaSCISugerencia2:

```

Mapping load ID,
`Sugerencia.2`
from '$(vRutaQVD)Dashboard\SCI.qvd' (qvd);

```

MapaSCISugerencia3:

```

Mapping load ID,
`Sugerencia.3`
from '$(vRutaQVD)Dashboard\SCI.qvd' (qvd);

```

Usuarios:

```
LOAD id AS PK_Usuario,
      id as [Usuario Truco Usuario 1],
      1 AS [Conteo Usuarios],
      email,
      date(`created_at`) AS [Fec. Creación],
      year(`created_at`) AS [Año. Creación],
      num(month(`created_at`)) AS [Mes. Creación],
      day(`created_at`) AS [Día. Creación],
      date(floor(monthstart(created_at))) AS [Fec. Creación Trunca],
      today()-date(created_at) as Dias_Creacion,
      afiliado,
      pais AS [País Usuario],
      date(last_login_date) AS last_login,
      year(last_login_date) & '/' & if( len(num(month(last_login_date)))=1, '0'," &
num(month(last_login_date)) AS [Fecha Login],
      if(estados=1 and acepta_news=1, 'Diaria',
        if(estados=1 and acepta_news=2,'Semanal',
          if(estados=1,'Alta',
            if(estados=0,'Baja',
              if(estados=4,'Invitado',
                'Revisar'))))
      AS [Acepta News],
      if(`tiene_app`=1,'Sí','No') as [Tiene App],
      if(estados=1 and (acepta_news=1 or acepta_news=2),'Receptivo','No
receptivo') as [Situación Usuario],
      acepta_news as num_acepta_news,
      if(today()-date(OpenDate)<30 or today()-date(ClickDate)<30,'<30',
        if(today()-date(OpenDate)<60 or today()-date(ClickDate)<60,'<60',
          if(today()-date(OpenDate)<100 or today()-date(ClickDate)<100,'<100',
            if(today()-date(OpenDate)<180 or today()-date(ClickDate)<180,'<180','No
Activo')))) as [Actividad Usuario],
```

```

    if(today()-date(OpenDate)<30 or today()-date(ClickDate)<30 or today()-
date(last_login_date)<30,'<30',
        if(today()-date(OpenDate)<60 or today()-date(ClickDate)<60 or today()-
date(last_login_date)<60,'<60',
            if(today()-date(OpenDate)<100 or today()-date(ClickDate)<100 or
today()-date(last_login_date)<100,'<100',
                if(today()-date(OpenDate)<180 or today()-date(ClickDate)<180 or
today()-date(last_login_date)<180,'<180','No Activo')))) as [Actividad Usuario
Total],
    date(floor>LastSentDate)) as LastSentDate,
    date(floor>LastOpenDate)) as LastOpenDate,
    date(floor>LastClikDate)) as LastClikDate,
    Capitalize(mid(estilo_1,3)) as estilo,
    if(sexo=0,'Hombre',if(sexo=1,'Mujer','Error')) as Sexo,
    Age(today(),fecha_nacimiento) as Edad,
    lang,
    date(floor>baja_news_at)) as baja_news_at,
    `preferencia_marca`,
    `preferencia_categoria`,
    ApplyMap('MapaSCISugerencia1',id) as [SCI Sugerencia 1],
    ApplyMap('MapaSCISugerencia2',id) as [SCI Sugerencia 2],
    ApplyMap('MapaSCISugerencia3',id) as [SCI Sugerencia 3],
    Recencia,
    Frecuencia,
    `Money`,
    Calificacion_RFM as RFM,
    `horario_envio`,
    estado as usuario_estado,

if(date(floor>LastOpenDate))<date(last_login_date),date(last_login_date),date(floor(
OpenDate))) as usuario_ultima_interaccion,
    if(estado=4,'Invitado',
        if(comprador=1,'Buyer',

```

```

        if(tiene_moto=1 or len(estilo_1)>0,'Prospect','Lead')) as
Fidelidad_Cliente,
    floor(date(baja_news_at)-date(created_at)) as Vida_Usuario,
    floor(date(baja_at)-date(created_at)) as Vida_Usuario_Baja,
    date(floor(baja_at)) as Fecha_Baja_Sistema,
    today()-date(floor(baja_at)) as Dias_Baja_Sistema,
    today()-date(floor(baja_news_at)) as Dias_Baja_News,
    ApplyMap('MapaDescCalificacion',calificacion,'No disponible') as calificacion,
    if(sensibilidad_cupones=1,'Compra sin cupones/no comprador',
        if(sensibilidad_cupones=2,'Muy baja sensibilidad',
            if(sensibilidad_cupones=3,'Baja sensibilidad',
                if(sensibilidad_cupones=4,'Sensibilidad cupones',
                    if(sensibilidad_cupones=5,'Alta sensibilidad',
                        if(sensibilidad_cupones=6,'Seeker cupones','No disponible')))))))) as
sensibilidad_cupones,
    ApplyMap('MapaTipologiaUsuario',id,'Lead') as Tipologia_Usuario,
    Familia_Afiliado,
    Familia_SubAfiliado
from '$(vRutaQVD)Dashboard\usuarios.qvd' (qvd);

```

usuarios_moto:

LOAD

```

    id_usuario as PK_Usuario,
    id_moto as PK_MOTO,
    if(visible=1,'Sí','No') as Usuario_Moto_Visible,
    1 as Conteo_Usuario_Moto
from '$(vRutaQVD)Dashboard\usuarios_moto.qvd' (qvd);

```

left keep(usuarios_moto)

Motos:

Load

```

    id as PK_MOTO,
    marca as Moto_Marca,

```



```

modelo as Moto_Modelo,
cubicaje as Moto_Cubicaje,
anio as Moto_Anio,
modelo_completo as Moto_Modelo_Completo,
mid(estilo_1,5) as Moto_Estilo
from '$(vRutaQVD)Dashboard\motos.qvd' (qvd);

```

provincias:

Load

```

    id as PK_Provincia,
    nombre as Provincia_Nombre
from '$(vRutaQVD)Dashboard\provincias.qvd' (qvd);

```

concatenate(Hechos)

Load

```

    'TarjetaToken' as TIPO,
    date(floor(created_on)) as Fecha,
    id_usuario as PK_Usuario,
    1 as [Conteo tarjetas tokenizadas]
from '$(vRutaQVD)Dashboard\usuario_tarjetas.qvd' (qvd);

```

/*Tramos el stock del amacen de motobuykers de forma diferente*/

concatenate(Hechos)

LOAD

```

    'StockAlmacen' as TIPO,
    date(floor(Fecha)) as Fecha,
    Almacen_Producto,
    replace(Almacen_Talla,'att_',") as Almacen_talla,
    Almacen_Cantidad,
    Almacen_Proveedor,
    Almacen_Marca,
    Almacen_Categoria,

```

```

Almacen_Ubicacion,
Almacen_Coste,
Almacen_Coste_Total,
Applymap('MapaNombreProducto', `id`, 'Sin Nombre Producto') AS
Almacen_Nombre_Producto
from '$(vRutaQVD)Dashboard\stock_almacen.qvd' (qvd);

```

concatenate (Hechos)

```

LOAD
    'Objetivo' AS TIPO,
    makedate( left(AnoMes,4), right(AnoMes,2),1) AS Fecha,
    Pedidos,
    Revenue,
    'Obj ' & AnoMes AS PK_Pedido
FROM '$(vRutaQVD)Dashboard\Objetivos.qvd'
(qvd);

```

concatenate (Lineas)

```

LOAD
    //'Objetivo' AS TIPO,
    //makedate( left(AnoMes,4), right(AnoMes,2),1) AS Fecha,
    //Pedidos,
    Revenue AS [Importe Objetivo] ,
    'Obj ' & AnoMes AS PK_Pedido
FROM '$(vRutaQVD)Dashboard\Objetivos.qvd'
(qvd);

```

concatenate(Hechos)

```

LOAD
    'ObjetivoGeneral' as TIPO,
    date(floor(Fecha)) as Fecha,
    Zona as Zona,

```

```

Members as Objetivo_Members,
`% buyers vs. mebers` as [Objetivo_Buyers vs Members],
`Total Visits` as [Objetivo_Visits],
`% conversion` as Objetivo_Conversion,
Investment as Objetivo_Investment,
`Sales vs. Investment` as [Objetivo_Sales vs Investment],
`Order. X members` as [Objetivo_Order x Member],
`Price x order` as [Objetivo_Price x Order],
`Price x shipm.` as [Objetivo_Price x Ship],
`Buyers` as [Objetivo_Buyers],
`Orders` as [Objetivo_Orders],
`Total sales` as [Objetivo_Sales],
`Returns` as [Objetivo>Returns],
`% Returns` as [Objetivo_%_Returns],
`NET SALES` as [Objetivo_Net_Sales],
    `Sales new registers` as [Objetivo_Sales_New_Registers],
`% New registers` as [Objetivo_%_New_Registers],
`Sales old registers` as [Objetivo_Sales_Old_Registers],
`% old registers` as [Objetivo_%_Old_Registers],
`Sales Outlet` as [Objetivo_Sales_Outlet],
`% Outlet` as [Objetivo_%_Outlet],
`Sales store` as [Objetivo_Sales_Store],
`% store` as [Objetivo_%_Store]
FROM '$(vRutaQVD)Dashboard\objetivo_empresa.qvd' (qvd);

```

concatenate(Hechos)

LOAD

```

    'ObjetivoCanal' as TIPO,
date(floor(Fecha)) as Fecha,
channelGrouping,
Zona,
Sesiones as Objetivo_Canal_Sesiones,
Buyers as Objetivo_Canal_Buyers,

```

```

Revenue as Objetivo_Canal_Revenue,
Investment as Objetivo_Canal_Investment
FROM '$(vRutaQVD)Dashboard\objetivo_empresa_canal.qvd' (qvd);

/*Inversion*/
Concatenate(Hechos)
LOAD
    'Inversion' as TIPO,
    date(floor(Fecha)) as Fecha,
    Investment as Inversion_Inversion,
    Zona as Zona,
    if(lower(Afiliado)='criteo' or
lower(Afiliado)='webgains',capitalize(Afiliado),Afiliado) as Inversion_Afiliado,
    if(lower(Afiliado)='criteo' or lower(Afiliado)='webgains',Capitalize(Afiliado)) as
Retargeting_Partner,
    if(lower(Afiliado)='criteo','criteo',if(lower(Afiliado)='webgains','webgains')) as
source_retargeting,
    Familia_Afiliado as Inversion_Familia,

if(channelGrouping='Email',channelGrouping,ApplyMap('MapaDescInversionCa
nal',Afiliado & '|' & Familia_Afiliado,'Sin Canal')) as channelGrouping
FROM [lib://Sense_Dashboard
(vps229099_administrator)/historico_inversion_qlik.xlsx]
(ooxml, embedded labels, table is Historico_Inversion);

```

```

concatenate (Hechos)
LOAD
    'Abono' AS TIPO,
    `id_pedido` AS PK_Pedido,
    `id_pedido` AS [Nº Pedido],
    1 as [Conteo Abono],

```

```

if(Applymap('MapaNumeroLineas',
id,0)=Applymap('MapaNumeroLineasConCampaña',id,0),'Outlet',
    if(Applymap('MapaNumeroLineasConCampaña',id,0)=0,'Store', 'Mixta'))
AS [Store/Outlet],

```

```

Applymap('MapaPaisPedido',id_pedido) AS País,

```

```

if( wildmatch(Applymap('MapaPaisPedido',id_pedido),'PT')>0,'ES',
    if( wildmatch(Applymap('MapaPaisPedido',id_pedido),'BE')>0,'FR',
        if(
            wildmatch(Applymap('MapaPaisPedido',id_pedido),
'IT','FR','ES','GB')>0,
            Applymap('MapaPaisPedido',id_pedido), 'COM')))) AS Zona,

```

```

if (Applymap('MapaEstado',id_pedido)=0,'Esperando Pago', if
(Applymap('MapaEstado',id_pedido)=1, 'Abierto', if
(Applymap('MapaEstado',id_pedido)=2, 'Cerrado',
    if (Applymap('MapaEstado',id_pedido)=3, 'Anulado','Sin Estado')))) AS
[Estado Pedido],

```

```

date(floor(fecha)) AS Fecha,

```

```

if(WildMatch(origen,1)>0,'Devolución.',
    if(WildMatch(origen,2)>0,'Anulación.',
        if(WildMatch(origen,3)>0,'Diferencia precios.',
            if(WildMatch(origen,4)>0,'Reembolso portes.',
                if(WildMatch(origen,5)>0,'Devolución y reembolso.',
                    if(WildMatch(origen,6)>0,'Devolución cambio talla/color no
disponible.',
                        if(WildMatch(origen,7)>0,'Devolución por error no
disponible (proveedor).',
                            if(WildMatch(origen,8)>0,'Devolucoón por error no
disponible (MTBKS).',

```

```

        if(WildMatch(origen,9)>0,'Anulación falta stock error proveedor.',
            if(WildMatch(origen,10)>0,'Anulación falta stock error
MTBKS.',
                if(WildMatch(origen,11)>0,'Anulación cliente.',
                    if(WildMatch(origen,12)>0,'Otros.',
                        if(WildMatch(origen,13)>0,'Fraude.',
                            'Error.')))))))))) as [Tipo Abono],

importe*Applymap('MapaRatio',id_pedido,1) AS [Importe Abono],
/*date(floor(Applymap('MapaFechaPedido',id_pedido))) as [Abono Fecha
Pedido],

    MakeDate(year(date(floor(fecha))),month(date(floor(fecha))),1) as [Abono
Mes Aplica]*/

if(date(floor(Applymap('MapaFechaPedido',id_pedido)))<MakeDate(year(date(fl
oor(fecha))),month(date(floor(fecha))),1),'Pedido Meses Anteriores','Pedido Mes
Actual') as [Abono Mes Aplica],
    ApplyMap('MapaPedidoUsuario',id_pedido,'No disponible') as PK_Usuario

from '$(vRutaQVD)Dashboard\abonos.qvd' (qvd);

//Analytics sobre qlik
concatenate(Hechos)
LOAD
    'Analytics' as TIPO,
    date(fecha) as Fecha,
    if (id_zona=1,'ES', IF(id_zona=2,'IT', IF(id_zona=3,'FR',
        IF(id_zona=4,'GB', IF(id_zona=5,'COM')))) AS Zona,
    deviceCategory,
    channelGrouping,
    source,

```



```

unique_link_clickthroughs,
total_conversion,
unique_conversion,
click2open_rate,
revenue,
pedidos,
if( len("Outlet/Store")=0,'Mixta', "Outlet/Store") AS [Store/Outlet],
eMailType,
Pais AS Zona
FROM '$(vRutaQVD)Dashboard\Mailing.qvd' (qvd);

```

DescripcionFechasMailing:

```

LOAD
name_mailing,
date(floor(send_date_mailing)) AS [Envío Fecha],
year(date(floor(send_date_mailing))) AS [Envío Año],
month(date(floor(send_date_mailing))) AS [Envío Mes],
day(date(floor(send_date_mailing))) AS [Envío Día]
FROM '$(vRutaQVD)Dashboard\Mailing.qvd' (qvd);

```

MapaEstadoRegalo:

```

MAPPING LOAD
id,
activo
FROM '$(vRutaQVD)Dashboard\regalos.qvd'
(qvd);

```

pedidos_regalos:

```

LOAD
id_pedido as PK_Pedido,
id_regalo AS PK_Regalo,
consumido as aux_regalo_consumido,
if (Applymap('MapaEstadoRegalo',id_regalo)=0 and consumido=1, 'Sí', 'No')
AS Consumido,

```



```
1+(ApplyMap('MapaZonasIvaGeneral',ApplyMap('MapaZonas',id_pedido))/100)
as Regalo_Iva_General,
    ApplyMap('MapaRatio',id_pedido,1) as Regalo_Ratio
FROM '$(vRutaQVD)Dashboard\pedidos_regalos.qvd'
(qvd);
```

MapaRegaloClase:

MAPPING LOAD

```
    id,
    nombre
FROM '$(vRutaQVD)Dashboard\regalos_clases.qvd'
(qvd);
```

MapaRegaloMotivo:

MAPPING LOAD

```
    id & '-' & id_regalo_clase,
    nombre
FROM '$(vRutaQVD)Dashboard\regalos_motivos.qvd'
(qvd);
```

MapaRegaloMotivoSingle:

MAPPING LOAD

```
    id,
    nombre
FROM '$(vRutaQVD)Dashboard\regalos_motivos.qvd'
(qvd);
```

Regalos:

LOAD

```
    /*1 AS [Conteo Regalos],
    id AS PK_Regalo,
```

```

    Applymap('MapaRegaloClase',id_clase, 'Sin Clase ' &id_clase ) AS [Regalo
Clase],
    Applymap('MapaRegaloMotivo', id_motivo & '-' & id_clase, 'Sin motivo ' &
id_motivo & '-' & id_clase ) AS [Regalo Motivo],
    id_promocion AS PK_Promocion,
    mensaje AS [Regalo Mensaje],
    date(floor(caducidad)) as [Regalo Caducidad],
    id_usuario,
    importe_minimo as [Regalo Importe Minimo],
    importe_regalado AS [Regalo Importe Regalado],
    if(aplicable_en=0,'Importe pedido','Gastos Envío') as [Regalo Aplicable En],
    if(tipo=1,'Importe fijo','Importe %') as [Regalo Tipo Regalo],
    date(floor(fecha_creacion)) as [Regalo Fecha Creacion]*/

```

```

1 AS [Conteo Regalos],
id AS PK_Regalo,
    Applymap('MapaRegaloClase',id_clase, 'Sin Clase ' &id_clase ) AS
[Promociones Clase],
    Applymap('MapaRegaloMotivo', id_motivo & '-' & id_clase, 'Sin motivo ' &
id_motivo & '-' & id_clase ) AS [Promociones Clase-Motivo],
    id_promocion AS PK_Promocion,
    mensaje AS [Promociones Mensaje],
    if(tipo=0,'Importe fijo','Importe %') as [Promociones Tipo Regalo],
    if(tipo=0,importe,importe & '%') as [Promociones Importe Otorgado],
    date(floor(caducidad)) as [Promociones Caducidad],
    importe_minimo as [Promociones Importe Minimo],
    Applymap('MapaNombreCampana',id_campania) as [Promociones
Campaña],
    Applymap('MapaMarcas',id_marca) as [Promociones Marca],
    Applymap('MapaProductosNombreParaLineas',id_producto) as
[Promociones Producto],
    if (`id_modo_pago`=1,'TPV',

```

```

        if (id_modo_pago=2,
        'Paypal',
        if (id_modo_pago=3,
        'Reembolso',
        if(id_modo_pago=4,
        'Paga+t',
        'Sin Modo Pago')))) AS [Promociones Modo Pago],
id_categoria as [Promociones Categoria Producto],
Applymap('MapaProveedores',id_proveedor) as [Promociones Proveedor],
activo as aux_promociones_activo,
if(aplicable_en=0,'Importe pedido','Gastos Envío') as [Promociones Aplicable
En],
date(floor(fecha_creacion)) as [Promociones Fecha Creacion],
anulado as aux_promociones_anulado,
if(restriccion_dispositivo=0,'Disponible para todos','Solo para app') as
[Promociones Restriccion Dispositivo],
importe_regalado AS [Promociones Importe Regalado],
ApplyMap('MapaRegaloMotivoSingle',id_motivo) as [Promociones Motivo]

FROM '$(vRutaQVD)Dashboard\regalos.qvd'
(qvd);

```

actividades_promocionales:

LOAD

```

/*id AS PK_Promocion,
codigo AS [AP Regalo Código],
date(floor(fecha_inicio)) AS [AP Fecha Regalo Inicio],
date(floor(fecha_fin)) AS [AP Fecha Regalo Fin],
floor(fecha_fin-fecha_inicio) as [AP Duracion],
if(tipo=0,'Aplicable en Registro','Aplicable en Cesta') as [AP Tipo Actividad
Promocional],
if(is_cupon=0,'No','Sí') as [AP Es Cupon],
if(posteriori=0,'Inmediatamente','A futuro') as [AP Posteriori],
ApplyMap('MapaMarcas',id_marca) AS [AP Marca],

```

```

    Applymap('MapaNombreProducto',`id_producto`,`Sin Nombre Producto') AS
[AP Producto],
    if (`id_modo_pago`=1,'TPV',
        if (id_modo_pago=2,
            'Paypal',
            if (id_modo_pago=3,
                'Reembolso',
                if(id_modo_pago=4,
                    'Paga+t',
                    'Sin Modo Pago')))) AS [AP Modo Pago],
    ApplyMap('MapaProveedores',id_proveedor,'Sin Nombre Proveedor') as [AP
Proveedor],
    if(restriccion_dispositivo=0,'Disponible para todos','Solo para app') as [AP
Restriccion Dispositivo],
    if(aplicable_en=0,'Importe pedido','Gastos Envío') as [AP Aplicable En],
    if(tipo_regalo=0,'Importe Fijo','Importe %') as [AP Tipo Regalo],
    importe as [AP Importe],
    id_campania as [AP id_campania],
    mensaje as [AP Mensaje],
    importe_minimo as [AP Importe Minimo],
    ApplyMap('MapaRegaloMotivoSingle',id_motivo) as [AP Motivo]*/

id as PK_Promocion,
if(tipo=0,'Aplicable en Registro','Aplicable en Cesta') as [Promociones Tipo
AP],
    codigo AS [PromocionesCodigo],
    if(is_cupon=0,'No','Sí') as [Promociones Es Cupon],
    date(floor(fecha_inicio)) AS [Promociones Fecha Inicio AP],
    date(floor(fecha_fin)) AS [Promociones Fecha Fin AP],
    if (id_zona=1,'ES', IF(id_zona=2,'IT', IF(id_zona=3,'FR',
        IF(id_zona=4,'GB', IF(id_zona=5,'COM')))) AS [Promociones Zona],
    if(posteriori=0,'Inmediatamente','A futuro') as [Promociones Posteriori]

```

```
FROM '$(vRutaQVD)Dashboard\actividades_promocionales.qvd'  
(qvd);
```

```
/*Popup*/
```

```
concatenate(Hechos)
```

```
Load
```

```
    date(floor(fecha)) as Fecha,  
    id_usuario as PK_Usuario,  
    'Popup' as TIPO,  
    1 as Conteo,  
    if(tipo_evento=1,'Apertura popup puntos',  
        if(tipo_evento=2,'Cierre popup puntos',  
            if(tipo_evento=3,'Apertura popup registro',  
                if(tipo_evento=4,'Cierre popup registro')))) as [Popup Tipo Evento],  
    if(tipo_dispositivo=1,'Desktop',  
        if(tipo_dispositivo=2,'Mobile',  
            if(tipo_dispositivo=3,'Tablet','No disponible')) as deviceCategory,  
    if(id_zona=1,'ES',  
        if(id_zona=2,'IT',  
            if(id_zona=3,'FR',  
                if(id_zona=4,'GB',  
                    if(id_zona=5,'COM','Error')))) as Zona,  
    if(tipo_plataforma=1,'Android','iOS') as [Popup Tipo Plataforma]  
FROM '$(vRutaQVD)Dashboard\popup_reincidente.qvd' (qvd);
```

```
concatenate (Hechos)
```

```
LOAD
```

```
    id AS PK_Usuario,  
    date(floor(`created_at`)) AS Fecha,
```

```

'Usuario' AS TIPO,
1 AS [Conteo Usuarios Hechos],
if (zona=1,'ES', IF(zona=2,'IT', IF(zona=3,'FR',
      IF(zona=4,'GB', IF(zona=5,'COM'))))) AS Zona
from '$(vRutaQVD)Dashboard\usuarios.qvd' (qvd);

```

/*----Para controlar las fechas de bajas-----*/

concatenate (Hechos)

LOAD

```

id AS PK_Usuario,
date(floor(baja_at)) AS Fecha,
'Usuario-Baja' AS TIPO,
1 AS [Conteo Usuarios Hechos Baja],
if (zona=1,'ES', IF(zona=2,'IT', IF(zona=3,'FR',
      IF(zona=4,'GB', IF(zona=5,'COM'))))) AS Zona
from '$(vRutaQVD)Dashboard\usuarios.qvd' (qvd)
where len(baja_at)<>0;

```

usuario_tarjetas_dedu:

LOAD

```

id_usuario as PK_Usuario,
1 as [Conteo Usuarios Tokenizados],
1 as Usuario_Con_Tarjeta_Tokenizada
from '$(vRutaQVD)Dashboard\usuario_tarjetas_dedu.qvd' (qvd);

```

lead_scoring_marcas:

LOAD

```

id_usuario as PK_Usuario,
estilo as Lead_Scoring_Marca,

```

```
scoring as Lead_Scoring_Marca_Scoring
from '$(vRutaQVD)Dashboard\lead_scoring_marcas.qvd' (qvd);
```

lead_scoring_categorias:

LOAD

```
    id_usuario as PK_Usuario,
    estilo as Lead_Scoring_Categoria,
    scoring as Lead_Scoring_Categoria_Scoring
from '$(vRutaQVD)Dashboard\lead_scoring_categorias.qvd' (qvd);
```

MapaPedidoMetodoPago:

Mapping Load

```
id,
modo_pago
from '$(vRutaQVD)Dashboard\Pedidos.qvd' (qvd);
```

MapaPedidoZona:

Mapping Load

```
id,
`id_pais`
from '$(vRutaQVD)Dashboard\Pedidos.qvd' (qvd);
```

concatenate(Hechos)

facturas:

load

```
    'Factura' as TIPO,
    date(floor(created_at)) as Fecha,
    id_pedido as PK_Pedido,
    id as PK_Factura,
    subtotal*ratio as factura_subtotal,
    if (ApplyMap('MapaPedidoMetodoPago',id_pedido)=1,'TPV',if
(ApplyMap('MapaPedidoMetodoPago',id_pedido), 'Paypal', if
(ApplyMap('MapaPedidoMetodoPago',id_pedido), 'Reembolso',
'Sin Modo Pago')))) AS [Modo Pago],
```

```

if( wildmatch(ApplyMap('MapaPedidoZona',id_pedido),'PT')>0,'ES',
    if( wildmatch(ApplyMap('MapaPedidoZona',id_pedido),'BE')>0,'FR',
        if(
            wildmatch(ApplyMap('MapaPedidoZona',id_pedido),
                'IT','FR','ES','GB')>0,
            ApplyMap('MapaPedidoZona',id_pedido),      'COM')))) AS Zona

```

```

from '$(vRutaQVD)Dashboard\facturas.qvd' (qvd);

```

```

concatenate(Hechos)

```

```

load

```

```

    'Devolucion' as TIPO,
    date(floor(fecha_alta)) as Fecha,
    id_pedido as PK_Pedido,
    id as PK_Devolucion,
    if(tipo=1,'Pedido erroneo o defectuoso',
        if(tipo=2,'Pedido correcto cambio de talla o color',
            if(tipo=3,'Pedido correcto devolución o reembolso',
                'Error')))) as Devoluciones_Tipo,
    if(error=0,'No aplica',
        if(error=1,'Motobuykers',
            if(error=2,'Proveedor',
                'Error')))) as Devoluciones_Error,
    if(cambiapor=0,'Pendiente',
        if(cambiapor=1,'Devolución tipo pedido erroneo o defectuoso',
            if(cambiapor=2,'Devolución pedido correcto cambio talla o color',
                if(cambiapor=3,'Devolución tipo pedido correcto devolución o
reembolso',
                    if(cambiapor=4,'Tramitar garantia',
                        'Pendiente')))) as Devoluciones_Cambiapor,
        if(estado=0,'Cerrada','Abierta') as Devoluciones_Estado,
    date(floor(fecha_reembolso)) as Devoluciones_Fecha_Reembolso,
    date(floor(fecha_cierre)) as Devoluciones_Fecha_Cierre,

```



```

if          (ApplyMap('MapaPedidoMetodoPago',id_pedido)=1,'TPV',if
(ApplyMap('MapaPedidoMetodoPago',id_pedido),          'Paypal',          if
(ApplyMap('MapaPedidoMetodoPago',id_pedido), 'Reembolso',
'Sin Modo Pago')))) AS [Modo Pago],

if( wildmatch(ApplyMap('MapaPedidoZona',id_pedido),'PT')>0,'ES',
if( wildmatch(ApplyMap('MapaPedidoZona',id_pedido),'BE')>0,'FR',
if(          wildmatch(ApplyMap('MapaPedidoZona',id_pedido),
'IT','FR','ES','GB')>0,
ApplyMap('MapaPedidoZona',id_pedido),          'COM')))) AS Zona,
coste as Devoluciones_Coste,
1 AS [Conteo Devoluciones]

from '$(vRutaQVD)Dashboard\devoluciones.qvd' (qvd);

```

MapaPedidoLineaCoste:

Mapping Load

[id Linea Pedido],

[Coste S/IVA]/cantidad

resident Lineas;

MapaPedidoLineaPrecio:

Mapping Load

[id Linea Pedido],

[Venta Producto S/IVA]/cantidad

resident Lineas;

MapaPedidoLineaProveedor:

Mapping LOAD

[id Linea Pedido],

Proveedor

resident Lineas;

```

devoluciones_lineas:
load
    id_devolucion as PK_Devolucion,
    cantidad as devoluciones_lineas_cantidad,
    if(estado=0,'Pendiente registro',if(estado=1,'Registrada','Error')) as
Devoluciones_lineas_estado,
    ApplyMap('MapaPedidoLineaCoste',id_pedido_linea) as
Devoluciones_lineas_coste,
    ApplyMap('MapaPedidoLineaPrecio',id_pedido_linea) as
Devoluciones_lineas_precio,
    ApplyMap('MapaPedidoLineaProveedor',id_pedido_linea) as
Devoluciones_proveedor
from '$(vRutaQVD)Dashboard\devoluciones_lineas.qvd' (qvd);

```

/*Para generar el feed de productos*/

```
concatenate(Hechos)
```

```
LOAD
```

```

    'AnalyticsFeedProducto' as TIPO,
    date(floor(fecha)) as Fecha,
    if (id_zona=1,'ES', IF(id_zona=2,'IT', IF(id_zona=3,'FR',
        IF(id_zona=4,'GB', IF(id_zona=5,'COM'))))) AS Zona,
    deviceCategory,
    channelGrouping,
    source,
    medium,
    pageviews,
    uniquePageviews,
    timeOnPage,
    entrances,

```

```

bounces,
exits,
fecha & '|' & id_producto as PK_Pedido

from '$(vRutaQVD)Dashboard\GA_RowData_Qlik_Inf_Productos.qvd' (qvd);

concatenate(Lineas)
LOAD
    'AnalyticsFeedProducto' as TipoLinea,
    fecha & '|' & id_producto as PK_Pedido,
    id_producto as PK_Producto,

Applymap('MapaProveedores',ApplyMap('MapaProductosPartner',id_producto),'
Sin Proveedor') as Proveedor
from '$(vRutaQVD)Dashboard\GA_RowData_Qlik_Inf_Productos.qvd' (qvd);

concatenate(Hechos)
adwords_hechos:
LOAD
    'AdwordsCampaigns' as TIPO,
    1 as [Conteo Adwords],
    CampaignID as Adwords_CampaignID,
    Clicks as Adwords_Clicks,
    Impressions as Adwords_Impressions,
    1.09*(Cost/1000000) as Adwords_Cost, /*aplicamos el 9% de comision de la
empresa*/
    date(floor(`Day`)) as Fecha,
    if (id_zona=1,'ES', IF(id_zona=2,'IT', IF(id_zona=3,'FR',
        IF(id_zona=4,'GB', IF(id_zona=5,'COM'))))) AS Zona
from '$(vRutaQVD)Dashboard\adwords.qvd' (qvd);

adwords:
load
    CampaignID as Adwords_CampaignID,

```

```
    FirstValue(cuenta) as adwords_cuenta,  
    FirstValue(Campaign) as adwords_campaign  
from '$(vRutaQVD)Dashboard\adwords.qvd' (qvd)  
GROUP BY CampaignID  
;
```

```
concatenate(Hechos)
```

```
LOAD
```

```
    'Netrivals' as TIPO,  
    date(floor(Fecha)) AS Fecha,  
    Fecha & '|' & Code as PK_Pedido
```

```
from '$(vRutaQVD)Dashboard\netrivals_child.qvd' (qvd);
```

```
concatenate(Lineas)
```

```
Load
```

```
    'Netrivals' as TipoLinea,  
    Fecha & '|' & Code as PK_Pedido,  
    Code as PK_Producto,  
    Price as Netrivals_Precio,  
    Name as Tienda,  
    Position as Netrivals_Position,  
    ApplyMap('MapaNombreProducto',Code) as Netrivals_Nombre_Producto  
from '$(vRutaQVD)Dashboard\netrivals_child.qvd' (qvd);
```

```
concatenate(Hechos)
```

```
load
```

```
    'Vozitel' as TIPO,  
    date(floor(FECHA_DIA)) AS Fecha,  
    time(FECHA,'hh:mm') as HoraMinuto,  
    1 as Conteo,  
    ID,  
    DIR,
```

CALLID,
CHANNEL,
CE_EXTENSION,
HANGUP_CAUSE,
CLASIFICACION,
REFERENCIA,
CALLAPIID,
CALLERIDNUM,
WAITTIME,
SLA,
WHU,
CALLERIDNAME,
CALLEDDID,
CALLBACK,
FLUJO,
CALLEDNUMBER,
HANGUP,
DURACION,
FILERECORDED,
`NUM` AS Vozitel_Num,
ESTADO_CLASIFICACION,
Vozitel_Group_Name,
Vozitel_Group_Support,
Vozitel_Dentro_Horario

from '\$(vRutaQVD)Dashboard\historico_vozitel.qvd' (qvd);

Anexo 2 – QVD de Qlik Sense sobre WeRide

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 MonthNames='ene;feb;mar;abr;may;jun;jul;ago;sep;oct;nov;dic';
SET DayNames='lun;mar;mié;jue;vie;sáb;dom';
SET
LongMonthNames='enero;febrero;marzo;abril;mayo;junio;julio;agosto;septiemb
e;octubre;noviembre;diciembre';
SET LongDayNames='lunes;martes;miércoles;jueves;viernes;sábado;domingo';
SET FirstWeekDay=0;
SET BrokenWeeks=1;
SET ReferenceDay=0;
SET FirstMonthOfYear=1;

```

```

Let vGeneraQVD=1;

```

```

Let vRutaQVD='lib://DatosLocal (vps229099_administrator)';

```

```

LIB CONNECT TO 'Produccion (vps229099_administrator)';

```

```

if $(vGeneraQVD)=1 then

```

```

User:

```

```

LOAD *;

```

```

SQL SELECT u.*, if(Picture='456158821e995dd6b93b8be132e9e981.jpg' or
Picture is null or Picture='',0,4)

```

```

+

```

```

if(Username is not null,4,0)

```

```

+

```

```

if(Telephone is not null and trim(Telephone)<>',3,0)

```

```

+

```

```

if((select 'Ok' from weride.UserMoto um where u.ID=um.UserId group by 1) is
not null,3,0)

```

```

+
if(Bio is not null or trim(Bio)<>" ,1,0)
+
if(FirstName is not null and trim(FirstName)<>" and Privacyity &
256>0,2,if(FirstName is not null and trim(FirstName)<>" and Privacyity &
256=0,1,0))
+
if(Gender in (1,2) and Privacyity & 512>0,2,if(Gender in (1,2) and Privacyity &
512=0,1,0))
+
if(date(u.Birthday)>'0000-00-00' and Privacyity &
1024>0,2,if(date(u.Birthday)>'0000-00-00' and Privacyity & 1024=0,1,0))
+
if(CityName is not null and trim(CityName)<>" and Privacyity &
8>0,2,if(CityName is not null and trim(CityName)<>" and Privacyity & 8=0,1,0))
+
if(CountryCode is not null and trim(CountryCode)<>" and Privacyity &
32>0,2,if(CountryCode is not null and trim(CountryCode)<>" and Privacyity &
32=0,1,0))
+
if(PostalCode is not null and trim(PostalCode)<>" and Privacyity &
64>0,2,if(PostalCode is not null and trim(PostalCode)<>" and Privacyity &
64=0,1,0))
+
if(Pilot in (0,1) and privacyity & 128>0,2,0)
+
if(Privacyity & 1>0,1,0) #friends
+
if(Privacyity & 2>0,1,0) #groups
+
if(Privacyity & 4>0,1,0) #routes
+
if(Notifications & 1>0,1,0) #solicitud amistad
+

```

```

if(Notifications & 2>0,1,0) #wechat amigos
+
if(Notifications & 4>0,1,0) #wespot grupos
+
if(Notifications & 16>0,1,0) #wespot
+
if(Notifications & 32>0,1,0) #WeForum
as Points
FROM weride.User u;
STORE User INTO '$(vRutaQVD)WeRide\User.qvd' (qvd);
DROP TABLE User;

```

Grupos:

```

LOAD *;
SQL SELECT *
FROM weride.`Group`;
STORE Grupos INTO '$(vRutaQVD)WeRide\Grupos.qvd' (qvd);
DROP TABLE Grupos;

```

Route:

```

LOAD *;
SQL SELECT
  `ID`,
  `Name`,
  `Public`,
  `Created`,
  `Modified`,
  `Deleted`,
  `City`,
  `CategoryID`,
  `UserID`,
  `Picture`,
  `Description`,
  `Distance`,

```



```
`Time`,
`Longitude`,
`Latitude`,
`CountryID`,
`CountryCode`
FROM weride.Route;
STORE Route INTO '$(vRutaQVD)WeRide\Route.qvd' (qvd);
DROP TABLE Route;
```

Subscriptions:

```
LOAD *;
SQL SELECT *
FROM weride.Subscriptions;
STORE Subscriptions INTO '$(vRutaQVD)WeRide\Subscriptions.qvd' (qvd);
DROP TABLE Subscriptions;
```

UserMoto:

```
LOAD *;
SQL SELECT *
FROM weride.UserMoto;
STORE UserMoto INTO '$(vRutaQVD)WeRide\UserMoto.qvd' (qvd);
DROP TABLE UserMoto;
```

Motos:

```
LOAD *;
SQL SELECT *
FROM weride.Motos;
STORE Motos INTO '$(vRutaQVD)WeRide\Motos.qvd' (qvd);
DROP TABLE Motos;
```

City:

```
LOAD *;
SQL SELECT *
FROM weride.City;
```

```
STORE City INTO '$(vRutaQVD)WeRide\City.qvd' (qvd);  
DROP TABLE City;
```

Country:

```
LOAD *;  
SQL SELECT *  
FROM weride.Country;  
STORE Country INTO '$(vRutaQVD)WeRide\Country.qvd' (qvd);  
DROP TABLE Country;
```

Category:

```
LOAD *;  
SQL SELECT *  
FROM weride.Category;  
STORE Category INTO '$(vRutaQVD)WeRide\Category.qvd' (qvd);  
DROP TABLE Category;
```

Province:

```
LOAD *;  
SQL SELECT *  
FROM weride.Province;  
STORE Province INTO '$(vRutaQVD)WeRide\Province.qvd' (qvd);  
DROP TABLE Province;
```

WeSpotUser:

```
LOAD *;  
SQL select  
  ws.UserID,  
  ws.Latitude,  
  ws.Longitude,  
  ws.Active  
from  
(  
select
```

```

UserID,
max(Created) as maximaCreacion
from weride.WeSpotUser
where Latitude<>0 and Longitude<>0
group by 1) as x
inner join weride.WeSpotUser ws on x.UserID = ws.UserID and
x.maximaCreacion=ws.Created;
STORE WeSpotUser INTO '$(vRutaQVD)WeRide\WeSpotUser.qvd' (qvd);
DROP TABLE WeSpotUser;

```

```

GroupUser:
LOAD *;
SQL SELECT *
FROM weride.GroupUser;
STORE GroupUser INTO '$(vRutaQVD)WeRide\GroupUser.qvd' (qvd);
DROP TABLE GroupUser;

```

```

ChatMessage:
LOAD *;
SQL SELECT *
FROM weride.ChatMessage;
STORE ChatMessage INTO '$(vRutaQVD)WeRide\ChatMessage.qvd' (qvd);
DROP TABLE ChatMessage;

```

```

`Comment`:
LOAD *;
SQL SELECT *
FROM weride.`Comment`;
STORE `Comment` INTO '$(vRutaQVD)WeRide\Comment.qvd' (qvd);
DROP TABLE `Comment`;

```

```

File:
LOAD *;
SQL SELECT *

```

```
FROM weride.File;
STORE File INTO '$(vRutaQVD)WeRide\File.qvd' (qvd);
DROP TABLE File;
```

Like:

```
LOAD *;
SQL SELECT *
FROM weride.Like;
STORE Like INTO '$(vRutaQVD)WeRide\Like.qvd' (qvd);
DROP TABLE Like;
```

PushMessagesUsers:

```
LOAD *;
SQL SELECT *
FROM weride.PushMessagesUsers;
STORE          PushMessagesUsers          INTO
'$(vRutaQVD)WeRide\PushMessagesUsers.qvd' (qvd);
DROP TABLE PushMessagesUsers;
```

UserRelation:

```
LOAD *;
SQL SELECT *
FROM weride.UserRelation;
STORE UserRelation INTO '$(vRutaQVD)WeRide\UserRelation.qvd' (qvd);
DROP TABLE UserRelation;
```

WeSpotUser_Abs:

```
LOAD *;
SQL SELECT *
FROM weride.WeSpotUser;
STORE WeSpotUser_Abs INTO '$(vRutaQVD)WeRide\WeSpotUser_Abs.qvd'
(qvd);
DROP TABLE WeSpotUser_Abs;
```

```

GA_RowData_Qlik_WeRide:
LOAD *;
SQL SELECT *
FROM BI_WeRide.GA_RowData_Qlik_WeRide;
STORE          GA_RowData_Qlik_WeRide          INTO
'$(vRutaQVD)WeRide\GA_RowData_Qlik_WeRide.qvd' (qvd);
DROP TABLE GA_RowData_Qlik_WeRide;

```

```

Endif
if (1=1) then

```

```

Category:
LOAD
    ID as PK_Category,
    Name as Category_Name
from '$(vRutaQVD)WeRide\Category.qvd' (qvd);

```

```

City:
LOAD
    ID as PK_City,
    Name as City_Name,
    CountryID as PK_Country
from '$(vRutaQVD)WeRide\City.qvd' (qvd);

```

```

/*Country:
LOAD
    ID as PK_Country,
    Name as Country_Name
from '$(vRutaQVD)WeRide\Country.qvd' (qvd);*/

```

```

Province:
load
    ID as PK_Province,
    Name as Province_Name

```

```
from '$(vRutaQVD)\WeRide\Province.qvd' (qvd);
```

Country:

```
LOAD
```

```
    'General' & '-' & ID as PK_Tipo_Country,
```

```
    ID as PK_Country,
```

```
    Name as Country_Name
```

```
from '$(vRutaQVD)\WeRide\Country.qvd' (qvd);
```

```
concatenate(Country)
```

```
LOAD
```

```
    'Usuario' & '-' & PK_Country as PK_Tipo_Country,
```

```
    //ID as PK_Country
```

```
    Country_Name as Country_Name
```

```
resident Country;
```

MapaUsuariosMTBKS:

```
Mapping LOAD
```

```
    email,
```

```
    'Yes'
```

```
from '$(vRutaQVD)\Dashboard\usuarios.qvd' (qvd);
```

UsuariosMTBKSCompradores:

```
Load
```

```
    email as mtbks_usuarios_email,
```

```
    id as mtbks_usuarios_id
```

```
from '$(vRutaQVD)\Dashboard\usuarios.qvd' (qvd);
```

```
inner join
```

```
Load
```

```
    id_usuario as mtbks_usuarios_id,
```

```
    'Yes'
```

```
from '$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);
```

MapaUsuariosCompradores:

```
mapping Load
```

```
    mtbks_usuarios_email,
```

```
    'Yes'
```

```
resident UsuariosMTBKSCmpradores;
```

MapaUserCountry:

```
Mapping Load
```

```
    ID,
```

```
    CountryID
```

```
from '$(vRutaQVD)WeRide\User.qvd' (qvd);
```

MapaWeSpotUser:

```
LOAD
```

```
    `UserID`,
```

```
    GeoMakePoint(Latitude,Longitude)
```

```
from '$(vRutaQVD)WeRide>WeSpotUser.qvd' (qvd);
```

MapaCountryCode:

```
Mapping LOAD
```

```
    `Code`,
```

```
    `Name`
```

```
from '$(vRutaQVD)WeRide\Country.qvd' (qvd);
```

Hechos:

```
LOAD
```

```
    'Grupo' as TIPO,
```

```
    1 as Conteo,
```

```
    Name,
```

```
    date(floor(Created)) as Fecha,
```

```
    CityID as PK_City,
```

```
    CategoryID as PK_Category,
```

```

//CountryID, will use country field from city table
`UserID` as PK_User,
if(Latitude<>0,GeoMakePoint(Latitude,Longitude)) as GeoGrupo,
if(`Public`=1,'Public','Private') as Group_Privacity,
ID as PK_Group,
Latitude as Group_Latitude,
Longitude as Group_Longitude

/*
`Public`,
Suggested,
`Tag`,
Weight,
Modified,
Deleted,
Picture,
Description,
CountryCode,
Telephone,
Mail,
Web,
Fax,
Address,
FacebookID,
TwitterID,
PublicCategoryID,
InstagramID,
CountryName,
BackgroundImage,
FacebookAccessToken,
FacebookPage,
TwitterUserName*/
from '$(vRutaQVD)WeRide\Grupos.qvd' (qvd);

```



```

//se crea clave sintetica, revisar como hacerlo
/*LOAD
    'Group' & '-' & GroupID as PK_Hechos_Group,
    'User' & '-' & `UserID` as PK_GroupUser_User,
    1 as Conteo_GroupUser
from '$(vRutaQVD)\WeRide\GroupUser.qvd' (qvd);*/

concatenate(Hechos)
LOAD
    1 as Conteo,
    'Ruta' as TIPO,
    ID as PK_Route,
    Name,
    Public,
    date(floor(Created)) as Fecha,
    City as PK_City,
    CategoryID as PK_Category,
    `UserID` as PK_User,
    Distance,
    GeoMakePoint(Latitude,Longitude) as GeoRuta
//CountryID, will use country field from city table
//CountryCode will use country field from city table
/*
    Modified,
    Deleted,
    Picture,
    Description,
    `Time`,
    Longitude,
    Latitude,*/

from '$(vRutaQVD)\WeRide\Route.qvd' (qvd);

```

```

concatenate(Hechos)
LOAD
    1 as Conteo,
    'Suscripcion' as TIPO,
    `UserID` as PK_User,
    if(SubscriptionType=0,'Gift',if(SubscriptionType=0,'3
Months',if(SubscriptionType=0,'1 Year','Error'))) as SubscriptionType,
    if(State=0,'Inactive',if(State=1,'In Process',if(State=2,'Active','Error'))) as
State,
    date(floor(ExpirationDate)),
    TransactionNumber,
    TransactionData,
    date(floor(Created)) as Fecha
    /*Modified*/
from '$(vRutaQVD)\WeRide\Subscriptions.qvd' (qvd);

```

```

concatenate(Hechos)
LOAD
    1 as Conteo,
    ID as PK_ChatMessage,
    'ChatMessage' as TIPO,
    if(MessageType=0,'1:1 wechat/weforum',
        if(MessageType=1,'Text + route wechat/weforum',
            if(MessageType=2,'Text + img wechat/weforum',
                if(MessageType=3,'Wespot act/deac',
                    if(MessageType=4,'Route + img weforum'))))) as MessageType,
    date(floor(Created)) as Fecha
    //ChatID,
    //FromUserID,
    //ToUserID,
    //Readed,
    //Modified,
from '$(vRutaQVD)\WeRide\ChatMessage.qvd' (qvd);

```

```

concatenate(Hechos)
LOAD
    1 AS Conteo,
    'Comment' as TIPO,
    ID as PK_Comment,
    `UserID` as PK_User,
    if(CommentType=3,'Route comment') as CommentType,
    date(floor(Created)) as Fecha
from '$(vRutaQVD)WeRide\Comment.qvd' (qvd);

```

```

concatenate(Hechos)
LOAD
    1 as Conteo,
    'File' as TIPO,
    ID as PK_File,
    if(`Type`=1,'Group image',
        if(`Type`=2,'Route image',
            if(`Type`=4,'Wechat picture',
                if(`Type`=5,'Post wall picture',
                    if(`Type`=32,'Public group picture',
                        if(`Type`=21,'Profile picture'))))))))
    as File_Type,
    date(floor(Created)) as Fecha,
    UserID as PK_User
from '$(vRutaQVD)WeRide\File.qvd' (qvd);

```

```

concatenate(Hechos)
LOAD
    1 AS Conteo,
    'Like' as TIPO,
    ID as PK_Like,
    UserID as PK_User,
    if(LikeType=3,'Route','Wall post') as Like_Type,
    date(floor(Created)) as Fecha

```

```
from '$(vRutaQVD)\WeRide\Like.qvd' (qvd);
```

```
concatenate(Hechos)
```

```
LOAD
```

```
1 as Conteo,
```

```
'PushMessagesUsers' as TIPO,
```

```
//id,
```

```
UserID as PK_User,
```

```
//Readed
```

```
date(floor(Created)) as Fecha
```

```
from '$(vRutaQVD)\WeRide\PushMessagesUsers.qvd' (qvd);
```

```
concatenate(Hechos)
```

```
LOAD
```

```
1 as Conteo,
```

```
'UserRelation' as TIPO,
```

```
ID as PK_UserRelation,
```

```
if(`Status`=0,'Waiting','Friends') as UserRelation_Status,
```

```
date(floor(Created)) as Fecha
```

```
from '$(vRutaQVD)\WeRide\UserRelation.qvd' (qvd);
```

```
concatenate(Hechos)
```

```
LOAD
```

```
1 as Conteo,
```

```
'WeSportUser_Abs' as TIPO,
```

```
UserID as PK_User,
```

```
date(floor(Created)) as Fecha
```

```
from '$(vRutaQVD)\WeRide\WeSpotUser_Abs.qvd' (qvd);
```

```
User:
```

```
LOAD
```

```
1 as [User count],
```

```
UserID as PK_User,
```

```

HasMoto,
date(floor(Created)) as [Fec. Creación],
CountryID as User_CountryID,
//CountryID as PK_Country,

if(DeviceType=0,'Unknown',if(DeviceType=1,'iOS',if(DeviceType=2,'Android')))
as DeviceType,
Notifications,
Status as [Status User],
if(Gender=0,'Unknown',if(Gender=1,'Men',if(Gender=2,'Female','Unknown')))
as Gender,
age(today(),Birthday) as `Age`,
Birthday,
ProvinceID as PK_Province,
Version,
AppVersion,
Affiliate,
DATE(FLOOR>LastAccess)) as LastAccess,
AcceptNews,
CountryCode as User_CountryCode,
ApplyMap('MapaWeSpotUser','UserID`) as GeoUser2,
ID as User_ID_Autonum,
ID as User_id,
ApplyMap('MapaUsuariosMTBKS',Mail,'No') as User_in_mtbks,
applymap('MapaUsuariosCompradores',Mail,'No') as User_Buyer_in_mtbks,
Points,
if(Picture='456158821e995dd6b93b8be132e9e981.jpg' or IsNull(Picture) or
Picture='', 'Empty', 'Filled') as [Has Profile Picture],
Mail,
UserName
/*ID,

Mail,
ValidatedMail,

```

Password,
Picture,
Telephone,
Modified,
DeviceToken,
FacebookID,
FacebookAccessToken,
TwitterID,
TwitterAccessToken,
Bio,
Public,
Units,

CityName,
PasswordHash,

Deleted,
LanguageID,
FirstName,
LastName,

Pilot,

PostalCode,
Privacy,

ApiVersion*/
from '\$(vRutaQVD)WeRide\User.qvd' (qvd);

UserMoto:
load

```

        MotoID as PK_Moto,
        if(Default=1,'Yes','No') as [Default Moto],
        `UserID` as User_id,
        1 as [Count UserMoto]
from '$(vRutaQVD)\WeRide\UserMoto.qvd' (qvd);

```

Motos:

```

load
        ID as PK_Moto,
        Modelo_Completo,
        Marca as Motos_Brand,
        Model as Motos_Model,
        Cubicaje as Motos_Capacity
from '$(vRutaQVD)\WeRide\Motos.qvd' (qvd);

```

concatenate(Hechos)

User:

```

LOAD
        'User' as TIPO,
        1 as Conteo,
        UserID as PK_User,
        date(floor(Created)) as Fecha
from '$(vRutaQVD)\WeRide\User.qvd' (qvd);

```

WeSpotUser:

```

LOAD
        `UserID` as User_ID_Autonom,
        Active as WeSportUser_Active,
        GeoMakePoint(Latitude,Longitude) as GeoUser
from '$(vRutaQVD)\WeRide\WeSpotUser.qvd' (qvd);

```

```
concatenate(Hechos)
LOAD
    'Analytics_Sessions' as TIPO,
    date(floor(fecha)) as Fecha,
    deviceCategory,
    operatingSystem,
    operatingSystemVersion,
    ApplyMap('MapaCountryCode',country,'No') as Analytics_country,
    sessions,
    users,
    screenviews,
    sessionDuration
```

```
from '$(vRutaQVD)WeRide\GA_RowData_Qlik_WeRide.qvd' (qvd);
```

```
Pedidos:
```

```
Load
```

```
    email as Mail,
```

```
    `total` as importe_total
```

```
from '$(vRutaQVD)Dashboard\pedidos.qvd' (qvd);
```

```
//*****
```

```
*****
```

```
//*** VARIABLES DE TRABAJO
```

```
//*****
```

```
*****
```

```
Let AñoDestino = Year(Today());
```

```
Let AñoOrigen = Year(Today()) - 1;
```

```
//*****
```

```
*****
```

```
//*** TRIMESTRES
```

```
//*****
```

```
*****
```

```
Trimestres :
```



```

LOAD
    rowno() as MesTrimestre,
    'Trimestre ' & Ceil(rowno()/3) as TrimestreMap
AUTOGENERATE(12);

```

Horas :

```

LOAD
    rowno() - 1 as Hora
AUTOGENERATE(24);

```

```

//*****
*****

/*** CALENDARIO TEMPORAL
//*****
*****

```

HechosOrdenados :

```

LOAD Fecha RESIDENT Hechos WHERE NOT ISNULL(Fecha) ORDER BY
Fecha;

```

```

LET vFechaMinima = Num(Peek( 'Fecha' , 0,'HechosOrdenados'));

```

```

sET vFechaMaxima = if ( date(Num(Peek( 'Fecha' , -1,'HechosOrdenados')) )
< YearEnd(today()), num(yearend(today())), Num(Peek( 'Fecha' , -
1,'HechosOrdenados')) ) ;

```

```

LET vFechaInicial = Date($(vFechaMinima));

```

```

LET vFechaFinal = Date($(vFechaMinima));

```

```

LET vFechaReferenciaDesde=YearStart(Today());

```

```

LET vFechaReferenciaHasta=MonthEnd(Today());

```

TmpCalendario:

LOAD

\$(vFechaMinima) + Iterno() - 1 as Num,

Date\$(vFechaMinima) + Iterno() - 1) as FechaTemporal

AUTOGENERATE 1 WHILE \$(vFechaMinima) + Iterno() - 1 <= \$(vFechaMaxima);

```
//*****  
*****  
  
//*** CALENDARIO MAESTRO  
//*****  
*****
```

CalendarioMaestro:

LOAD

```
FechaTemporal AS Fecha,  
Week(FechaTemporal) as Semana,  
Year(FechaTemporal) as Año,  
Month(FechaTemporal) as Mes,  
day(FechaTemporal) as día,  
weekDay(FechaTemporal) as diaSemana,  
'Trimestre ' & ceil(num(Month(FechaTemporal))/3) as Trimestre,  
'T' & ceil(num(Month(FechaTemporal))/3) & '-' &  
right(year(FechaTemporal),2) as TrimestreAño,  
date(monthStart(FechaTemporal),'MMMM-YYYY') as MesAño2,  
date(monthstart(FechaTemporal),'MMM-YY') as MesAño,  
if(week(FechaTemporal)=1 and  
year(FechaTemporal)<>weekyear(FechaTemporal), 53,  
Week(FechaTemporal) ) & '-' & right(year(FechaTemporal),2) as  
SemanaAño
```

```
RESIDENT TmpCalendario
ORDER BY FechaTemporal ASC;
```

```
DROP Table TmpCalendario;
DROP Table Horas;
DROP Table Trimestres;
drop table HechosOrdenados;
```

```
endif
```

Anexo 3 – Incorporación Zendesk

```
package org.zendesk.client.v2;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import org.zendesk.client.v2.model.Ticket;
import org.zendesk.client.v2.model.Comment;
import org.zendesk.client.v2.model.Audit;
import org.zendesk.client.v2.model.events.Event;
import java.sql.*;
import java.sql.SQLException;
import java.util.Iterator;

import java.util.logging.Level;
import java.util.logging.Logger;

public class Main {

    public static void main(String[] args) throws ParseException{
```

```

final String jdbc_driver = "com.mysql.jdbc.Driver";
final String usuario = "contingutEliminat_InformacioConfidencial";
final String contrasena = "contingutEliminat_InformacioConfidencial";
final String url = "contingutEliminat_InformacioConfidencial";
String query="select 1;";
String queryTags="";
String queryComments="";
String queryAudit="";
Connection conn = null;
Statement st = null;
ResultSet rs = null;
java.util.Date utilDate = new java.util.Date();

try {
    Class.forName("com.mysql.jdbc.Driver");
} catch (ClassNotFoundException ex) {
    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
}

//Abrimos connexion
try{
    conn = DriverManager.getConnection(url, usuario, contrasena);
    st = conn.createStatement();
    rs = st.executeQuery("select 1;");
}catch (SQLException ex){
    //System.out.println(ex);
    //System.out.println("No he podido establecer la connexion");
}

PreparedStatement preparedStmt;
PreparedStatement preparedStmtTags;
PreparedStatement preparedStmtComments;
PreparedStatement preparedStmtAudit;

```

```

try {
    preparedStmt = conn.prepareStatement(query);
} catch (SQLException ex) {
    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
}
Zendesk          zd          =          new
Zendesk.Builder("https://motobuykerssupport.zendesk.com")
    .setUsername("contingutEliminat_InformacioConfidencial")
    .setToken("contingutEliminat_InformacioConfidencial")
    .build();

int i=0;
int tags=0;
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy kk:mm:ss");
/*Date d = sdf.parse("21/03/2016 00:00:00");
Date a = sdf.parse("21/03/2016 23:59:59");*/
Date d = sdf.parse(args[0]);
Date a = sdf.parse(args[1]);
int limite = Integer.parseInt(args[2]);
SimpleDateFormat sdf2 = new SimpleDateFormat("dd/MM/yyyy
kk:mm:ss");

for (Ticket ticket: zd.getTicketsIncrementally(d,a) ){
    query          =          "insert          into          Bl.tmp_zendesk_tickets
(url,id,external_id,channel,`from`,`to`,`rel,unix_created_at,unix_updated_at`,`type
`,`subject,description,priority`,`status`,`recipient,requesterEmail,requesterPhone,s
ubmitterEmail,submitterPhone,assignee_id,assignee_email,organization_id,gro
up_name,has_incidents,score,score_comment,custom_field_1,custom_field_1_
value,custom_field_2,custom_field_2_value,custom_field_3,custom_field_3_val
ue,custom_field_4,custom_field_4_value,custom_field_5,custom_field_5_value)
values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?);";
    try {
        preparedStmt = conn.prepareStatement(query);
    }
}

```

```

preparedStmt.setString(1, ticket.getUrl());
preparedStmt.setInt(2, ticket.getId().intValue());
if(ticket.getExternalId()==null){
    preparedStmt.setNull(3, java.sql.Types.NULL);
}
else{
    preparedStmt.setString(3, ticket.getExternalId());
}
if(ticket.getVia().getChannel()==null){
    preparedStmt.setNull(4, java.sql.Types.NULL);
}
else{
    preparedStmt.setString(4, ticket.getVia().getChannel());
}
if(ticket.getVia().getSource().get("from")==null){
    preparedStmt.setNull(5, java.sql.Types.NULL);
}
else{
    preparedStmt.setString(5,
ticket.getVia().getSource().get("from").toString());
}
if(ticket.getVia().getSource().get("to")==null){
    preparedStmt.setNull(6, java.sql.Types.NULL);
}
else{
    preparedStmt.setString(6,
ticket.getVia().getSource().get("to").toString());
}
if(ticket.getVia().getSource().get("rel")==null){
    preparedStmt.setNull(7, java.sql.Types.NULL);
}
else{
    preparedStmt.setString(7,
ticket.getVia().getSource().get("rel").toString());
}

```

```

}

preparedStmt.setLong(8, ticket.getCreatedAt().getTime()/1000);

preparedStmt.setLong(9,ticket.getUpdatedAt().getTime()/1000);
if(ticket.getType()==null){

    preparedStmt.setNull(10, java.sql.Types.NULL);
}
else{

    preparedStmt.setString(10, ticket.getType().toString());
}
if(ticket.getSubject()==null){

    preparedStmt.setNull(11, java.sql.Types.NULL);
}
else{

    preparedStmt.setString(11, ticket.getSubject());
}
if(ticket.getDescription()==null){

    preparedStmt.setNull(12, java.sql.Types.NULL);
}
else{

    preparedStmt.setString(12, ticket.getDescription());
}
if(ticket.getPriority()==null){

    preparedStmt.setNull(13, java.sql.Types.NULL);
}
else{

```

```

        preparedStmt.setString(13, ticket.getPriority().toString());
    }
    if(ticket.getStatus()==null){

        preparedStmt.setNull(14, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(14, ticket.getStatus().toString());
    }
    if(ticket.getRecipient()==null){

        preparedStmt.setNull(15, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(15, ticket.getRecipient());
    }
    if(zd.getUser(ticket.getRequesterId()).getEmail()==null){

        preparedStmt.setNull(16, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(16,
zd.getUser(ticket.getRequesterId()).getEmail());
    }
    if(zd.getUser(ticket.getRequesterId()).getPhone()==null){

        preparedStmt.setNull(17, java.sql.Types.NULL);
    }
    else{

```



```

        preparedStmt.setString(17,
zd.getUser(ticket.getRequesterId()).getPhone());
    }
    if(zd.getUser(ticket.getSubmitterId()).getEmail()==null){

        preparedStmt.setNull(18, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(18,
zd.getUser(ticket.getSubmitterId()).getEmail());
    }

    if(zd.getUser(ticket.getSubmitterId()).getPhone()==null){

        preparedStmt.setNull(19, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(19,
zd.getUser(ticket.getSubmitterId()).getPhone());
    }
    if(ticket.getAssigneeId()==null){

        preparedStmt.setNull(20, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setInt(20, ticket.getAssigneeId().intValue());
    }
    if(ticket.getAssigneeId()==null ||
zd.getUser(ticket.getAssigneeId()).getEmail()==null){

        preparedStmt.setNull(21, java.sql.Types.NULL);

```

```

    }
    else{

        preparedStmt.setString(21,
zd.getUser(ticket.getAssigneeId()).getEmail());
    }
    if(ticket.getOrganizationId()==null){

        preparedStmt.setNull(22, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setInt(22, ticket.getOrganizationId().intValue());
    }
    if(zd.getGroup(ticket.getGroupId()).getName()==null){

        preparedStmt.setNull(23, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(23,
zd.getGroup(ticket.getGroupId()).getName());
    }

    preparedStmt.setBoolean(24, ticket.isHasIncidents());

    if(ticket.getSatisfactionRating().getScore()==null){

        preparedStmt.setNull(25, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(25,
ticket.getSatisfactionRating().getScore());
    }

```

```

    }
    if(ticket.getSatisfactionRating().getComment()==null){

        preparedStmt.setNull(26, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(26,
ticket.getSatisfactionRating().getComment());
    }

    if(ticket.getCustomFields().get(0).getId()==null){

        preparedStmt.setNull(27, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setInt(27,
ticket.getCustomFields().get(0).getId().intValue());
    }
    if(ticket.getCustomFields().get(0).getValue()==null){

        preparedStmt.setNull(28, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(28,
ticket.getCustomFields().get(0).getValue());
    }

    if(ticket.getCustomFields().get(1).getId()==null){

        preparedStmt.setNull(29, java.sql.Types.NULL);
    }

```

```

else{

    preparedStmt.setInt(29,
ticket.getCustomFields().get(1).getId().intValue());
    }
    if(ticket.getCustomFields().get(1).getValue()==null){

        preparedStmt.setNull(30, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(30,
ticket.getCustomFields().get(1).getValue());
    }

    if(ticket.getCustomFields().get(2).getId()==null){

        preparedStmt.setNull(31, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setInt(31,
ticket.getCustomFields().get(2).getId().intValue());
    }
    if(ticket.getCustomFields().get(2).getValue()==null){

        preparedStmt.setNull(32, java.sql.Types.NULL);
    }
    else{

        preparedStmt.setString(32,
ticket.getCustomFields().get(2).getValue());
    }
}

```

```

if(ticket.getCustomFields().get(3).getId()==null){

    preparedStmt.setNull(33, java.sql.Types.NULL);
}
else{

    preparedStmt.setInt(33,
ticket.getCustomFields().get(3).getId().intValue());
}
if(ticket.getCustomFields().get(3).getValue()==null){

    preparedStmt.setNull(34, java.sql.Types.NULL);
}
else{

    preparedStmt.setString(34,
ticket.getCustomFields().get(3).getValue());
}

if(ticket.getCustomFields().get(4).getId()==null){

    preparedStmt.setNull(35, java.sql.Types.NULL);
}
else{

    preparedStmt.setInt(35,
ticket.getCustomFields().get(4).getId().intValue());
}
if(ticket.getCustomFields().get(4).getValue()==null){

    preparedStmt.setNull(36, java.sql.Types.NULL);
}
else{

```

```

        preparedStmt.setString(36,
ticket.getCustomFields().get(1).getValue());
    }

    if((ticket.getUpdatedAt().getTime()/1000)>limite)
    {

        break;
    }
    preparedStmt.execute();

} catch (Exception e) {
    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, e);
    continue;
}

tags=0;
while (tags<ticket.getTags().size())
{
    queryTags="insert into BI.tmpZendesk_tags(id,tag) values (?,?)";
    try {

        preparedStmtTags = conn.prepareStatement(queryTags);
        preparedStmtTags.setInt(1, ticket.getId().intValue());
        preparedStmtTags.setString(2, ticket.getTags().get(tags));

        preparedStmtTags.execute();
    } catch (Exception ex) {
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null,
ex);

        continue;
    }

    tags++;
}

```

```

    }

    if(ticket.getStatus().toString().equals("deleted"))
    {
        continue;
    }

    for (Comment comment: zd.getTicketComments(ticket.getId() )){
        queryComments="insert      into      BI.tmpZendesk_comments
(id,ticket_id,body,public,author_id,author_email,author_phone,unix_created_at)
values (?,?,?,?,?,?,?,?);";
        try {

            preparedStmtComments =
conn.prepareStatement(queryComments);

            preparedStmtComments.setInt(1, comment.getId().intValue());

            preparedStmtComments.setInt(2, ticket.getId().intValue());

            if (comment.getBody()==null)
            {

                preparedStmtComments.setNull(3, java.sql.Types.NULL);
            }
            else
            {

                preparedStmtComments.setString(3, comment.getBody());
            }

            preparedStmtComments.setBoolean(4, comment.isPublic());

```

```

        preparedStmtComments.setInt(5,
comment.getAuthorId().intValue());

        if (zd.getUser(comment.getAuthorId()).getEmail()==null)
        {

            preparedStmtComments.setNull(6, java.sql.Types.NULL);
        }
        else
        {

            preparedStmtComments.setString(6,
zd.getUser(comment.getAuthorId()).getEmail());
        }
        if (zd.getUser(comment.getAuthorId()).getPhone()==null)
        {

            preparedStmtComments.setNull(7, java.sql.Types.NULL);
        }
        else
        {

            preparedStmtComments.setString(7,
zd.getUser(comment.getAuthorId()).getPhone());
        }
        preparedStmtComments.setLong(8,
comment.getCreatedAt().getTime()/1000);
        preparedStmtComments.execute();
    } catch (Exception ex) {
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null,
ex);
        continue;
    }
}
}

```



```

        i++;

    }

    try {
        conn.close();
    } catch (Exception ex) {
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
    }

    System.exit(0);
}
}

```

Anexo 4 – Incorporación Google Adwords

```

import com.google.api.ads.adwords.lib.client.AdWordsSession;
import com.google.api.ads.adwords.lib.client.reporting.ReportingConfiguration;
import com.google.api.ads.adwords.lib.jaxb.v201509.DateRange;
import com.google.api.ads.adwords.lib.jaxb.v201509.DownloadFormat;
import com.google.api.ads.adwords.lib.jaxb.v201509.ReportDefinition;
import
com.google.api.ads.adwords.lib.jaxb.v201509.ReportDefinitionDateRangeType;
import
com.google.api.ads.adwords.lib.jaxb.v201509.ReportDefinitionReportType;
import com.google.api.ads.adwords.lib.jaxb.v201509.Selector;
import com.google.api.ads.adwords.lib.utils.ReportDownloadResponse;
import
com.google.api.ads.adwords.lib.utils.ReportDownloadResponseException;
import com.google.api.ads.adwords.lib.utils.v201509.ReportDownloader;
import com.google.api.ads.common.lib.auth.OfflineCredentials;
import com.google.api.ads.common.lib.auth.OfflineCredentials.Api;
import com.google.api.client.auth.oauth2.Credential;

```

```
//import com.google.api.client.util.DateTime;
import com.google.common.collect.Lists;
```

```
import java.io.File;
```

```
/*-----
```

Este programa se crea con el objetivo de descargar la información de adwords.

La extracción de datos se puede hacer por rest api pero primero se necesita el refresh token que, al menos, es mucho mas facil sacarlo por java.

Primero desarrolle el programa para que solo exportara los datos del día de ayer

pero me he dado cuenta que adwords me acaba engañando. Los tiempos de descarga

son irrisorios, por tanto decido que cada día se exporten todos los datos.

```
-----*/
```

```
public class Main {
```

```
    private static final int PAGE_SIZE = 100;
```

```
    public static void main(String[] args) throws Exception{
```

```
        if(args.length==0)
```

```
        {
```

```
            System.exit(0);
```

```
        }
```

```
        String fechaInicial = args[0];
```

```
        String fechaFinal = args[1];
```

```
        String clientCustomerId = args[2];
```

```
        // Generate a refreshable OAuth2 credential.
```

```
        Credential oAuth2Credential = new OfflineCredentials.Builder()
```

```

        .forApi(Api.ADWORDS)
        .fromFile()
        .build()
        .generateCredential();

// Construct an AdWordsSession.
AdWordsSession session = new AdWordsSession.Builder()
    .fromFile()
    .withClientCustomerId(clientCustomerId)
    .withOAuth2Credential(oAuth2Credential)
    .build();

// Location to download report to.
String reportFile = System.getProperty("user.home") + File.separatorChar
+ "report" + System.currentTimeMillis() + ".tsv";

if(!System.getProperty("os.name").contains("Windows"))
{
    reportFile =
"/home/pentaho/ETL/Adwords/GetCampaign_Performance/resultados/report" +
System.currentTimeMillis() + ".tsv";
}

descargarInforme(session, reportFile, fechaInicial, fechaFinal);

}

public static void descargarInforme(AdWordsSession session, String
reportFile, String fechaInicial, String fechaFinal) throws Exception {
// Create selector.
Selector selector = new Selector();
selector.getFields().addAll(Lists.newArrayList(
"CampaignId",
"CampaignName",

```

```

    "Clicks",
    "Impressions",
    "Cost",
    "Date"
)
);
DateRange range = new DateRange();
range.setMin(fechaInicial);
range.setMax(fechaFinal);
/*selector.setDateRange(range); activar si solo se quiere un periodo*/

// Create report definition.
ReportDefinition reportDefinition = new ReportDefinition();
reportDefinition.setReportName("Campaign performance report #" +
System.currentTimeMillis());

/*reportDefinition.setDateRangeType(ReportDefinitionDateRangeType.CUSTOM_DATE); Activar si solo se quiere un periodo*/

reportDefinition.setDateRangeType(ReportDefinitionDateRangeType.ALL_TIME);

reportDefinition.setReportType(ReportDefinitionReportType.CAMPAIGN_PERFORMANCE_REPORT);
reportDefinition.setDownloadFormat(DownloadFormat.TSV);

// Optional: Set the reporting configuration of the session to suppress header,
column name, or
// summary rows in the report output. You can also configure this via your
ads.properties
// configuration file. See AdWordsSession.Builder.from(Configuration) for
details.

```

// In addition, you can set whether you want to explicitly include or exclude zero impression

// rows.

ReportingConfiguration reportingConfiguration =

new ReportingConfiguration.Builder()

.skipReportHeader(true)

.skipColumnHeader(false)

.skipReportSummary(true)

// Enable to allow rows with zero impressions to show.

.includeZeroImpressions(false)

.build();

session.setReportingConfiguration(reportingConfiguration);

reportDefinition.setSelector(selector) ;

try {

// Set the property api.adwords.reportDownloadTimeout or call

// ReportDownloader.setReportDownloadTimeout to set a timeout (in milliseconds)

// for CONNECT and READ in report downloads.

ReportDownloadResponse response =

new ReportDownloader(session).downloadReport(reportDefinition);

/*System.out.println(response.getAsString());*/

response.saveToFile(reportFile);

System.out.printf("Report successfully downloaded to: %s%n", reportFile);

} catch (ReportDownloadResponseException e) {

System.out.printf("Report was not downloaded due to: %s%n", e);

}

}

}

Anexo 5 – Incorporación Google Analytics

- GetAnalyticsQlik.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package GetAnalyticsQlik;

/**
 *
 * @author LT14
 */

import java.sql.*;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;
import java.sql.*;

public class GetAnalyticsQlik {
    private final String jdbc_driver = "com.mysql.jdbc.Driver";
    private final String usuario = "contingutEliminat_InformacioConfidencial";
    private final String contrasena = "contingutEliminat_InformacioConfidencial";
    private final String url = "contingutEliminat_InformacioConfidencial";
```

```

private String fechalInicio;
private String fechaFin;

private GetAnalyticsQlik(String arg1, String arg2/*, String arg3, String arg4,
String arg5, String arg6,String arg7,String arg8*/) {
    this.fechalInicio=arg1;
    this.fechaFin=arg2;
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) throws ClassNotFoundException {
    System.out.println("Iteracion con argumentos: "+ args.length);
    System.out.println("Argunmentos: "+args[0]+"|"+args[1]);
    GetAnalyticsQlik gau1 = new GetAnalyticsQlik(args[0],args[1]);
    ArrayList resultados;
    Class.forName("com.mysql.jdbc.Driver");

    String [][] segmentos = new String[5][2];
    segmentos[0][0]="1"; //Espana
    segmentos[0][1]="gaid::DdZCxSXUTli1okcMea95Pg";
    segmentos[1][0]="2"; //Italia
    segmentos[1][1]="gaid::8Z5ifc_HRGGL28b49B8f5g";
    segmentos[2][0]="3"; //Francia
    segmentos[2][1]="gaid::cNOL6J6GSJeiwHBC_eOvZg";
    segmentos[3][0]="4"; //UK
    segmentos[3][1]="gaid::X1aA1b8hQ46C9SL33QZN5w";
    segmentos[4][0]="5"; //COM
    segmentos[4][1]="gaid::QihylQKzTC-POByduEcTVQ";

    Connection conn = null;
    Statement st = null;

```

```

ResultSet rs = null;
String query = null;
int xx=0;
//Abrimos connexion
try{
    conn = DriverManager.getConnection(gau1.getUrl(), gau1.getUsuario(),
gau1.getContrasena());
    st = conn.createStatement();
    rs = st.executeQuery("select 1");
}catch (SQLException ex){
    System.out.println(ex);
    System.out.println("No he podido establecer la conexion");
}

/*Creamos un objeto para pedir a analytics*/

int i=0;

for(i=0;i<segmentos.length;i++)
{
    //System.out.println(i);
    xx=0;
    PeticionAnalytics pa1 = new
PeticionAnalytics(args[0],args[1],segmentos[i][1]);

    resultados = pa1.peticionUsuarios(args[0],args[1],segmentos[i][1]);

    while (xx<resultados.size()){
        try {
            st.executeUpdate(resultados.get(xx).toString());
        } catch (SQLException ex) {

Logger.getLogger(GetAnalyticsQlik.class.getName()).log(Level.SEVERE, null,
ex);

```



```

    }
    xx+=1;
}
}
//Ahora hacemos la misma taera pero exportamos los ids de transacción
i=0;
for(i=0;i<segmentos.length;i++)
{
    //System.out.println(i);
    xx=0;
    PeticionAnalyticsTransacciones pa2 = new
PeticionAnalyticsTransacciones(args[0],args[1],segmentos[i][1]);
    resultados = pa2.peticionUsuarios(args[0],args[1],segmentos[i][1]);
    while (xx<resultados.size()){
        try {
            st.executeUpdate(resultados.get(xx).toString());
        } catch (SQLException ex) {

Logger.getLogger(GetAnalyticsQlik.class.getName()).log(Level.SEVERE, null,
ex);
        }
        xx+=1;
    }
}

}

/**
 * @return the fechalnicio
 */
public String getFechalnicio() {
    return fechalnicio;
}

```

```
/**
 * @param fechalnicio the fechalnicio to set
 */
public void setFechalnicio(String fechalnicio) {
    this.fechalnicio = fechalnicio;
}
```

```
/**
 * @return the fechaFin
 */
public String getFechaFin() {
    return fechaFin;
}
```

```
/**
 * @param fechaFin the fechaFin to set
 */
public void setFechaFin(String fechaFin) {
    this.fechaFin = fechaFin;
}
```

```
/**
 * @return the jdbc_driver
 */
public String getJdbc_driver() {
    return jdbc_driver;
}
```

```
/**
 * @return the usuario
 */
public String getUsuario() {
```

```

        return usuario;
    }

    /**
     * @return the contrasena
     */
    public String getContrasena() {
        return contrasena;
    }

    /**
     * @return the url
     */
    public String getUrl() {
        return url;
    }
}

```

- PeticionAnalytics.java

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package GetAnalyticsQlik;

import java.io.IOException;
import java.security.GeneralSecurityException;
import java.util.HashSet;
import java.util.Set;

import com.google.api.client.googleapis.auth.oauth2.GoogleCredential;
import com.google.api.client.http.HttpRequest;

```

```

import com.google.api.client.http.HttpRequestInitializer;
import com.google.api.client.http.HttpTransport;
import com.google.api.client.http.javanet.NetHttpTransport;
import com.google.api.client.json.jackson2.JacksonFactory;
import com.google.api.services.analytics.Analytics;
import com.google.api.services.analytics.AnalyticsScopes;
import com.google.api.services.analytics.model.GaData;
import java.util.ArrayList;
import java.util.HashMap;

public class PeticionAnalytics {

    private String profile;
    private String fechalnicio;
    private String fechaFin;
    private String filtros;
    private String segmentos;
    private String tabla;
    private String zona;
    private String entidad;

    private String urlAnalytics;

    private static final String APPLICATION_NAME = "JavaAnalytics";

    private static final String SERVICE_ACCOUNT_EMAIL =
"eliminadoporseguridad@developer.gserviceaccount.com";

    /** Path to the Service Account's Private Key file */
    private static String SERVICE_ACCOUNT_PKCS12_FILE_PATH =
"C:\\Users\\LT14\\Desktop\\UnidadD\\p12\\JavaAnalytics-
317ef31e05f0.p12";

```

```

public String getSistemaOperativo() {
    return SistemaOperativo;
}

public void setSistemaOperativo(String SistemaOperativo) {
    this.SistemaOperativo = SistemaOperativo;
}

String SistemaOperativo = System.getProperty("os.name");

public PeticionAnalytics(String arg0, String arg1, String arg2) {
    this.fechaInicio=arg0;
    this.fechaFin=arg1;
    this.segmentos=arg2;
}

public static Analytics getAnalyticsService() throws
GeneralSecurityException, IOException {
    Set<String> scopes = new HashSet<>();
    scopes.add(AnalyticsScopes.ANALYTICS_READONLY); // You can set
other scopes if needed

    HttpTransport httpTransport = new NetHttpTransport();
    JacksonFactory jsonFactory = new JacksonFactory();
    GoogleCredential credential = new GoogleCredential.Builder()
        .setTransport(httpTransport)
        .setJsonFactory(jsonFactory)
        .setServiceAccountId(SERVICE_ACCOUNT_EMAIL)
        .setServiceAccountScopes(scopes)
        .setServiceAccountPrivateKeyFromP12File(
            new java.io.File(SERVICE_ACCOUNT_PKCS12_FILE_PATH))
        .build();
    final Analytics service = new Analytics.Builder(httpTransport,
jsonFactory, setHttpTimeout(credential))

```

```

        .setHttpRequestInitializer(credential)
        .setApplicationName(APPLICATION_NAME)
        .build();
    return service;
}

private static HttpRequestInitializer setHttpTimeout(final
HttpRequestInitializer requestInitializer) {
return new HttpRequestInitializer() {
@Override
public void initialize(HttpRequest httpRequest) throws IOException {
requestInitializer.initialize(httpRequest);
httpRequest.setConnectTimeout(10 * 60000); // 10 minutes connect
timeout
httpRequest.setReadTimeout(10 * 60000); // 10 minutes read timeout
}
};}

public ArrayList petitionUsuarios(String arg0, String arg1, String arg2){
ArrayList<String> resultados = new ArrayList();
String query="";

try {
int x = 0;
/*System.out.println(SistemaOperativo);*/
if (!SistemaOperativo.contains("Windows")){

SERVICE_ACCOUNT_PKCS12_FILE_PATH="/home/pentaho/JavaAnalytic
s-317ef31e05f0.p12";
}

Analytics service = getAnalyticsService();
GaData result = null;

```

```

result = service.data().ga().get(
    "ga:85263851",
    arg0, //Start Date
    arg1, // End date

"ga:sessions,ga:users,ga:newUsers,ga:pageviews,ga:transactions,ga:transactionRevenue,ga:bounces,ga:sessionDuration,ga:searchSessions")

.setDimensions("ga:date,ga:deviceCategory,ga:channelGrouping,ga:source,ga:medium")
    .setSamplingLevel("HIGHER_PRECISION")
    .setMaxResults(10000)
    .setSegment(arg2)
    .execute();

String fecha="";
String fechaF="";
while (x<result.getRows().size()){

    fecha=result.getRows().get(x).get(0);
    String zona="";
    if(segmentos=="gaid::DdZCxSXUTli1okcMea95Pg")
    {
        zona="1";
    }
    else if (segmentos=="gaid::8Z5ifc_HRGGL28b49B8f5g")
    {
        zona="2";
    }
    else if (segmentos=="gaid::cNOL6J6GSJeiwHBC_eOvZg")
    {
        zona="3";
    }
    else if (segmentos=="gaid::X1aA1b8hQ46C9SL33QZN5w")

```

```

        {
            zona="4";
        }
        else if (segmentos=="gaid::QihylQKzTC-POByduEcTVQ")
        {
            zona="5";
        }

        query="insert          into          BI.GA_RowData_Qlik_tmp
(fecha,id_zona,deviceCategory,channelGrouping,source,medium,sessions,u
sers,newUsers,pageviews,transactions,transactionRevenue,bounces,sessio
nDuration,searchSessions) values ("+ fechal.substring(0, 4) + "-" +
fechal.substring(4, 6) + "-" + fechal.substring(6, 8) + ",";
        query=query + Integer.parseInt(zona) + "," +
result.getRows().get(x).get(1) + "," + result.getRows().get(x).get(2) + "," +
result.getRows().get(x).get(3) + "," + result.getRows().get(x).get(4) + ",";
        query=query + result.getRows().get(x).get(5) + "," +
result.getRows().get(x).get(6) + "," + result.getRows().get(x).get(7) + "," +
result.getRows().get(x).get(8) + ",";
        query=query + result.getRows().get(x).get(9) + "," +
result.getRows().get(x).get(10) + ",";
        query=query + result.getRows().get(x).get(11) + "," +
result.getRows().get(x).get(12) + "," + result.getRows().get(x).get(13);
        query=query + ");";
        resultados.add(query);
        //System.out.println(query);

        x+=1;
    }

} catch (GeneralSecurityException | IOException |
NumberFormatException e) {

```



```

        System.err.println(e.getMessage());
    }

    return resultados;
}
}

```

- PeticionAnalyticsTransacciones.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package GetAnalyticsQlik;

import com.google.api.client.googleapis.auth.oauth2.GoogleCredential;
import com.google.api.client.http.HttpRequest;
import com.google.api.client.http.HttpRequestInitializer;
import com.google.api.client.http.HttpTransport;
import com.google.api.client.http.javanet.NetHttpTransport;
import com.google.api.client.json.jackson2.JacksonFactory;
import com.google.api.services.analytics.Analytics;
import com.google.api.services.analytics.AnalyticsScopes;
import com.google.api.services.analytics.model.GaData;
import java.io.IOException;
import java.security.GeneralSecurityException;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Set;

/**
 *
 * @author Oscar
 */

```

```

public class PeticionAnalyticsTransacciones {

    private String fechaInicio;
    private String fechaFin;
    private String segmentos;

    private String urlAnalytics;

    private static final String APPLICATION_NAME = "JavaAnalytics";

    private static final String SERVICE_ACCOUNT_EMAIL =
"eliminadoporseguridad@developer.gserviceaccount.com";

    /** Path to the Service Account's Private Key file */
    private static String SERVICE_ACCOUNT_PKCS12_FILE_PATH =
"C:\\Users\\LT14\\Desktop\\UnidadD\\p12\\JavaAnalytics-317ef31e05f0.p12";

    public String getSistemaOperativo() {
        return SistemaOperativo;
    }

    public void setSistemaOperativo(String SistemaOperativo) {
        this.SistemaOperativo = SistemaOperativo;
    }

    String SistemaOperativo = System.getProperty("os.name");

    public PeticionAnalyticsTransacciones(String arg0, String arg1, String arg2) {
        this.fechaInicio=arg0;
        this.fechaFin=arg1;
        this.segmentos=arg2;
    }
}

```

```

public static Analytics getAnalyticsService() throws
GeneralSecurityException, IOException {
    Set<String> scopes = new HashSet<>();
    scopes.add(AnalyticsScopes.ANALYTICS_READONLY); // You can set
other scopes if needed

    HttpTransport httpTransport = new NetHttpTransport();
    JacksonFactory jsonFactory = new JacksonFactory();
    GoogleCredential credential = new GoogleCredential.Builder()
        .setTransport(httpTransport)
        .setJsonFactory(jsonFactory)
        .setServiceAccountId(SERVICE_ACCOUNT_EMAIL)
        .setServiceAccountScopes(scopes)
        .setServiceAccountPrivateKeyFromP12File(
            new java.io.File(SERVICE_ACCOUNT_PKCS12_FILE_PATH))
        .build();
    final Analytics service = new Analytics.Builder(httpTransport, jsonFactory,
setHttpTimeout(credential))
        .setHttpRequestInitializer(credential)
        .setApplicationName(APPLICATION_NAME)
        .build();
    return service;
}

```

```

private static HttpRequestInitializer setHttpTimeout(final
HttpRequestInitializer requestInitializer) {
return new HttpRequestInitializer() {
    @Override
    public void initialize(HttpRequest httpRequest) throws IOException {
        requestInitializer.initialize(httpRequest);
        httpRequest.setConnectTimeout(10 * 60000); // 10 minutes connect
timeout
        httpRequest.setReadTimeout(10 * 60000); // 10 minutes read timeout
    }
}

```

```
};}
```

```
public ArrayList peticionUsuarios(String arg0, String arg1, String arg2){
    ArrayList<String> resultados = new ArrayList();
    String query="";

    //System.out.println("Entro en la segunda");

    /*System.out.println("he entrado en la funcion");*/
    try {
        int x = 0;
        /*System.out.println(SistemaOperativo);*/
        if (!SistemaOperativo.contains("Windows")){

SERVICE_ACCOUNT_PKCS12_FILE_PATH="/home/pentaho/JavaAnalytics-
317ef31e05f0.p12";
        }

        Analytics service = getAnalyticsService();
        GaData result = null;

        result = service.data().ga().get(
            "ga:85263851",
            arg0, //Start Date
            arg1, // End date
            "ga:transactions,ga:transactionRevenue,ga:transactionShipping")

        .setDimensions("ga:date,ga:deviceCategory,ga:channelGrouping,ga:source,ga:
medium,ga:transactionId,ga:adwordsCampaignID")
        .setSamplingLevel("HIGHER_PRECISION")
        .setMaxResults(10000)
        .setSegment(arg2)
        .execute();
```

```

String fechal="";
while (x<result.getRows().size()){

    fechal=result.getRows().get(x).get(0);
    String zona="";
    if(segmentos=="gaid::DdZCxSXUTli1okcMea95Pg")
    {
        zona="1";
    }
    else if (segmentos=="gaid::8Z5ifc_HRGGL28b49B8f5g")
    {
        zona="2";
    }
    else if (segmentos=="gaid::cNOL6J6GSJeiwHBC_eOvZg")
    {
        zona="3";
    }
    else if (segmentos=="gaid::X1aA1b8hQ46C9SL33QZN5w")
    {
        zona="4";
    }
    else if (segmentos=="gaid::QihylQKzTC-POByduEcTVQ")
    {
        zona="5";
    }

    query="insert    into    BI.GA_RowData_Qlik_Transacciones_tmp
(fecha,id_zona,deviceCategory,channelGrouping,source,medium,transactionId,t
ransactions,transactionRevenue,adwordsCampaignID,transactionShipping)
values ("+ fechal.substring(0, 4) + "-" + fechal.substring(4, 6) + "-" +
fechal.substring(6, 8) + ",";
        query=query    +    Integer.parseInt(zona)    +    ","    +
result.getRows().get(x).get(1) + "," + result.getRows().get(x).get(2) + "," +
result.getRows().get(x).get(3) + "," + result.getRows().get(x).get(4) + ",";

```

```

        query=query + result.getRows().get(x).get(5) + "," +
result.getRows().get(x).get(7) + "," + result.getRows().get(x).get(8) + "," +
result.getRows().get(x).get(6);
        query=query + "," + result.getRows().get(x).get(9) + " ";
        resultados.add(query);
        /*System.out.println(query);*/

        x+=1;
    }

} catch (Exception e) {
    System.err.println(e.getMessage());
}

return resultados;
}
}

```

Anexo 6 – Google Analytics información de rendimiento de producto

/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

```
package main;
```

```
import java.io.IOException;

import java.security.GeneralSecurityException;

import java.util.HashSet;

import java.util.Set;

import com.google.api.client.googleapis.auth.oauth2.GoogleCredential;

import com.google.api.client.http.HttpRequest;

import com.google.api.client.http.HttpRequestInitializer;

import com.google.api.client.http.HttpTransport;

import com.google.api.client.http.javanet.NetHttpTransport;

import com.google.api.client.json.jackson2.JacksonFactory;

import com.google.api.services.analytics.Analytics;

import com.google.api.services.analytics.AnalyticsScopes;

import com.google.api.services.analytics.model.GaData;

import java.util.ArrayList;

import java.util.HashMap;
```

```
public class PeticionAnalytics {

    private String profile;

    private String fechaInicio;

    private String fechaFin;

    private String filtros;

    private String segmentos;

    private String tabla;

    private String zona;

    private String entidad;

    private String urlAnalytics;

    private static final String APPLICATION_NAME = "JavaAnalytics";

    private static final String SERVICE_ACCOUNT_EMAIL =
"eliminadoporseguridad@developer.gserviceaccount.com";
```



```

/** Path to the Service Account's Private Key file */

private static String SERVICE_ACCOUNT_PKCS12_FILE_PATH =
"C:\\Users\\LT14\\Desktop\\UnidadD\\p12\\JavaAnalytics-317ef31e05f0.p12";

public String getSistemaOperativo() {

    return SistemaOperativo;

}

public void setSistemaOperativo(String SistemaOperativo) {

    this.SistemaOperativo = SistemaOperativo;

}

String SistemaOperativo = System.getProperty("os.name");

public PeticionAnalytics(String arg0, String arg1) {

    this.fechaInicio=arg0;

```

```
this.fechaFin=arg1;  
  
}
```

```
public static Analytics getAnalyticsService() throws  
GeneralSecurityException, IOException {
```

```
    Set<String> scopes = new HashSet<>();
```

```
    scopes.add(AnalyticsScopes.ANALYTICS_READONLY); // You can set  
other scopes if needed
```

```
    HttpTransport httpTransport = new NetHttpTransport();
```

```
    JacksonFactory jsonFactory = new JacksonFactory();
```

```
    GoogleCredential credential = new GoogleCredential.Builder()
```

```
        .setTransport(httpTransport)
```

```
        .setJsonFactory(jsonFactory)
```

```
        .setServiceAccountId(SERVICE_ACCOUNT_EMAIL)
```

```
        .setServiceAccountScopes(scopes)
```

```
        .setServiceAccountPrivateKeyFromP12File(  
            new java.io.File(SERVICE_ACCOUNT_PKCS12_FILE_PATH))
```

```

        .build();

        final Analytics service = new Analytics.Builder(httpTransport, jsonFactory,
setHttpTimeout(credential))

        .setHttpRequestInitializer(credential)

        .setApplicationName(APPLICATION_NAME)

        .build();

return service;

}

```

```

private static HttpRequestInitializer setHttpTimeout(final
HttpRequestInitializer requestInitializer) {

return new HttpRequestInitializer() {

@Override

public void initialize(HttpRequest httpRequest) throws IOException {

requestInitializer.initialize(httpRequest);

httpRequest.setConnectTimeout(10 * 60000); // 10 minutes connect
timeout

httpRequest.setReadTimeout(10 * 60000); // 10 minutes read timeout

}
}

```

```
};}
```

```
public ArrayList peticionUsuarios(String arg0, String arg1){
```

```
    ArrayList<String> resultados = new ArrayList();
```

```
    String query="";
```

```
    /*System.out.println("he entrado en la funcion");*/
```

```
    try {
```

```
        /*System.out.println(SistemaOperativo);*/
```

```
        if (!SistemaOperativo.contains("Windows")){
```

```
SERVICE_ACCOUNT_PKCS12_FILE_PATH="/home/pentaho/JavaAnalytics-  
317ef31e05f0.p12";
```

```
    }
```

```
    Analytics service = getAnalyticsService();
```

```

GaData result = null;

int startIndex=1;

int x;

do

{

    x = 0;

    result = service.data().ga().get(

        "ga:85263851",

        arg0, //Start Date

        arg1, // End date

        "ga:pageviews,ga:uniquePageviews,ga:timeOnPage,ga:entrances,ga:bounces,

        ga:exits")

        .setDimensions("ga:date,ga:pagePath,ga:deviceCategory,ga:channelGrouping,

        ga:source,ga:medium,ga:hostname")

        .setSamplingLevel("HIGHER_PRECISION")

        .setStartIndex(startIndex)

```

```
.setFilters("ga:pagePath=@/producto/,ga:pagePath=@/prodotto,ga:pagePath=@/produit/,ga:pagePath=@/product/")
```

```
    .setMaxResults(10000)
```

```
    .execute();
```

```
if (result.getRows().size()==10000)
```

```
{
```

```
    startIndex+=10000;
```

```
}
```

```
String fechal="";
```

```
String fechaF="";
```

```
while (x<result.getRows().size()){
```

```
    fechal=result.getRows().get(x).get(0);
```

```
        query="insert    into    BI.GA_RowData_Qlik_Inf_Productos_tmp
(fecha,deviceCategory,channelGrouping,source,medium,pageviews,uniquePag
eviews,timeOnPage,entrances,bounces,exits,pagePath,hostname) values (" +
fechal.substring(0, 4) + "-" + fechal.substring(4, 6) + "-" + fechal.substring(6, 8) +
",";
```

```
        query=query + "" + result.getRows().get(x).get(2) + "," +
result.getRows().get(x).get(3) + "," + result.getRows().get(x).get(4) + "," +
result.getRows().get(x).get(5) + ",";
```

```
        query=query + result.getRows().get(x).get(7) + "," +
result.getRows().get(x).get(8) + "," + result.getRows().get(x).get(9) + "," +
result.getRows().get(x).get(10) + ",";
```

```
        query=query + result.getRows().get(x).get(11) + "," +
result.getRows().get(x).get(12) + ",";
```

```
        query=query + result.getRows().get(x).get(1).replace("", "") + "," +
result.getRows().get(x).get(6) + ",";
```

```
        resultados.add(query);
```

```
        /*System.out.println(query);*/
```

```
        x+=1;
```

```
    }
```

```
    } while (startIndex-1==resultados.size());
```

```
    System.out.println(resultados.size());
```

```
} catch (Exception e) {  
  
    System.err.println(e.getMessage());  
  
}  
  
return resultados;  
  
}  
  
}
```