Talend Tips

BIG DATA – BUSINESS INTELLIGENCE – MACHINE LEARNING

Google Big Query-Cargas Incrementales-Debugging

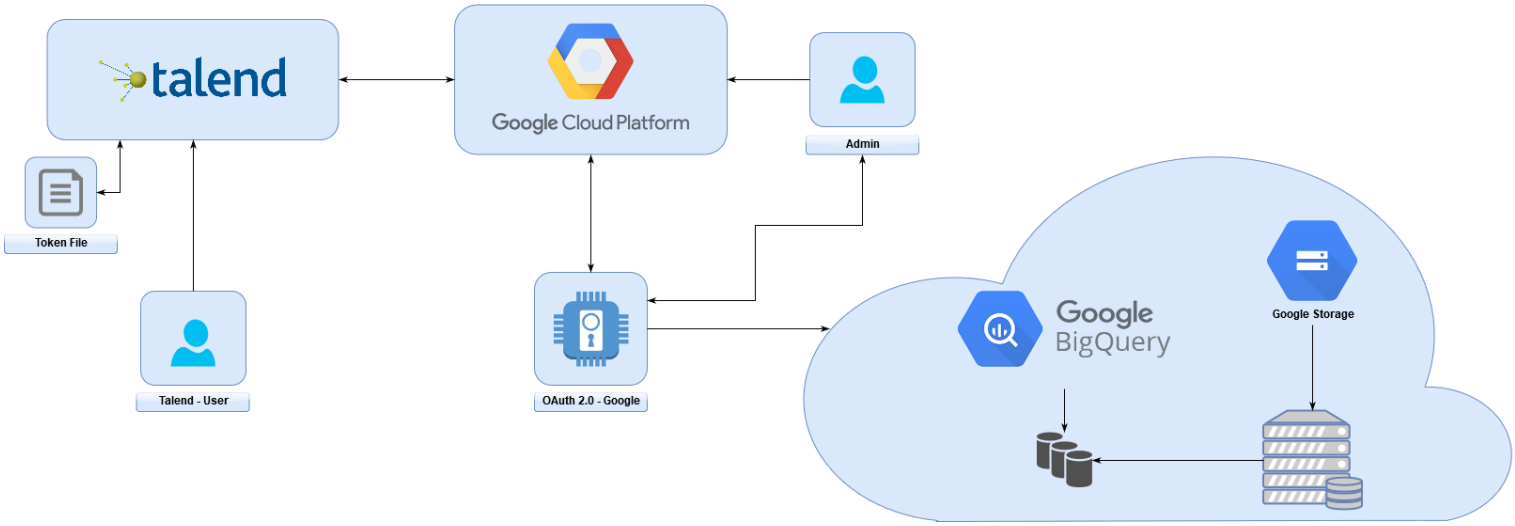
2019

1. GOOGLE BIG QUERY

1.1 Introducción

BigQuery es un servicio web RESTful perteneciente a Google Cloud que permite un análisis interactivo con *datasets* masivos, trabaja en conjunto con Google Storage, para la carga de datos. Google Cloud es una *Infrastructure as a Service* (IaaS), donde provee APIs, almacenamiento y demás servicios.

1.2 Arquitectura



1.2.1 TALEND

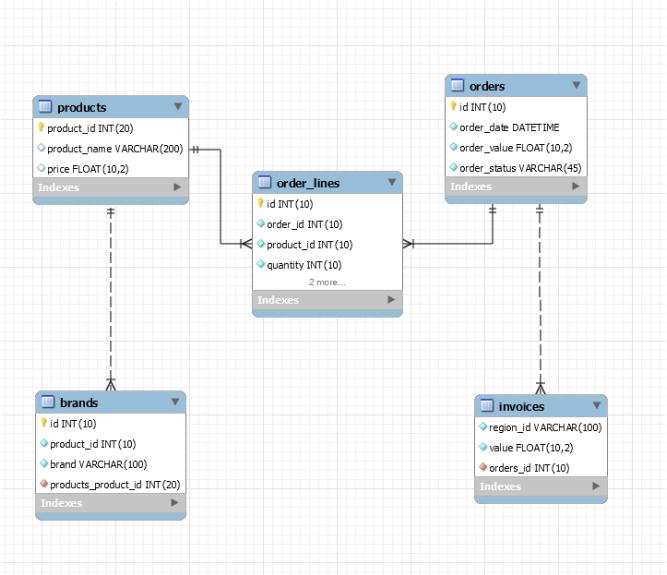
Talend dispone de distintos componentes para la interactuación con Google Bigquery, los cuales se dividen en *inputs* e *outputs*.

1.2.1.1 BULKS

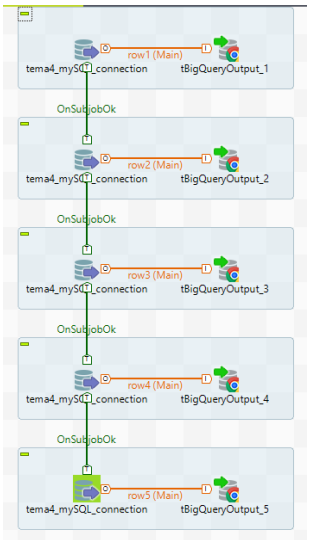
Los componentes tBigQueryOutputBulk y tBigQueryBulkExec se usan juntos para la carga masiva de una base de datos en Google BigQuery, dado un fichero con todas las inserciones se sube a Google Storage para su inserción.

1.2.1.2 OUTPUTS

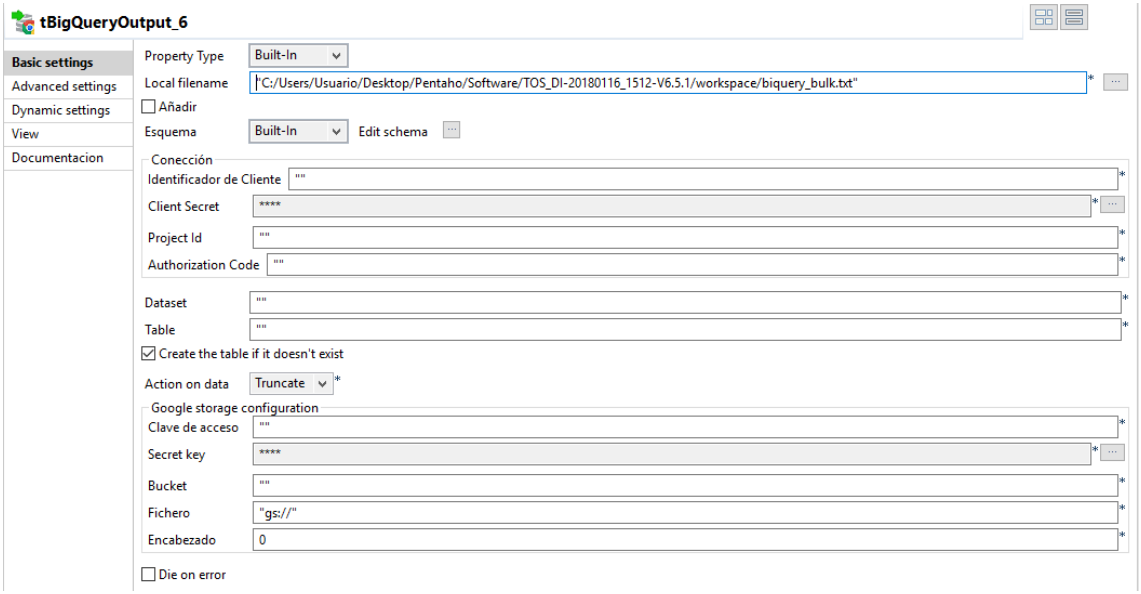
Para la carga de datos y creación de tablas se usa el componente tBigQueryOutput  . Se quiere proceder a cargar la siguiente base de datos (MySQL) en Google Bigquery:



Por tanto, en Talend se crea el siguiente *Job*, el cual creará cada una de las tablas y las cargará con los datos disponibles:



Antes de ejecutar el *Job* se procede a configurar el componente, la vista inicial sin alguna configuración es:



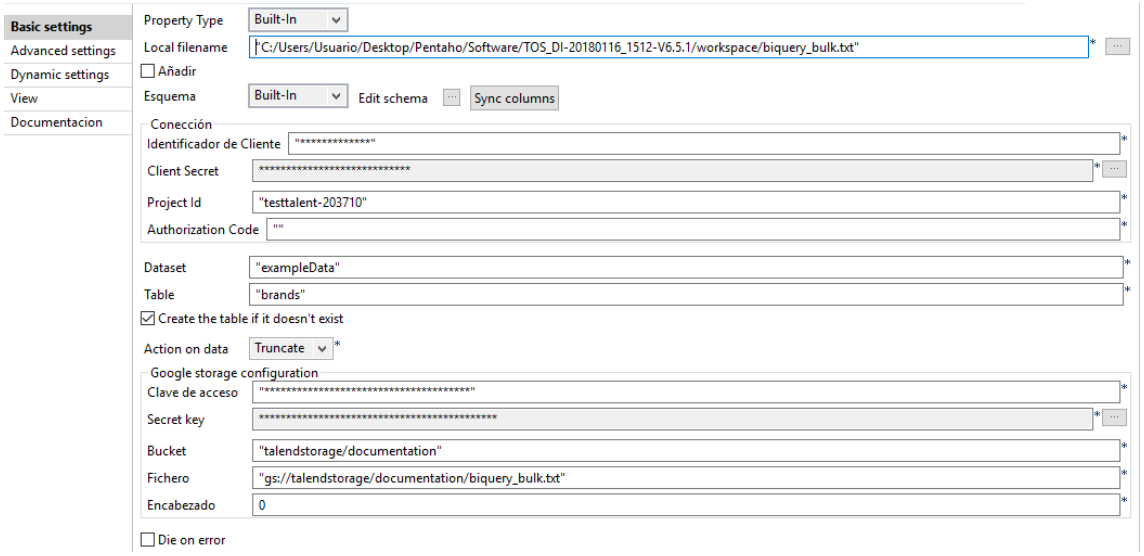
Como se observa dispone de varios parámetros obligatorios, la mayoría de ellos dependen del administrador de la cuenta de Google Cloud, el cual tendrá que suministrar un usuario para la escritura en BigQuery.

Google BigQuery: parámetros.

- ID cliente: suministrado por Administrador y generado por el mismo.
- Client Secret: clave asociada al cliente generado.
- Project ID: ID asociado al proyecto en Google Cloud.
- Authorization Code: Este parámetro se dejará en blanco inicialmente y se explicará tras la configuración del resto de parámetros.
- Dataset: nombre de la bbdd.
- Table: nombre de tabla.

Google Storage: parámetros.

- Clave de acceso: suministrada por Administrador asociada a un segmento del storage.
- Secret key: clave asociada al segmento.
- Bucket: ruta de almacenamiento.
- Fichero: ruta de localización del fichero que se subirá y servirá para la carga de datos.



Tras rellenar todos los parámetros exceptuando *Authorization Code*, se procede a ejecutar el *Job* y aparecerá un mensaje parecido al siguiente en forma de error de compilación:

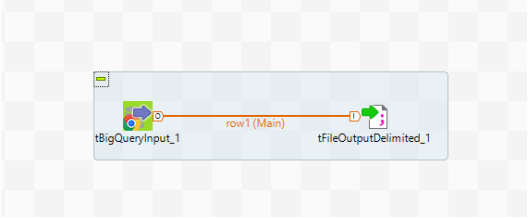
```
Starting job testBigQuery1_inputQuery at 11:31 14/05/2018.
[statistics] connecting to socket on port 4029
[statistics] connected
Exception in component tBigQueryInput_1 (testBigQuery1_inputQuery)
java.lang.Exception: Authorization Code error
    at
formacion_talent.testbigquery1_inputquery_1_0.testBigQuery1_inputQuery.tBigQuery
Input_1Process(testBigQuery1_inputQuery.java:953)
    at
formacion_talent.testbigquery1_inputquery_1_0.testBigQuery1_inputQuery.runJobInT
OS(testBigQuery1_inputQuery.java:4277)
    at
formacion_talent.testbigquery1_inputquery_1_0.testBigQuery1_inputQuery.main(test
BigQuery1_inputQuery.java:4106)
Paste this URL into a web browser to authorize BigQuery Access:
https://accounts.google.com/o/oauth2/auth?client_id=92337050928-f3cjj8uvp8vhiv0j
n5bqop5n2n0u5o48.apps.googleusercontent.com&redirect_uri=urn:ietf:wg:oauth:2.0:o
ob&response_type=code&scope=https://www.googleapis.com/auth/bigquery&state
[statistics] disconnected
Job testBigQuery1_inputQuery ended at 11:31 14/05/2018. [exit code=1]
```

El cual indica una URL para conseguir el *Authorization Code*, una vez introducido se vuelve a ejecutar el *Job*.

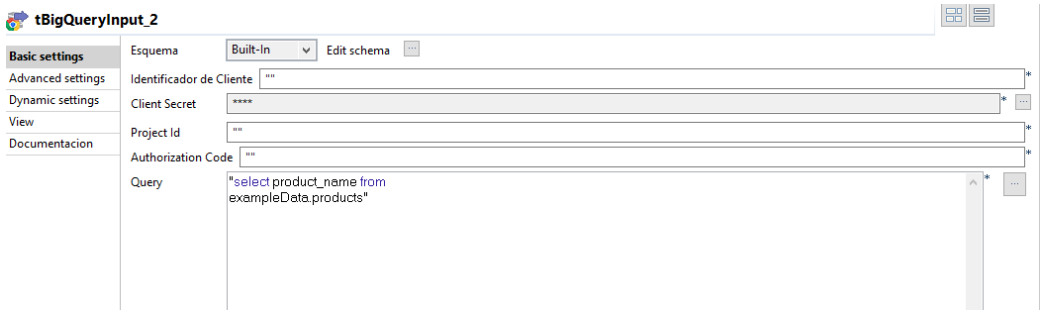
Donde finalmente se obtiene su correcta ejecución.

1.2.1.3 INPUTS

El componente **tBigQueryInput**  , para la descarga de datos o inserciones en otra base de datos requiere también de una previa configuración del componente. Se supone el simple caso de un *Job* para sacar los datos de una consulta en un *csv*.



La configuración del componente es más simple que su hermano *ouput*, ya que no es necesario indicar el repositorio de Google Storage.



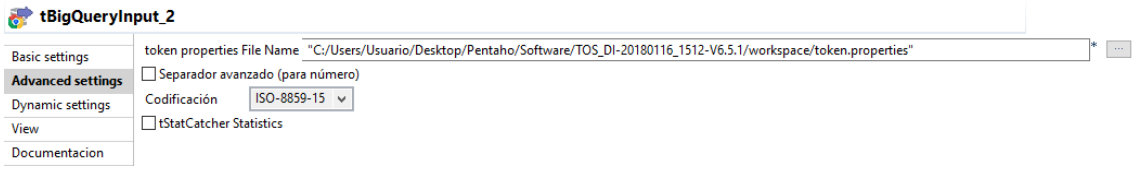
Para su ejecución se debe ejecutar el *Job* sin rellenar el *Authorization Code*, obtenerlo a través de la URL y realizar el *Job* una vez más.

ANEXO CUENTAS CLIENTES DE ESCRITURA Y LECTURA

Se ha observado que es necesario obtener dos cuentas dadas por el Administrador una para realizar *inputs* y otra para realizar *outputs*.

ANEXO TOKENS

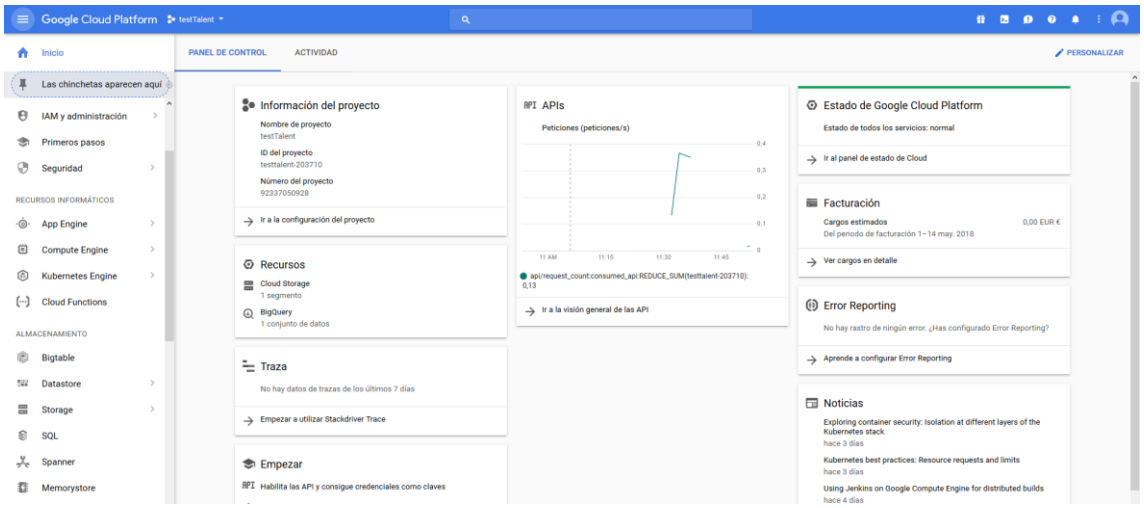
Cuando finalmente se ejecuta un *Job* se crea en el workspace de Talend un fichero *properties* donde se almacena un *token* para la autorización del *Job* en google, se puede definir en propiedades avanzadas del componente.



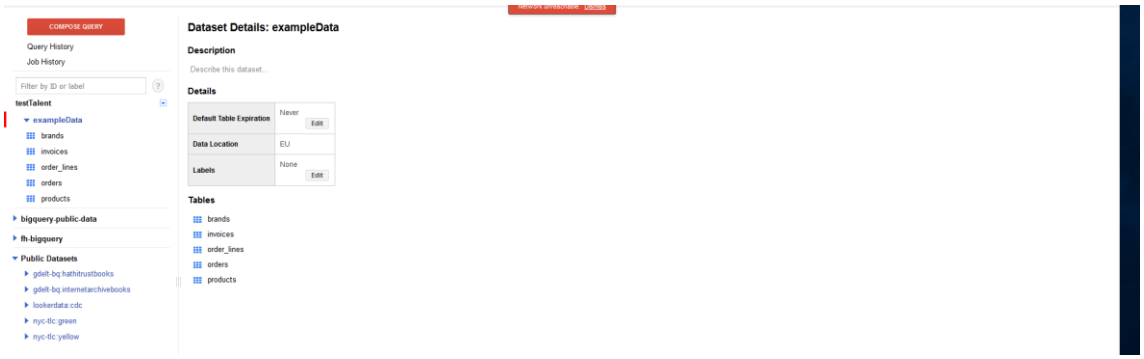
Se recomienda renombrar el *token* porque todos los componentes llaman al fichero *token* por el mismo nombre así que no podrán existir dos *Jobs* con el mismo fichero *token*.

1.2.2 GOOGLE CLOUD PLATAFORM

Como se ha mencionado Google Cloud es un *IaaS*, por lo tanto, dispone de una consola para controlar los servicios asociados desde https://console.cloud.google.com/home/dashboard, donde se puede encontrar en este caso BigQuery y Google Storage.



Para el acceso a BigQuery se accede desde "https://bigquery.cloud.google.com" en donde se almacenan los distintos *datasets* disponibles, ya sean suministrados de ejemplo por Google o los creados por el administrador de la cuenta.



Se puede observar que se dispone de la bbdd creada previamente a partir del *Job* anterior.

1.2.2.1 CLIENTES Y CREDENCIALES

Como se mencionó en el apartado de Talend es necesario suministrar a los *Jobs* de los parámetros necesarios para su correcta ejecución, en este bloque se abarca como obtener los parámetros de Cliente ID y Client Secret.

Para ello el administrador debe ir a la consola de Google Cloud y acceder a "APIs y servicios" > "Credenciales", seguidamente debe crear dos credenciales uno de escritura y otro de lectura, a través, del botón "Crear credenciales" > "Id de cliente en

OAuth". En tipo de aplicación debe seleccionar "Otro" e indicarle un nombre, a partir de este momento ya se dispondrá del Cliente ID y Client Secret.

1.2.2.2 GOOGLE STORAGE

Google Storage es accesible desde la consola de Google Cloud, en su bloque se debe crear el repositorio donde se almacenarán los datos de los inputs para las cargas en Google BigQuery.

Para ello se debe pulsar en "crear segmento" e indicar el tipo de almacenamiento deseado. Tras su creación se debe ir al apartado de "configuración" y habilitar la "Interoperabilidad" para obtener los parámetros de *clave de acceso* y *secret* pertenecientes a Google Storage.

1.3 Conclusiones

Google BigQuery con Talend es una combinación para tratamiento de *BigData* muy potente pero no se encuentra totalmente integrada con la potencia de Talend, por ejemplo, Talend permite ser "*buildeado*" para ejecutarse de forma independiente en cualquier maquina sin tener instalado Talend Open Studio, pero al existir los *tokens de autorización* y sus *respectivos códigos* suministrados por Google Cloud no va a permitir una correcta ejecución.

Actualmente los *Authorization Codes* se tratan como un error de compilación el cual nos devuelve un mensaje de excepción en forma de URL, por lo tanto, existirán problemas a la hora de intentar ejecutar un *Job* sin tener instalado Talend Open Studio.

Otro punto negativo a resaltar que a diferencia de Talend, Google Cloud es un *IaaS* y no es gratuito, aunque den un año de prueba se debe establecer un plan de subscrición de pagos, uno para Google Bigquery y otro para Google Storage.

2. INCREMENTAL LOADS

Introducción

Cuando se habla de *incremental loads* normalmente se refiere a la carga en una *data-warehause* los registros que hayan sido modificados o que no existan (*inserts, updates*) desde la última carga. Difiere totalmente de las cargas totales o *full loads* que incluyen todos los registros incluso aquellos que no hayan cambiado desde la última carga.

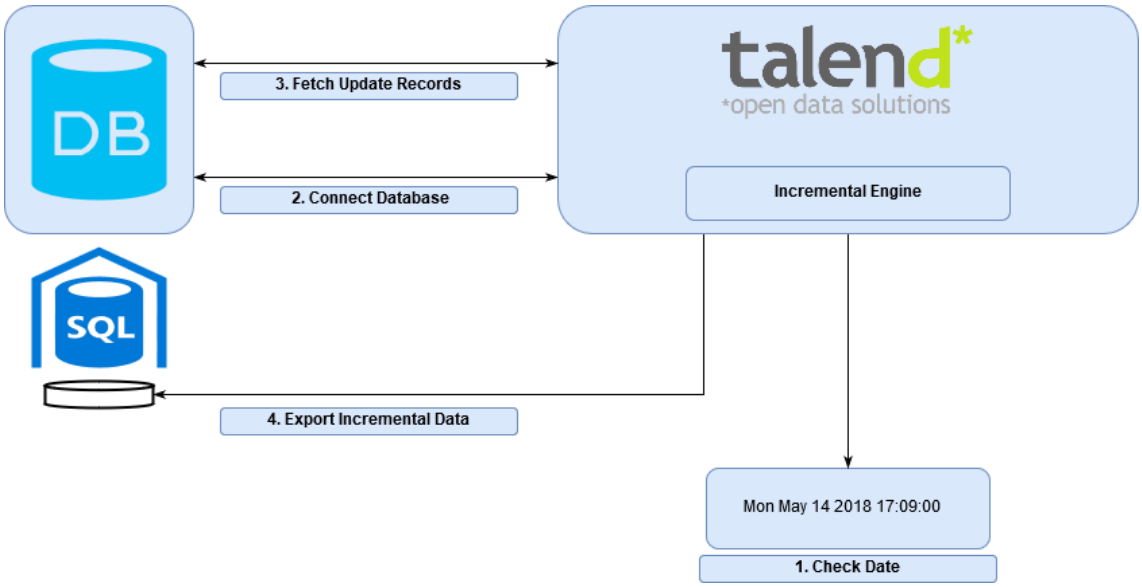
VENTAJAS

La principal ventaja es la reducción de la cantidad de datos que están siendo transferidos pudiendo reducir en horas o días la carga.

DESVENTAJAS

La principal desventaja gira entorno de la mantenibilidad. Con una *full load*, si existe algún error simplemente debes ejecutar esta carga de nuevo, mientras que, con una carga incremental deben realizarse las cargas en orden nuevamente para mantener la consistencia en el *data-warehouse*.

Arquitectura



Slowly Changing Dimensions SCD

Las *SCD* son dimensiones que están cambiando lentamente a medida que pasa el tiempo. En una data-warehouse es necesario llevar un registro de estos cambios para indicar los datos históricos.

Divididas en distintos tipos, desde tipo 0 a tipo 6. Pero donde se puede aplicar las cargas incrementales son en las de tipo 2. En las cuales se tiene un registro del tiempo.

Los registros adicionales de tipo 2 por parte de talend son:

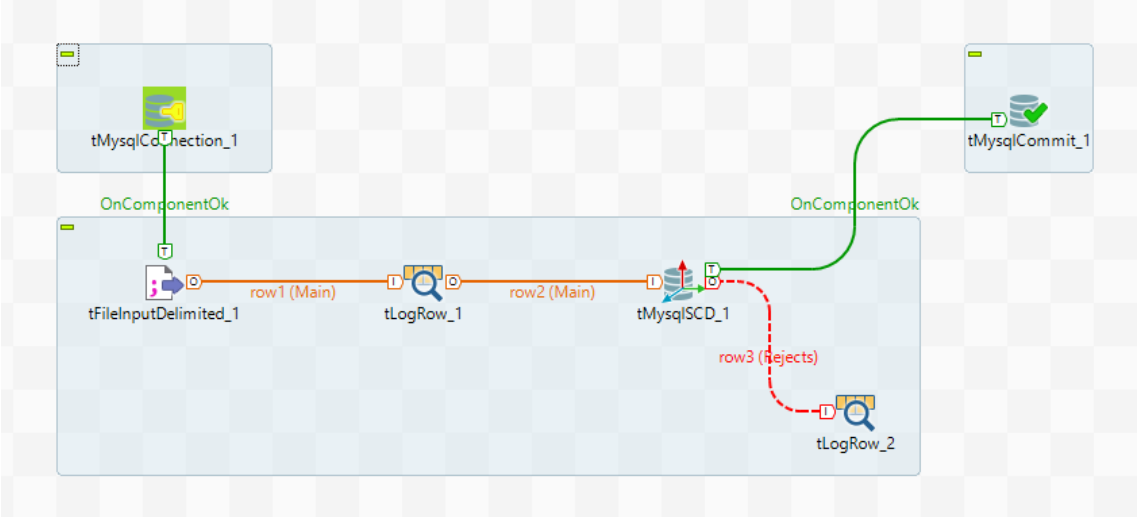
Versioning				
	type	name	creation	complement
	start	scd_start	Job start time	
	end	scd_end	Fixed year value	3000
☑	version	scd_version		
☑	active	scd_active		

Declarados en el editor del componente SCD permiten crear un registro con las versiones de la *fila/row*.

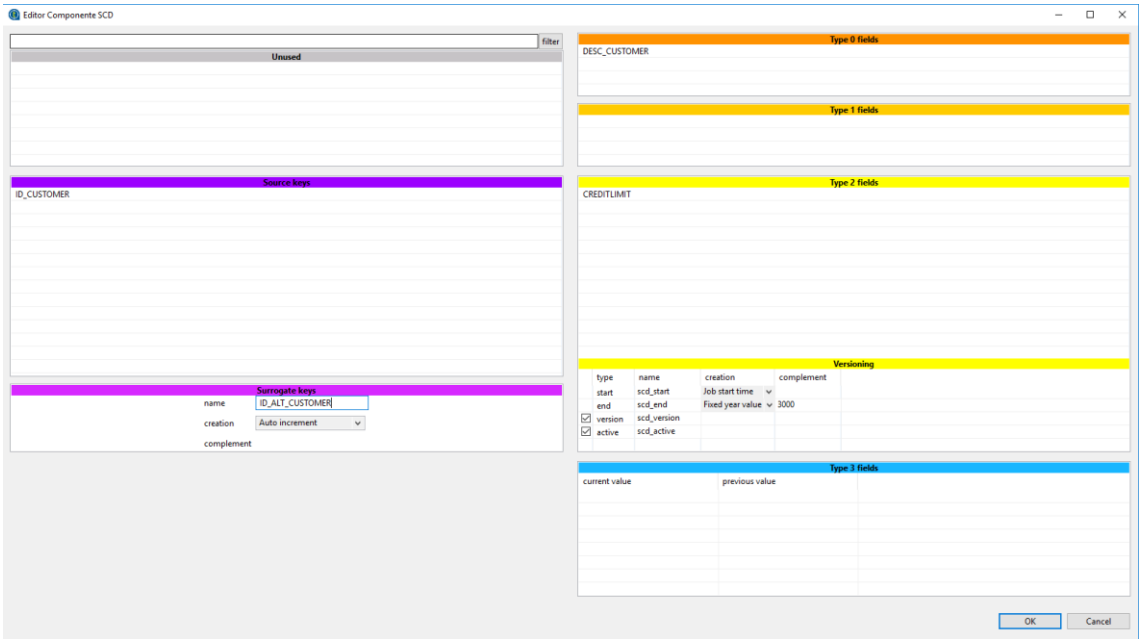
Gracias a estos registros de fechas se podrán realizar las *incremental loads* en dos pasos:

- Step 1: si se quieren actualizar datos lanzar una *query* que elimine las filas comprendidas entre las fechas deseadas (diarias, semanales, mensuales o anuales dependiendo de la granularidad de la fecha).
- Step 2: insertar los datos actualizados.

Por supuesto este proceso conlleva a transformar las tablas requeridas para los *incremental loads* teniendo que realizar *Jobs* previos como el siguiente:



Donde se transformarán las tablas a *SCD* de tipo 2 y posteriormente se realizarán las *queries* necesarias para implantar las *incremental loads*.



En el caso de ejemplo nuestra tabla final dispondrá de los siguientes atributos:

Db Column	Cl...	Tipo
ID_CUSTOMER	☑	Integer
DESC_CUSTOMER	☐	String
CREDITLIMIT	☐	Long
scd_start	☐	Date
scd_end	☐	Date
scd_version	☐	int
scd_active	☐	boolean

Como se ha mencionado previamente, se usarán las columnas scd_start y scd_end para la obtención de las *queries* para las cargas.

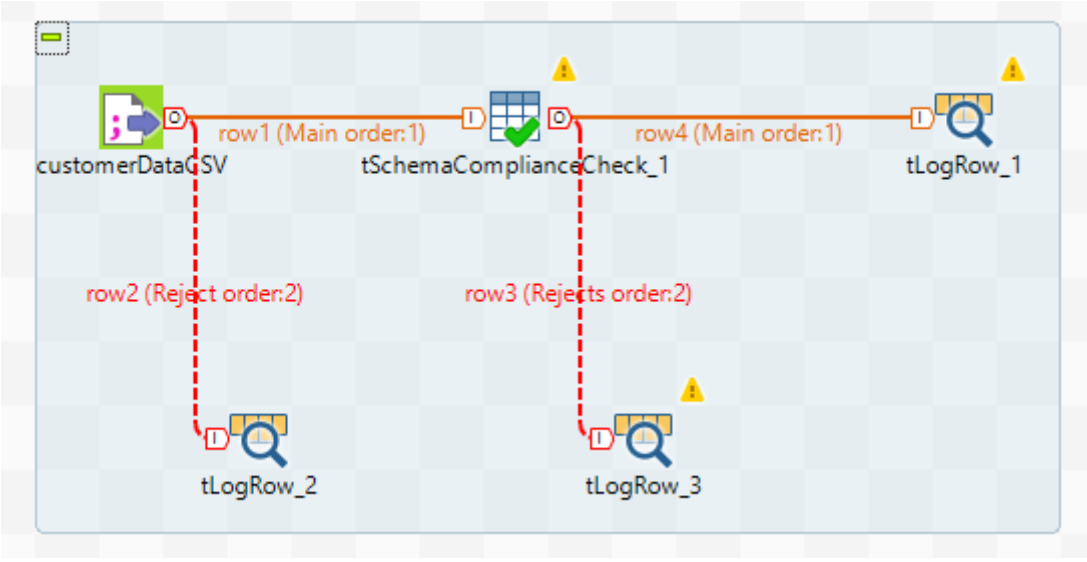
3. DEBUGGING

Talend permite ejecutar los *Jobs* de distintas formas desde la forma más normal de compilación hasta los distintos modos de debugging como *Java Debug Mode* o *Traces Debug Mode*.

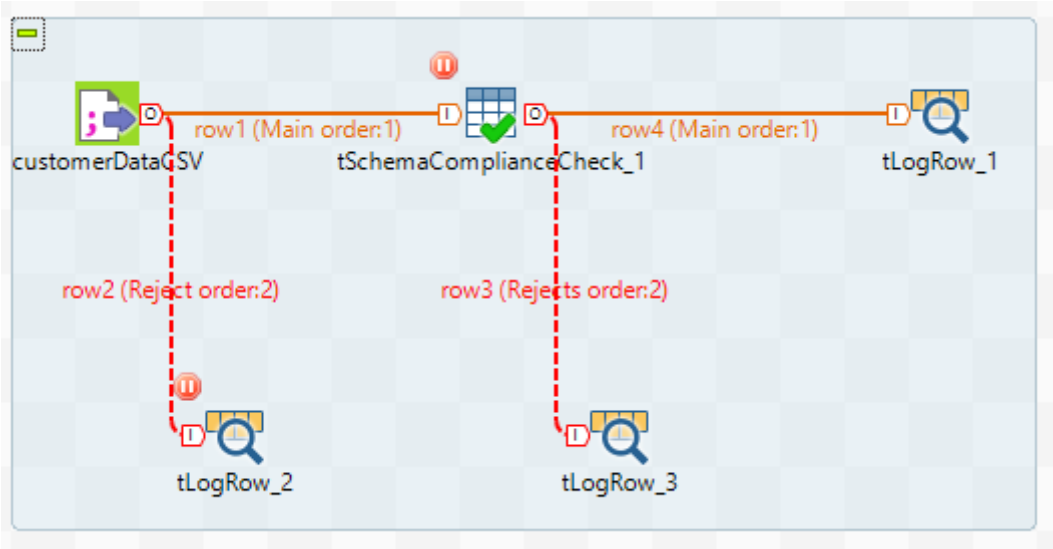
Java Debug

Para poder iniciar un *Job* en modo *Java Debug*, es necesario previamente acceder a *run* y cambiar la vista a *debug run*, además de seleccionar el modo de *debug* en este caso *Java Debug*.

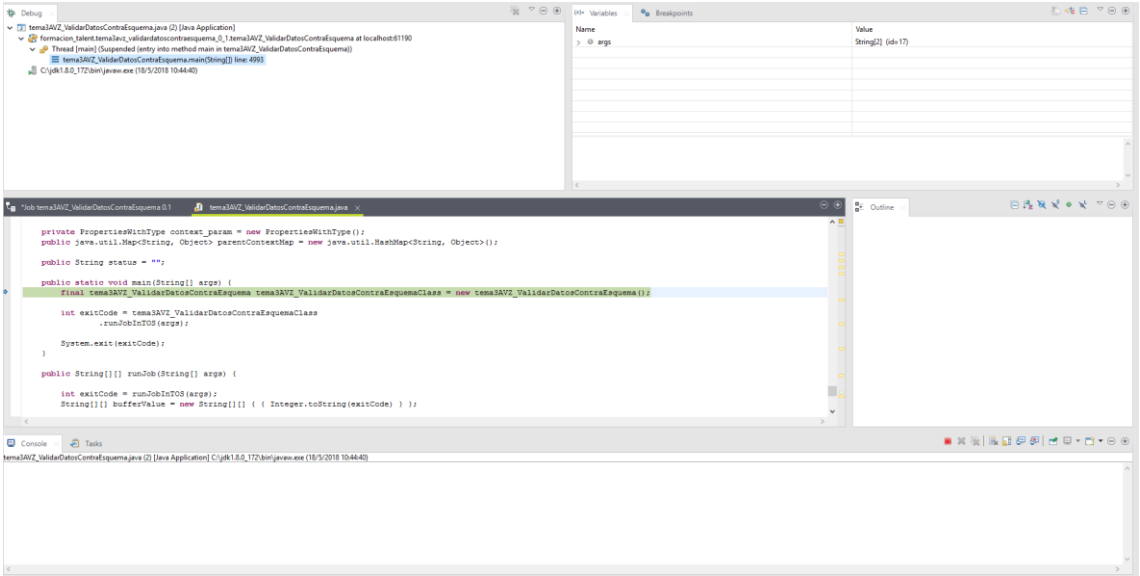
Se dispone del siguiente *Job:*



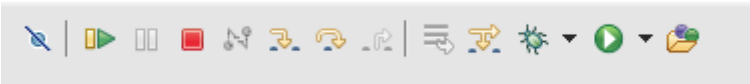
El cual valida el schema de CustomerData contra el Componente "tSchemaCompilanceCheck". Se procede a la ejecución en *Java Debug*, seguidamente se introducen un *breakpoint* para el modo debug (click derecho al componente *add breakpoint*).



Cuando iniciemos el modo *Java Debug*, se cambiará la perspectiva de Talend.



En la cual se puede trabar con los distintos comandos permitidos por *Java*



, los comandos se dividen en:

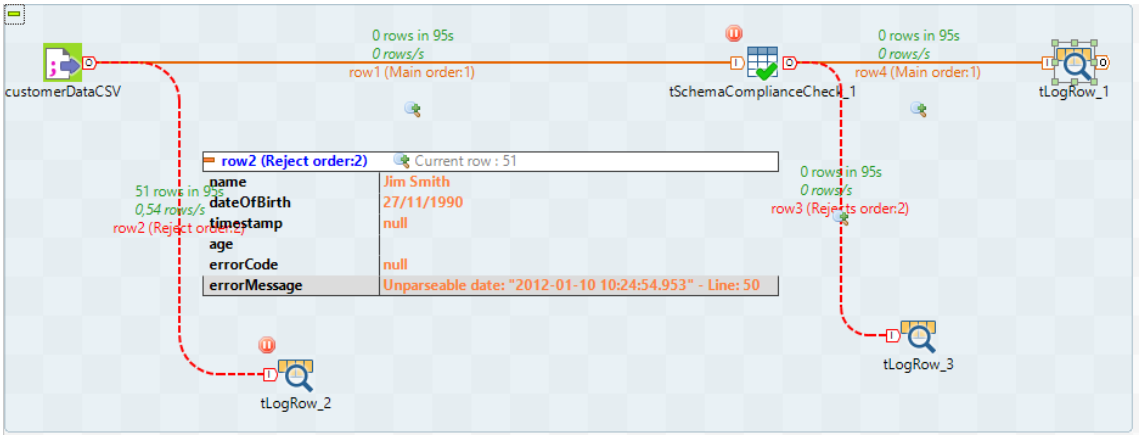
- un botón para activar o desactivar los *breakpoints* declarados previamente.
- continuar con la compilación hasta el siguiente *breakpoint*.
- pausar la compilación.

-  detener la compilación.
-  los tres siguientes son los llamados *steps*, el primero *step into* ejecutará linea por línea el código, el segundo *step over* se usará principalmente para ejecutar un método sin tener que entrar al codigo interno. Finalmente, *step return* para volver al principio de un método que se ha entrado con *step into* y no se quería entrar.

Además de tener la pantalla del codigo mostrándonos el flujo de compilación es interesante observar el cuadro de variables y como se actualizan a medida que se avanza en la compilación.

Traces Debug

Este modo *Debug* permite apreciar el flujo/procesamiento de los datos cuando se ejecuta mantenido la perspectiva de *Integración*. Permite ver el análisis de fila por fila y el comportamiento de cada una.



La ventaja principal de este modo es poder observar el comportamiento de todos los componentes sin tener que cambiar a *Java Debug Mode*. Las ventanas emergentes proporcionan los datos del *schema* y un mensaje de error, para ayudar al usuario a identificar posibles *bugs*, por ejemplo, en esta última imagen nos indica que no está reconociendo el formato de fecha en la row 51.

4. INTEGRACION CONTINUA

La versión Talend Enterprise permite la integración continua de sus proyectos ya sea a través de *Talend Administration Center* o al aplicar la integración con Jenkins u otro tipo de servidor de CI.

Talend Administration Center

Talend Administration Center es una aplicación Web que permite a los administradores manejar usuarios, proyectos y acceso a repositorios remotos.

Dispone de una herramienta *Publisher* capaz de ejecutar Test sobre distintos Jobs, por supuesto, estas publicaciones se pueden programar para que se realicen en un periodo de tiempo.

Integración con Jenkins

En caso de que se disponga de propio servidor para la integración Continua (ej Jenkings), Talend Enterprise suministra Talend CI Builder, un plugin maven que transforma todos los Jobs requeridos en clases Java para poder ejecutar los Test deseados.

5. DESCRIPCIÓN

En Stratebi ofrecemos **gran cantidad de soluciones analíticas** por una compañía de **rápido crecimiento,** innovando en las áreas tecnológicas de mayor desarrollo en la actualidad: **Business Intelligence, Big Data y Social Intelligence**, muchas de ellas, basadas en soluciones **Open Source**.

Además, somos **Partners Certificados en Microsoft PowerBI, Talend y Vertica**, con gran número de proyectos con ámbas tecnologías, siendo creadores de la Solución Big Data Analytics LinceBI

info@stratebi.com

www.stratebi.com

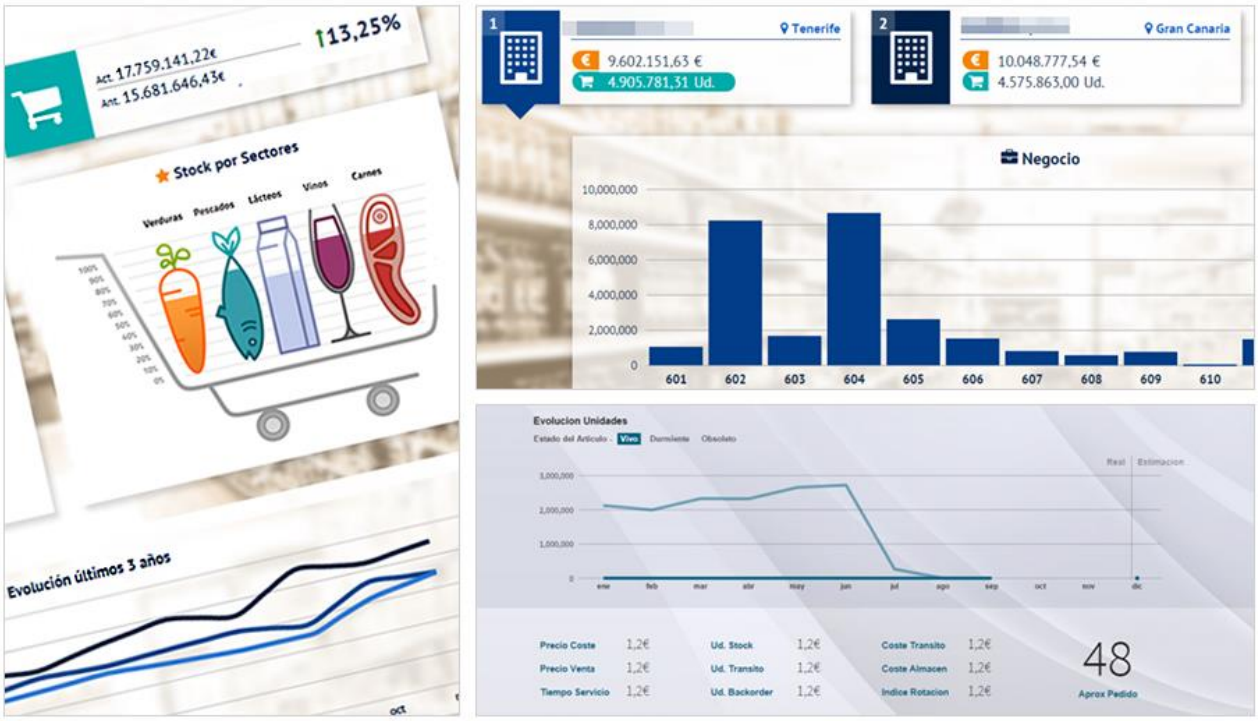
Desarrollamos nuevas soluciones analíticas basadas en Open Source, para la generación de Cuadros de Mando en tiempo real, con tecnologías IoT para SmartCities, machine learning, etc...

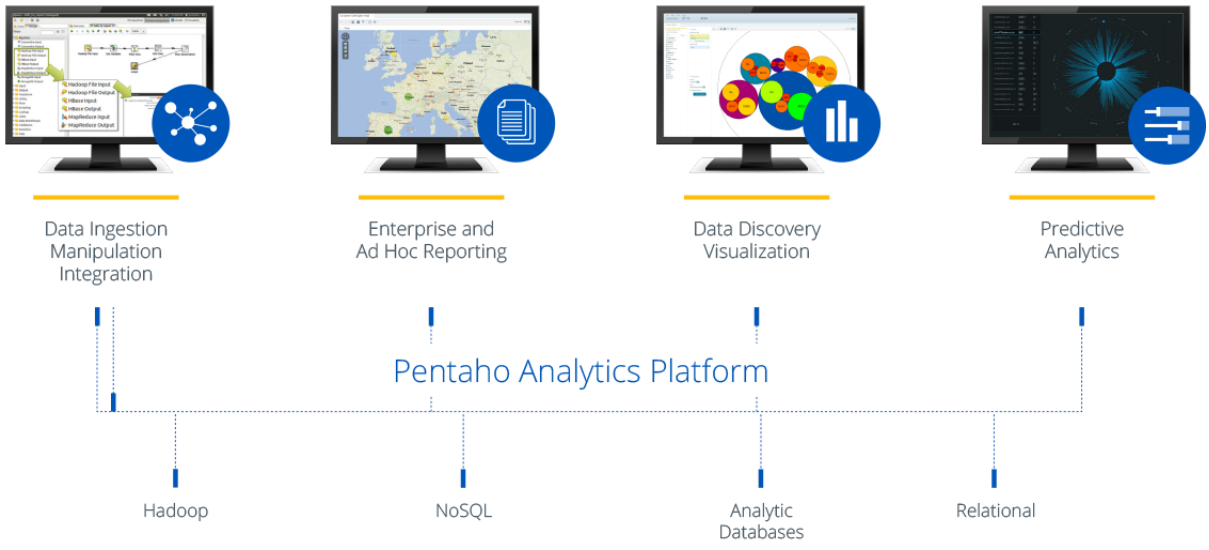
6. TECNOLOGÍAS

Recientemente, hemos sido nombrados Partners Certificados de Vertica, Talend y Microsoft PowerBI

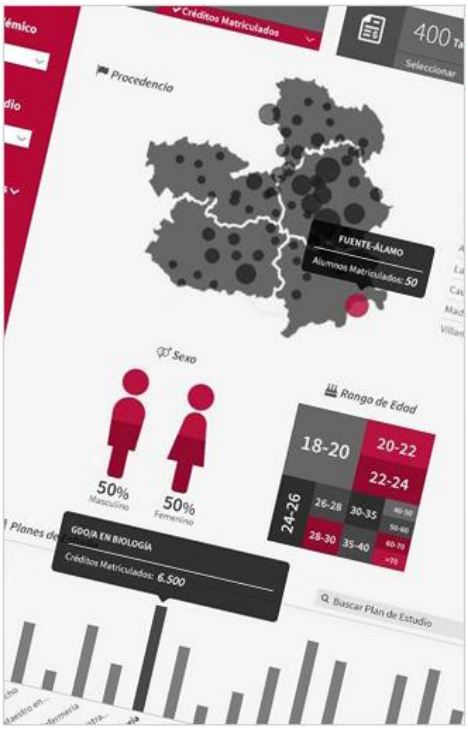
7. EJEMPLOS DE DESARROLLOS ANALYTICS

A continuación se presentan **ejemplos de algunos screenshots** de cuadros de mando diseñados por Stratebi, con el fin de dar a conocer lo que se puede llegar a obtener, así como Demos Online en la web de Stratebi:



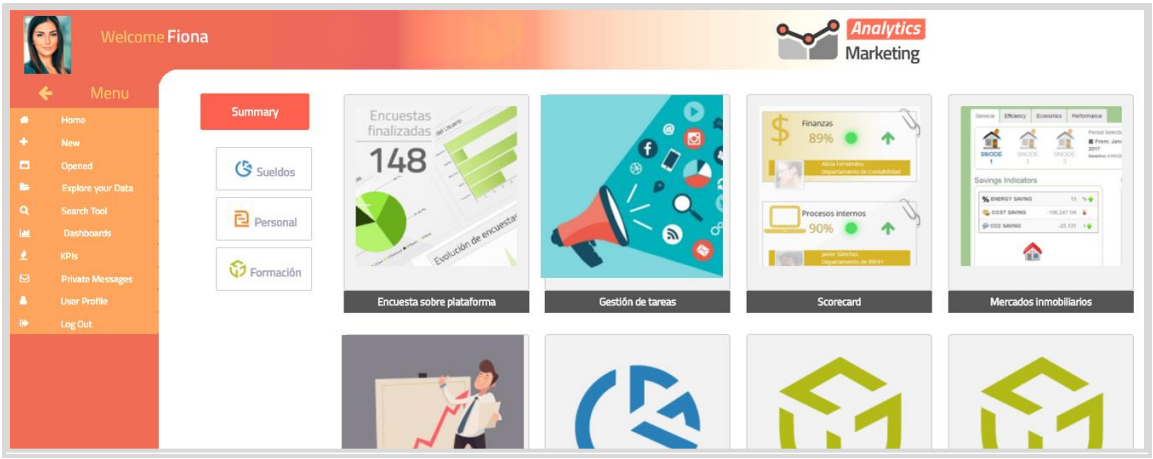



facebook DASHBOARD COMENTARIOS
Sexo
Sentimiento
Localización
Tipo de recurso
Evolución Comentarios por días
SMART MADRID
Distrito | Barrio
M30 | Urbano
Última Actualización
18:24
TOP 5 PUNTOS DE MEDIDA
PROXIMIDAD DE PUNTOS DE MEDIDA
EVOLUCIÓN INTENSIDAD DE TRÁFICO


Créditos Matriculados
Procedencia
FUENTE-ÁLAMO
Alumnos Matriculados: 50
Sexo
50% Masculino
50% Femenino
Rango de Edad
18-20
20-22
22-24
24-26
26-28
28-30
30-35
35-40
GDO/A EN BIOLOGÍA
Créditos Matriculados: 6.500
Buscar Plan de Estudio


INTERÉS PLATAFORMA
181 Inscritos en Cursos
8 Desinscritos en Cursos
181 Inicios de Cursos
110 Inscritos en Eventos Virtuales
63 Inscritos en Debates
181 Inscritos en Talleres
USUARIOS ACTIVOS POR PAÍS
INSCRITOS EN CURSOS POR COMUNIDAD EDU


JORNADA COMPLETA
IDIOMAS
TIC
TIPO DE CENTRO
TIPO DE AUSENCIA
Nº de profesores por sexo
36%
64%
Nº total de profesores
1.587
Media de edad
43 años
Búsqueda
TERRITORIAL : Madrid - Capital


Welcome Fiona
Analytics
Marketing
Menu
Home
New
Opened
Explore your Data
Search Tool
Dashboards
KPIs
Private Messages
User Profile
Log Out
Summary
Sueldos
Personal
Formación
Encuestas finalizadas
148
Encuesta sobre plataforma
Gestión de tareas
Finanzas
89%
Procesos internos
90%
Scorecard
Mercados inmobiliarios


2,963
68