

The logo for Sistel, featuring the word "Sistel" in a white, sans-serif font, enclosed within a white rectangular border. This logo is positioned on the left side of a large blue rectangular area that contains a white geometric pattern of overlapping squares and rectangles.

**Sistel**

CONSULTORÍA  
Y SERVICIOS  
INFORMÁTICOS

# **Automatización Datawarehouse en GCP - COVAP**

Marcos Remón Salazar

---

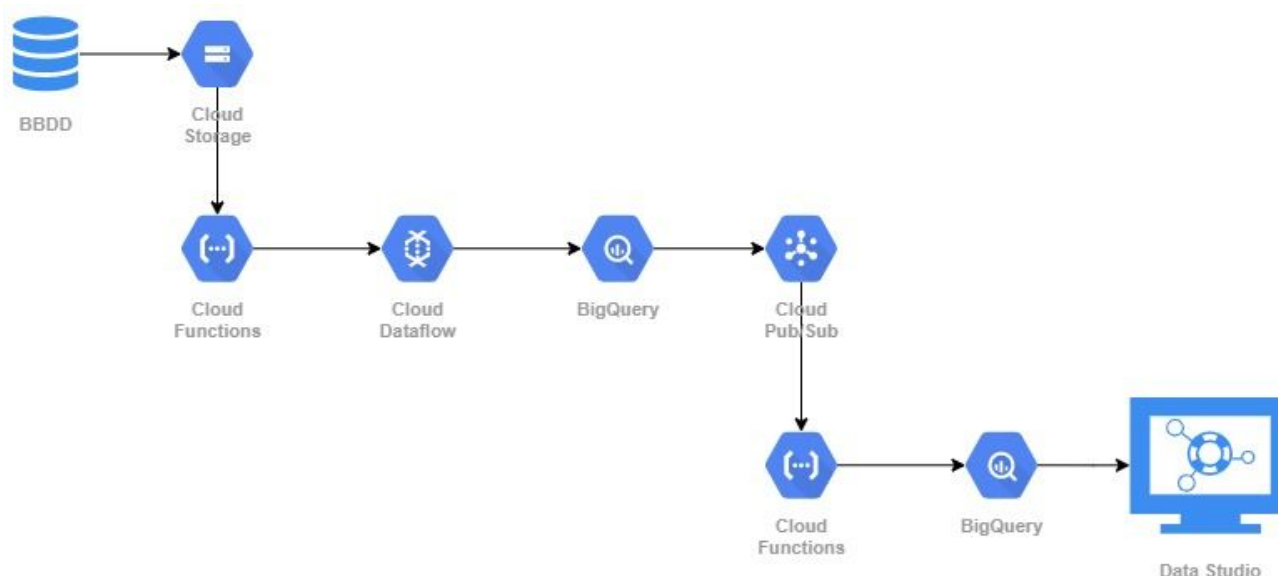
# Índice

<b>Índice</b>	<b>1</b>
<b>Estructura del proyecto</b>	<b>3</b>
Diseño	3
<b>Subida de archivos a BigQuery</b>	<b>3</b>
Cloud Storage	3
Cloud Functions + Dataflow	4
<b>BigQuery</b>	<b>6</b>
Estructura Datasets	6
Actualización Datawarehouse	6
<b>Data Studio</b>	<b>7</b>
Conexión de datos	7
Actualización de los datos	7
<b>Flujo de datos</b>	<b>8</b>
Cloud Storage	8
Dataflow	11
BigQuery	11
Data Studio	12

## Estructura del proyecto

### Diseño

El diseño de la estructura del proyecto para la automatización de subida de datos desde el servidor de Qlik a BigQuery y su posterior explotación con Data Studio es el siguiente:



## Subida de archivos a BigQuery

### Cloud Storage

En primer lugar, hay que generar un archivo bat que contiene las instrucciones de subida de archivos del servidor On-Premise a Cloud Storage.

```
gsutil -m cp D:\QlikSense\Desarrollo\QVD\Procesados\Maestros\CSV\*.csv  
gs://covap-gestion-financiera
```

```
gsutil -m cp D:\QlikSense\Desarrollo\QVD\Procesados\Hechos\CSV\*.csv  
gs://covap-gestion-financiera
```

Este archivo se encuentra en **D:\GoogleCloud\subidaCloudStorage.bat** y se ejecutará al terminar la recarga de la **app procesa\_QVD\_Ventas\_Google** de Qlik Sense.

```

C:\Users\Administrador> dir
[30/39 files][ 22.0 MiB/111.2 MiB] 19% Done
[31/39 files][ 22.0 MiB/111.2 MiB] 19% Done
[32/39 files][ 22.0 MiB/111.2 MiB] 19% Done
[33/39 files][ 25.1 MiB/111.2 MiB] 22% Done
[34/39 files][ 26.4 MiB/111.2 MiB] 23% Done
[35/39 files][ 34.0 MiB/111.2 MiB] 30% Done
[35/39 files][ 46.2 MiB/111.2 MiB] 41% Done
[35/39 files][ 58.0 MiB/111.2 MiB] 52% Done
[35/39 files][ 70.1 MiB/111.2 MiB] 63% Done
[36/39 files][ 78.1 MiB/111.2 MiB] 70% Done
[37/39 files][ 84.3 MiB/111.2 MiB] 75% Done
[38/39 files][ 89.0 MiB/111.2 MiB] 80% Done
[38/39 files][ 92.3 MiB/111.2 MiB] 83% Done
[38/39 files][ 95.4 MiB/111.2 MiB] 85% Done
[38/39 files][ 98.5 MiB/111.2 MiB] 88% Done      4.3 MiB/s ETA 00:00:03
[38/39 files][101.6 MiB/111.2 MiB] 91% Done      3.2 MiB/s ETA 00:00:03
[38/39 files][104.7 MiB/111.2 MiB] 94% Done      3.0 MiB/s ETA 00:00:02
[38/39 files][107.8 MiB/111.2 MiB] 96% Done      3.0 MiB/s ETA 00:00:01
[38/39 files][110.9 MiB/111.2 MiB] 99% Done      3.0 MiB/s ETA 00:00:00
[39/39 files][111.2 MiB/111.2 MiB] 100% Done     2.8 MiB/s ETA 00:00:00
Operation completed over 39 objects/111.2 MiB.
C:\Users\Administrador>

```

## Cloud Functions + Dataflow

Una vez subidos los archivos a Cloud Storage saltará la Cloud Function **csv-to-dataflow** que ejecuta la plantilla de Dataflow para cada archivo, para ello es necesario que en el bucket **gs://covap-gestion-financiera/schemas** esté subido un archivo json con el esquema de la tabla correspondiente.



Una vez se haya finalizado correctamente la tarea en **Dataflow**, ya estarán los archivos csv correctamente subidos a **BigQuery**.

# BigQuery

## Estructura Datasets

La estructura del proyecto **gestionfinanciera** en BigQuery es la siguiente:

- STG\_GestionFinanciera: En este Dataset se transfieren los datos de Cloud Storage a BigQuery y sin transformaciones.
- ETL\_GestionFinanciera: Están las vistas que contienen la lógica ETL que se aplica a las tablas de STG.
- DWH\_GestionFinanciera: Dataset destino para las tablas una vez se han realizado las transformaciones correspondientes.
- HECHOS\_GestionFinanciera: Este Dataset contiene el modelo de datos que utilizaremos en Data Studio.

## Actualización Datawarehouse

La actualización del Datawarehouse se hace a través de la Cloud Function **bq-stg-to-dwh** que ejecuta las vistas ETL y guarda los datos en DWH\_GestionFinanciera. También genera el modelo de datos en HECHOS\_GestionFinanciera.

# Data Studio

## Conexión de datos

La aplicación CM Ventas utiliza la tabla **gestionfinanciera.HECHOS\_GestionFinanciera.Hechos** como modelo de datos.

← EDITAR LA CONEXIÓN | FILTRAR POR CORREO ELECTRÓNICO

Campo ↓	Tipo ↓	Agregación predeterminada ↓	Descripción ↓
DIMENSIONES (61)			
Actividad	RBC Texto	Ninguna	
Fecha	Fecha	Ninguna	
mCifraVentas	123 Número	Total	
mCifraVentas_AnyoAnteri...	123 Número	Total	
mCifraVentasAnyoCY	123 Número	Total	
mCifraVentasAnyoCY_An...	123 Número	Total	
mCifraVentasAnyoLY	123 Número	Total	
mCifraVentasAnyoLY_Any...	123 Número	Total	
mCifraVentasMesCY	123 Número	Total	
mCifraVentasMesCY_Any...	123 Número	Total	
mCifraVentasMesLY	123 Número	Total	
mCifraVentasMesLY_Any...	123 Número	Total	

## Actualización de los datos

Los datos se actualizan con la tabla **gestionfinanciera.HECHOS\_GestionFinanciera.Hechos** automáticamente. Desde el cuadro también se pueden refrescar manualmente los datos.

CM Ventas

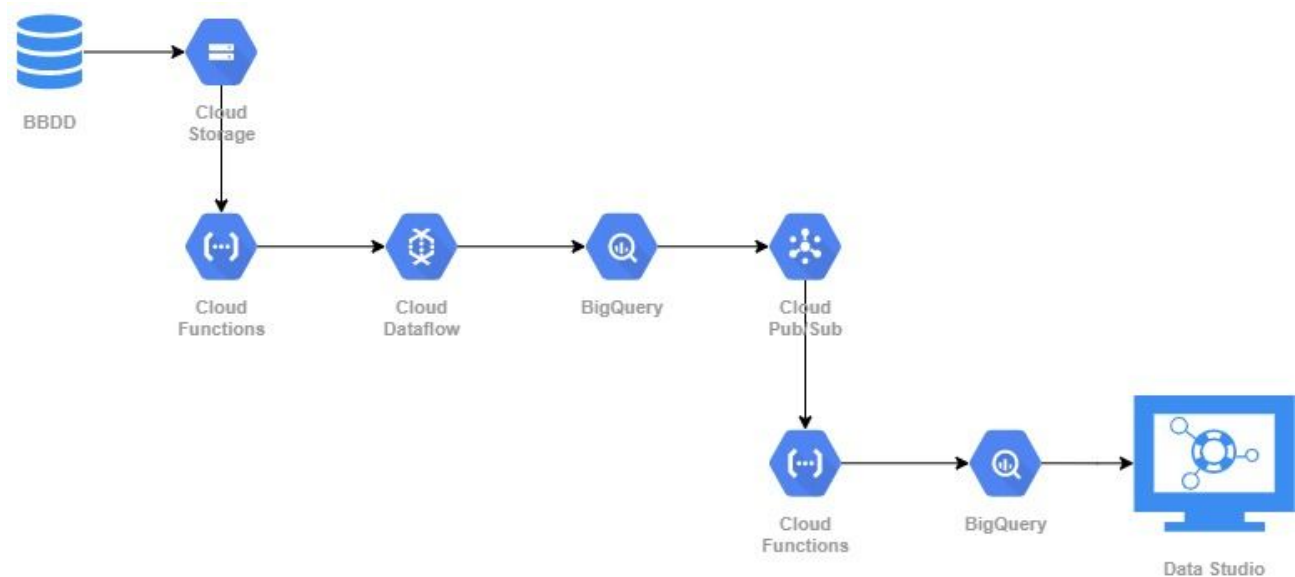
< Resumen Ventas (Página 1 de 6) >

<

En el caso del Informe Diario por Agentes la información se obtiene desde una consulta personalizada en la que filtramos los datos por Agente y partición de la tabla.



## Flujo de datos



## Cloud Storage

En primer lugar se ejecutará la tarea **procesa\_QVD\_Ventas\_Google** que lanzará los .bat **D:\GoogleCloud\subidaCloudStorage.bat** y **D:\GoogleCloud\subidaCloudStorageHechos.bat**.



Los csv subidos a **Cloud Storage** en el segmento **gs://covap-gestion-financiera** son los siguientes:

- ActividadCompras.csv
- ActividadVentas.csv
- ActividadVentas\_V2.csv
- Agentes.csv
- AgentesCompras1.csv
- AgentesCompras2.csv
- AgentesResp.csv
- Articulos.csv
- ArticulosVentas.csv
- ArticulosVentas\_V2.csv
- Cat01.csv
- Cat01\_V2.csv
- Cat02.csv
- Cat02\_V2.csv
- Cat03.csv
- Cat03\_V2.csv
- Cat04.csv
- Cat04\_V2.csv
- Cat05.csv
- Cat05\_V2.csv
- Cat06.csv

- Cat06\_V2.csv
- Cat07.csv
- Cat07\_V2.csv
- Cat08.csv
- Cat08\_V2.csv
- Clasificacion.csv
- Clasificacion\_V2.csv
- Clientes.csv
- ClientesVentas.csv
- ClientesVentas\_V2.csv
- Comisionista.csv
- Comisionistas.csv
- Companias.csv
- CompaniasVentas.csv
- CompaniasVentas\_V2.csv
- Compañías.csv
- CompañíasVentas.csv
- DireccionClientesPlusVentas.csv
- DireccionClientesPlusVentas\_V2.csv
- Direcciones.csv
- FamiliaCompras.csv
- Hechos.csv

- HechosTest.csv
- Hechos\_V2.csv
- MAPVENTAS\_CP\_Coordenadas\_V2.csv
- MAPVENTAS\_DatosGEO\_V2.csv
- MAPVENTAS\_HH0\_3W0\_45GridTable\_V2.csv
- MAPVENTAS\_HH0\_5W0\_75GridTable\_V2.csv
- MasterCalendar.csv
- MasterCalendar\_V2.csv
- Mercados.csv
- MotivoCancelacion.csv
- Provincia.csv
- Rutas.csv
- Secciones.csv
- Sectores.csv
- Sectores2.csv
- SubActividadCompras.csv
- SubActividadVentas.csv
- SubActividadVentas\_V2.csv
- SubFamiliaCompras.csv
- SubSecciones.csv
- SubSectores.csv
- UnidadesNegocio.csv

## Dataflow

Una vez subidos los archivos csv a **Cloud Storage** se activa la función **csv-to-dataflow** que ejecuta la plantilla de Dataflow y convierte los csv en tablas en BigQuery.

Por ejemplo, **ActividadVentas.csv** -> **gestionfinanciera.STG\_GestionFinanciera.ActividadVentas**

La función **csv-to-dataflow** está configurada para que cada vez que se suba un csv a **Cloud Storage** se active.

## BigQuery

En **BigQuery** el proceso de los datos va desde **STG** (datos que provienen de los csv) a **DWH** (datos preparados para explotar desde **BigQuery** o con herramientas como **Data Studio**).

Para pasar los datos de **STG** a **DWH** se generan vistas en **ETL** que contienen la lógica de las transformaciones.

El flujo es el siguiente:

**STG\_GestionFinanciera.ActividadVentas** -> **ETL\_GestionFinanciera.ActividadVentas** -> **DWH\_GestionFinanciera.ActividadVentas**

Para activar automáticamente este proceso se utiliza la función **bq-stg-to-dwh**.

Esta función ejecuta las vistas de **ETL** y guarda los resultados en **DWH**, para que se active la función se ha creado un Tema en **Pub/Sub** con el nombre **bq-Sink**, este tema recoge los logs de las tablas de **BigQuery** (En este caso la última modificación de las tablas de **STG**).

Los Logs se han filtrado desde **Logging** creando un Sumidero (en este caso **bq-jobcompleted-logs**) que recoge esa información.

```
protoPayload.methodName="jobservice.jobcompleted"  
protoPayload.serviceData.jobCompletedEvent.job.jobConfigurati  
load.destinationTable.datasetId="STG_GestionFinanciera"
```

Una vez los datos están en DWH se activa una última función **hechos\_por\_agente** que actualiza la tabla de Hechos.

### Data Studio

Data Studio se actualiza automáticamente con los datos que hay en la tabla de Hechos en **BigQuery**.