

# Caso práctico final módulo 5

Se solicita:

1. Análisis de fuentes:
  - a. Descripción global de las fuentes
  - b. Descripción en detalle de cada campo
  - c. Tipo de campo, naturaleza, cardinalidad aproximada

Los datos se obtienen de una fuente estática .csv, en la que se registran las encuestas realizadas y otros detalles relacionados con dichas encuestas.

La fuente principal donde se obtienen los datos es un archivo plano .csv, donde el delimitador es “;”.

Para realizar el análisis de los campos de la fuente, utilizo Python:

```
import pandas as pd
df = pd.read_csv('IMF_M5_Mystery_Shopping.csv', sep=';', encoding='latin-1')
```

```
a = pd.DataFrame(df)
a
```

COD_LOC	NOMBRE_LOC	CP	POBLACION	OFICINA	PROVINCIA	COD_PROY	ID_EVALUACION	Fecha de ejecucion	COD_AUDITOR	RESULTADO	TITULO_CUESTI
12	MUCHAS BOIRO	15930	Boiro	910	A CORUÑA	0267_001	1938117	16/01/2014	R29407	0,9513	Nuevo cue
17	MUCHAS CARBALLIÑO	32500	Carballiño	910	ORENSE	0267_001	1938118	10/01/2014	33466	0,8351	Nuevo cue
21	MUCHAS BUEU	36930	Bueu	910	PONTEVEDRA	0267_001	1938119	10/01/2014	29100	0,8958	Nuevo cue
31	MUCHAS VIGO	36202	Vigo	910	PONTEVEDRA	0267_001	1938120	21/01/2014	34161	0,7625	Nuevo cue
34	MUCHAS REDONDELA	36800	Redondela	910	PONTEVEDRA	0267_001	1938121	23/01/2014	34161	0,6069	Nuevo cue
...	...	...	...	...	...	...	...	...	...	...	...
266	STAND - MEDIA MARKT BARAKALDO	48000	BARACALDO	905	VIZCAYA	0457_004	2016569	16/06/2014	A32801	1	CONSUMO PROFI SUPERFIC
266	STAND - MEDIA MARKT BARAKALDO	48000	BARACALDO	905	VIZCAYA	0457_004	2016570	16/06/2014	A32801	1	CONSUMO PROFI P.
324	GRUPO BONATEL S.L.U.	12540	VILLAREAL	917	CASTELLON	0457_002	2016576	17/04/2014	31433	0,95	MYSTERY SHOPPING P1
161	AVEIRO	8000	AVEIRO	901	EXTRANJERO	0569_001	2016755	15/05/2014	33592	0,873	Prenatal Visu
161	AVEIRO	8000	AVEIRO	901	EXTRANJERO	0569_001	2016756	15/05/2014	33592	0,975	Prenatal Visu

ows × 12 columns

A pesar de lo que se observa en la imagen, Pandas (Python) reconoce la mayoría de las variables como cadena de texto, sólo reconoce a CP e ID\_EVALUACION como números:

```
a.dtypes
```

```
COD_LOC          object
NOMBRE_LOC       object
CP                int64
POBLACION        object
OFICINA          object
PROVINCIA        object
COD_PROY         object
ID_EVALUACION    int64
Fecha de ejecucion object
COD_AUDITOR      object
RESULTADO        object
TITULO_CUESTIONARIO object
dtype: object
```

La base de datos está conformada por 32.797 filas y 12 variables, con las siguientes características:

1. **COD\_LOC**: código de localización del centro al que se le ha realizado la encuesta. Esta es una variable cualitativa y no responde a ningún orden, porque existen valores numéricos y alfanuméricos.

Como se puede observar, la cardinalidad de la variable es elevada porque existen 5.646 valores únicos, donde el valor que más se repite es "1" (269 veces).

```
a['COD_LOC'].value_counts()
1      269
2      197
15     182
12     178
5      177
...
6155    1
D011    1
2317    1
2915    1
5840    1
Name: COD_LOC, Length: 5646, dtype: int64
```

```
a['COD_LOC'].describe()
count      32797
unique     5646
top         1
freq       269
Name: COD_LOC, dtype: object
```

2. **NOMBRE\_LOC**: nombre de localización del centro al que se le ha realizado la encuesta.

Se trata de una variable cualitativa. Su cardinalidad es elevada, con 6.702 valores únicos, siendo el que más se repite "LUALBO, S.L."

```
a['NOMBRE_LOC'].value_counts()
LUALBO, S.L.      872
Salesland        690
Aventel          519
GRUPCOM RETAIL, S.L.  417
INTERNITY (AVENIR TELECOM S.A.) 346
...
Pistes Vallnord Planells    1
SEVILLA 1                  1
TALLERES BURGUERA          1
Escola taquilles Cubil     1
Tienda PHONE HOUSE CC Alpo Sant Quirze (Bar) 1
Name: NOMBRE_LOC, Length: 6702, dtype: int64
```

```
a['NOMBRE_LOC'].describe()
count      32797
unique     6702
top      LUALBO, S.L
freq       872
Name: NOMBRE_LOC, dtype: object
```

3. **CP**: código postal. Es una variable cualitativa. Existen 1.797 valores únicos, el que más se repite es el código postal "28000".

```
a['CP'].value_counts()
28000    2321
8000     908
43840    233
8820     184
8028     153
...
17834     1
33947     1
12350     1
24411     1
29560     1
Name: CP, Length: 1797, dtype: int64
```

```
a['CP'].describe()
count    32797
unique    1797
top      28000
freq      2321
Name: CP, dtype: int64
```

4. **POBLACION:** Es una variable cualitativa y con alta cardinalidad. Existen 2.045 valores únicos y el que más que repite es “MADRID”.

```
a['POBLACION'].value_counts()
MADRID          1725
BARCELONA       1288
Madrid          1013
TE SIN FEE       782
Barcelona        745
...
Massalfasar      1
Funchal           1
Menorca           1
PRIEGO            1
Viana Do Castelo 1
Name: POBLACION, Length: 2045, dtype: int64
```

```
a['POBLACION'].describe()
count    32716
unique    2045
top      MADRID
freq      1725
Name: POBLACION, dtype: object
```

5. **OFICINA:** oficina desde la cual se realizó la encuesta. Es una variable cualitativa y su cardinalidad es baja.

Está conformada por 13 valores únicos. El que más se repite es la oficina “911”.

```
a['OFICINA'].value_counts()
911    9221
901    7915
915    4469
917    2392
922    2171
910    1970
905    1554
908    1314
919     710
921     640
SGS     297
999      91
959      53
Name: OFICINA, dtype: int64
```

```
a['OFICINA'].describe()
count    32797
unique     13
top       911
freq      9221
Name: OFICINA, dtype: object
```

6. **PROVINCIA:** provincia en la que se realiza la encuesta

```
a['PROVINCIA'].value_counts()
```

```
MADRID          6663
BARCELONA       5438
VALENCIA        1879
ALICANTE        1319
SEVILLA         988
ASTURIAS        941
A CORUÑA        928
TARRAGONA       874
MALAGA          859
MURCIA          793
PALMAS, LAS     716
SANTA CRUZ DE TENERIFE 648
CADIZ           624
PONTEVEDRA      607
VIZCAYA         603
GIRONA          596
BALEARES        550
LLEIDA          487
ZARAGOZA        467
CORDOBA         431
GRANADA         397
CASTELLON       362
EXTRANJERO      358
GUIPUZCOA       353
ALMERIA         317
BADAJOZ         311
NAVARRA         309
VALLADOLID      308
JAEN            298
LEON            280
CANTABRIA       270
CIUDAD REAL     251
HUELVA          238
ALBACETE        233
TOLEDO          230
ORENSE          221
LUGO            211
SALAMANCA       171
ALAVA           162
CACERES         140
BURGOS          128
RIOJA, LA       128
HUESCA          101
SEGOVIA         92
AVILA           85
CUENCA          80
PALENCIA        77
GUADALAJARA     70
ZAMORA          70
TERUEL          46
SORIA           33
MELILLA        31
CEUTA           25
Name: PROVINCIA, dtype: int64
```

```
a['PROVINCIA'].describe()
```

```
count    32797
unique      53
top      MADRID
freq      6663
Name: PROVINCIA, dtype: object
```

Es una variable categórica y cardinalidad es relativamente baja. Existen 53 valores únicos y el que más se repite es “MADRID” (6.663 veces).

7. **COD\_PROY**: código de proyecto, debido a que las encuestas responderán a las características descritas en cada proyecto.

La variable COD\_PROY es cualitativa y con cardinalidad elevada. Existen 223 proyectos, siendo el “0457\_003” el que más se repite.

```
a['COD_PROY'].value_counts()

0457_003      9633
0457_002      2158
0457_004      1414
0010_001      1269
0111_001        911
...
474_001         1
PHARMA MAR       1
CIR              1
ZOUK             1
0107_003         1
Name: COD_PROY, Length: 223, dtype: int64
```

```
a['COD_PROY'].describe()

count      32797
unique       223
top      0457_003
freq       9633
Name: COD_PROY, dtype: object
```

## 8. ID\_EVALUACION: identificación de cada encuesta.

Se trata de una variable cualitativa, por tanto, se realiza un cambio de tipo de variable, transformándola de valor numérico a cadena, ya que no tiene sentido realizar operaciones matemáticas sobre esta variable. Este cambio de tipo de variable sólo afecta al DataFrame utilizado para el análisis inicial de las variables, y no a su transformación posterior con Pentaho.

Esta es la variable con más cardinalidad, habiendo 32.737 valores únicos, igual que el número de filas.

```
a['ID_EVALUACION'].value_counts()

1966079      1
1988295      1
1987861      1
1987864      1
1987871      1
...
2007487      1
2007489      1
2007492      1
2007493      1
1966080      1
Name: ID_EVALUACION, Length: 32797, dtype: int64

a['ID_EVALUACION'].describe()

count      32797
unique      32797
top      1966079
freq       1
Name: ID_EVALUACION, dtype: int64
```

## 9. Fecha\_de\_ejecucion: Fecha de ejecución/realización de la encuesta

Fecha\_de\_ejecucion es una variable cualitativa, con cardinalidad elevada. Tiene 188 valores únicos y el valor que más se repite es 25/02/2014. La fecha más antigua es 05/03/1995 y la más reciente es 12/12/2014.

```
a['Fecha de ejecucion'].value_counts()

25/02/2014    520
19/02/2014    519
21/02/2014    503
20/02/2014    501
23/04/2014    497
...
28/06/2014      1
18/06/2014      1
23/01/2013      1
30/06/2014      1
20/03/2013      1
Name: Fecha de ejecucion, Length: 188, dtype: int64

a['Fecha de ejecucion'].describe()

count      32796
unique       188
top    25/02/2014
freq         520
Name: Fecha de ejecucion, dtype: object

a['Fecha de ejecucion'] = pd.to_datetime(a['Fecha de ejecucion'])
a['Fecha de ejecucion'].agg(['min', 'max'])

min    1995-03-05
max    2014-12-12
Name: Fecha de ejecucion, dtype: datetime64[ns]
```

## 10. COD\_AUDITOR: código del auditor que realizó la encuesta.

Es una variable cualitativa, con 1.169 valores únicos. El código de auditor que más se repite es "citi\_lop".

```
a['COD_AUDITOR'].value_counts()

citi_lop      782
R29530        467
A32809        412
R29502        410
a02401        380
...
33837          1
33974          1
32120          1
34288          1
34200          1
Name: COD_AUDITOR, Length: 1169, dtype: int64

a['COD_AUDITOR'].describe()

count      32797
unique     1169
top    citi_lop
freq         782
Name: COD_AUDITOR, dtype: object
```

## 11. RESULTADO: resultado de la encuesta.

Es una variable cuantitativa, con valores entre [0, 1]. La media de valores es 0,797175. Su cardinalidad es elevada.

```
a['RESULTADO'].describe()

count      32797.000000
mean       0.797175
std        0.275862
min        0.000000
25%        0.765300
50%        0.900000
75%        0.970000
max        1.000000
Name: RESULTADO, dtype: float64
```

```
a['RESULTADO'].value_counts()

1.0000    6174
0.0000    1743
0.0100     817
0.9500     795
0.9667     627
...
0.7426      1
0.7391      1
0.7607      1
0.5502      1
0.5436      1
Name: RESULTADO, Length: 4205, dtype: int64
```

**12. TITULO\_CUESTIONARIO:** esta es una variable cualitativa y su cardinalidad es elevada.

TITULO\_CUESTIONARIO está formada por 439 valores únicos. El valor que más se repite es: "MYSTERY SHOPPER ESPECIALISTA P6 (FEB\_14)"

```
a['TITULO_CUESTIONARIO'].describe()

count      32797
unique      439
top  MYSTERY SHOPPER ESPECIALISTA P6 (FEB_14)
freq      979
Name: TITULO_CUESTIONARIO, dtype: object
```

```
a['TITULO_CUESTIONARIO'].value_counts()

MYSTERY SHOPPER ESPECIALISTA P6 (FEB_14)    979
MYSTERY SHOPPER ESPECIALISTA P1 (ABR_14)    936
MYSTERY SHOPPER ESPECIALISTA P5 (ENE_14)    859
Areas Compra Programada_Abril_2012        847
RESUMEN ESPECIALISTAS PEC - P6 (13.14)      843
...
Cuestionario MS Perfumerías Gilgo Junio 2014    1
CONTACTO WEB - FACEBOOK                        1
TTPP NUEVA IMAGEN - P6 (13.14)                 1
Rolex                                           1
LLAOLLAO AUDITORIAS (FEB_14)                   1
Name: TITULO_CUESTIONARIO, Length: 439, dtype: int64
```

Es importante conocer la existencia de valores nulos para que no limiten el análisis de los datos. Al diseñar el Data Warehouse, durante el proceso ETL se puede notificar a la empresa la existencia de celdas de información vacía, y como en este caso no han especificado qué quieren hacer con datos, se les puede notificar para que conozcan de su existencia.

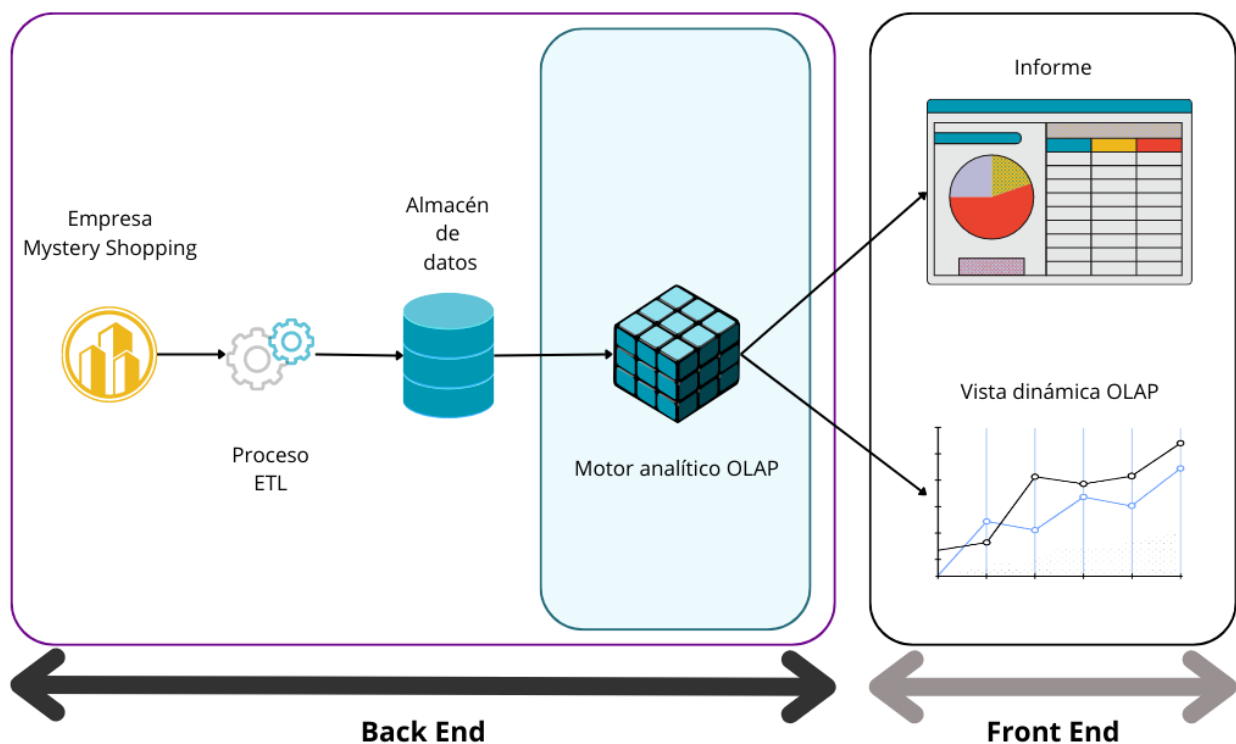
El conjunto de datos presenta 82 valores nulos, en su mayoría presentes en la variable **POBLACIÓN** (81) y un solo valor nulo en la variable **Fecha de ejecución**.

```
a.isnull().sum()
COD_LOC      0
NOMBRE_LOC   0
CP            0
POBLACION    81
OFICINA       0
PROVINCIA     0
COD_PROY     0
ID_EVALUACION 0
Fecha de ejecucion 1
COD_AUDITOR   0
RESULTADO     0
TITULO_CUESTIONARIO 0
dtype: int64
```

## 2. Análisis funcional y diagrama de arquitectura de flujo de datos

El objetivo fundamental de la compañía Mystery Shopping es poder realizar un seguimiento y analizar la información obtenida de las encuestas que ellos realizan en los centros de sus clientes.

El diagrama de arquitectura de flujo de datos es el siguiente:



### Back-End:

- **Fuente de datos:** El conjunto de datos se obtiene de fuentes internas, porque es la empresa Mystery Shopping quien realiza las encuestas en los centros de sus clientes, por tanto, considero que la misma empresa almacena la información.
- **Proceso ETL:** Incluye la extracción de los datos, su transformación y carga en el Data Warehouse. Aquí se procede a optimizar el estado de los datos, realizando (por ejemplo) un filtrado de datos nulos.



- Data Warehouse: Aquí se guardan los datos, una vez aplicado el proceso ETL, para que estén disponibles para responder las preguntas que los usuarios (departamento antifraude de Mystery Shopping) tengan sobre su negocio.
- Motor OLAP: El motor analítico OLAP se utiliza para el análisis y consultas de los datos.

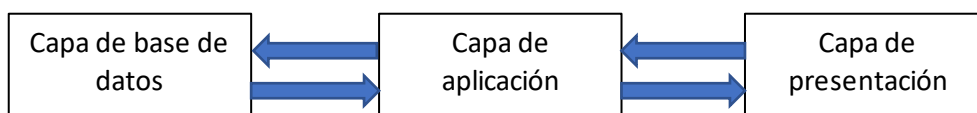
#### Front-End:

La empresa no especifica qué quiere hacer con los datos, así que propongo que puedan obtener un informe sobre los datos suministrados, en caso de que quieran tener un informe automatizado al nivel de detalle que ellos quieran.

Además, de una vista dinámica OLAP para que puedan realizar análisis de datos según su conveniencia.

#### 3. ¿Qué arquitectura de referencia usaría? Justifique la respuesta

Se realizará una arquitectura de inteligencia de negocio de tres pasos porque, además de ser el sistema de inteligencia de negocio estándar tradicional, creo conveniente que cumpla con los requisitos de la empresa de acceso a la información, porque no especifica que necesite acceder a datos actuales, sino que son datos históricos.



Si cambian sus necesidades relacionadas con el acceso y tratamiento de la información, se valorará la implementación de otro sistema de inteligencia de negocio.

#### 4. ¿Qué tecnología OLAP usaría? Justifique la respuesta

La tecnología a utilizar sería ROLAP, porque es un sistema eficaz en las consultas con alto nivel de detalle, su rendimiento es mayor con grandes volúmenes de datos, no necesita copia de datos por lo que necesita menos espacio en disco si se compara con el sistema MOLAP.

Es adecuado en los casos donde el usuario final necesite o quiera realizar consultas ad-hoc de cualquier atributo, siendo más limitado en el caso de MOLAP porque este último realiza precálculos y los guarda, siendo poco flexible para crear nuevos.

#### 5. Si se utiliza ROLAP, ¿Cuál de estos dos modelos se ajustaría mejor: el modelo en estrella o el de copo de nieve?

Modelo en estrella porque, la búsqueda de información es más sencilla a pesar de que las tablas de dimensiones sean más grandes. Además, en este caso no existen tantas variables como para tener que usar el esquema de copo de nieve, con la consiguiente relación entre tablas de dimensiones, tener que realizar joins para relacionar las tablas de dimensiones y que la consulta tarde más en ejecutarse.

6. Si se utiliza ROLAP, hay que identificar y justificar si existe algún proceso de desnormalización de información que se deba realizar

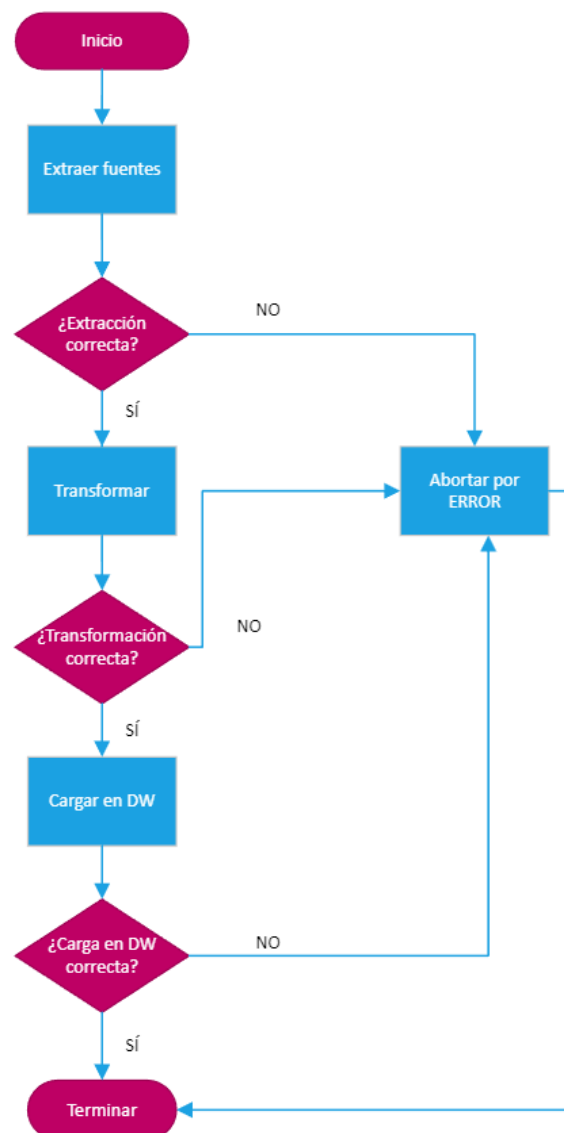
En principio, considero que no es necesario ningún proceso de desnormalización, porque existe redundancia de datos en algunas variables, por ejemplo, en localización y en el ID que identifica cada entrada.

A parte de la redundancia de datos, se pueden observar valores vacíos, lo que se puede dar al desnormalizar los datos.

Al usarse el modelo de estrella, se necesitan normalizar las distintas tablas de dimensiones, asignándoles una clave primaria y un ID.

7. Si se utiliza ROLAP, se debe incluir un diseño conceptual a modo explicativo junto con un diagrama

En el diseño conceptual que se muestra a continuación, explico cómo se van a procesar los datos suministrados.



8. Si se utiliza ROLAP, se debe incluir un diseño modelo lógico

Dimensiones:

1. Localización
2. Proyecto
3. Auditor
4. Tiempo

Hechos:

1. Resultado encuestas

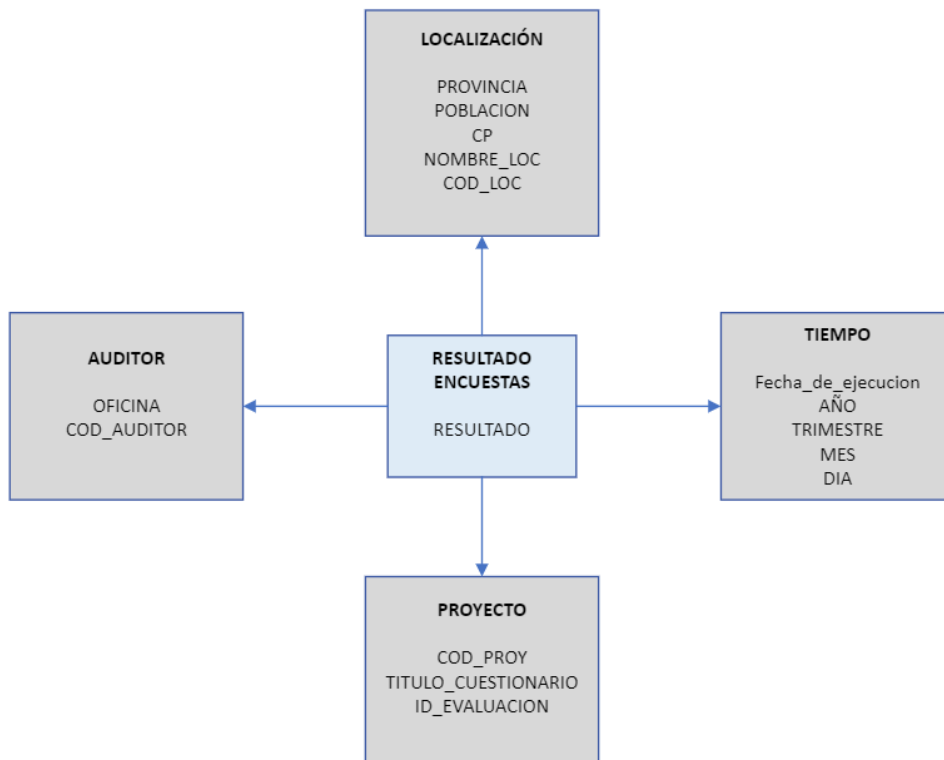
Métricas directas:

1. Resultado → Función de agregación: Promedio
2. Resultado → Función de agregación: Valor mínimo
3. Resultado → Función de agregación: Valor máximo
4. Resultado → Función de agregación: Número de cuestionarios

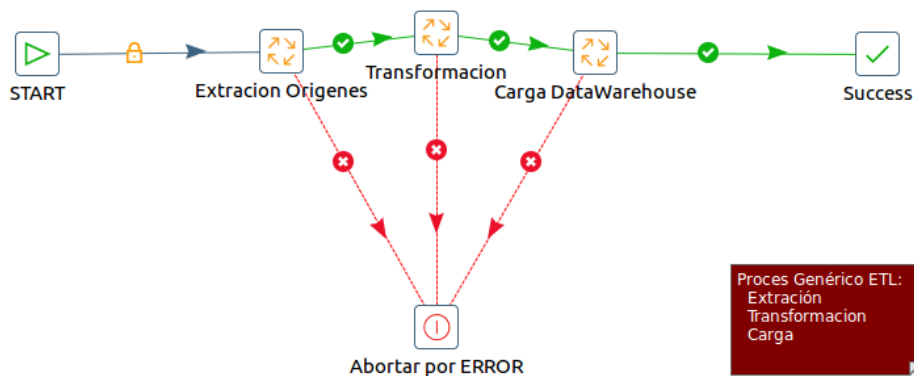
Jerarquías y niveles de jerarquía:

1. Localización:
  - 1.1. Provincia
  - 1.2. Población
  - 1.3. CP
  - 1.4. Nombre localización
  - 1.5. Código localización
2. Proyecto:
  - 2.1. Código proyecto
  - 2.2. Título cuestionario
  - 2.3. ID evaluación
3. Auditor:
  - 3.1. Oficina
  - 3.2. Código auditor
4. Tiempo:
  - 4.1. Fecha de ejecución
  - 4.2. Año
  - 4.3. Trimestre
  - 4.4. Mes
  - 4.5. Día

9. Si se utiliza ROLAP, se debe incluir un diseño modelo físico



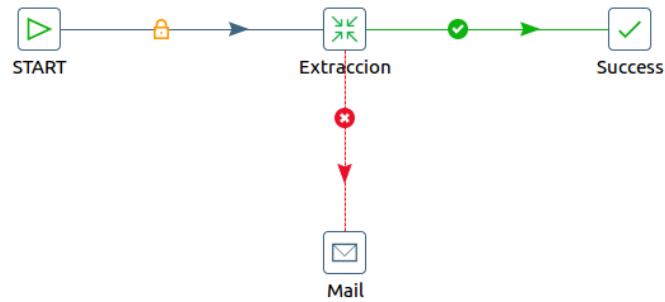
10. Realizar la implementación del proceso ETL para generar y poblar el modelo multidimensional diseñado en los apartados anteriores. Para ellos, se partirá del JOB/Trabajo global “Global\_IMF.kjb”. Para la creación del DM/DW, hay que usar la base de datos MySQL de la máquina virtual master\_imf.



El job “Global\_IMF”, además de los steps de inicio, fin y abortar la ejecución del job, incluye tres steps: Extracción Orígenes, Transformación, y Carga DataWarehouse. Estos steps enlazan con su respectivo job que se detallará a continuación.

### 1. Extracción:

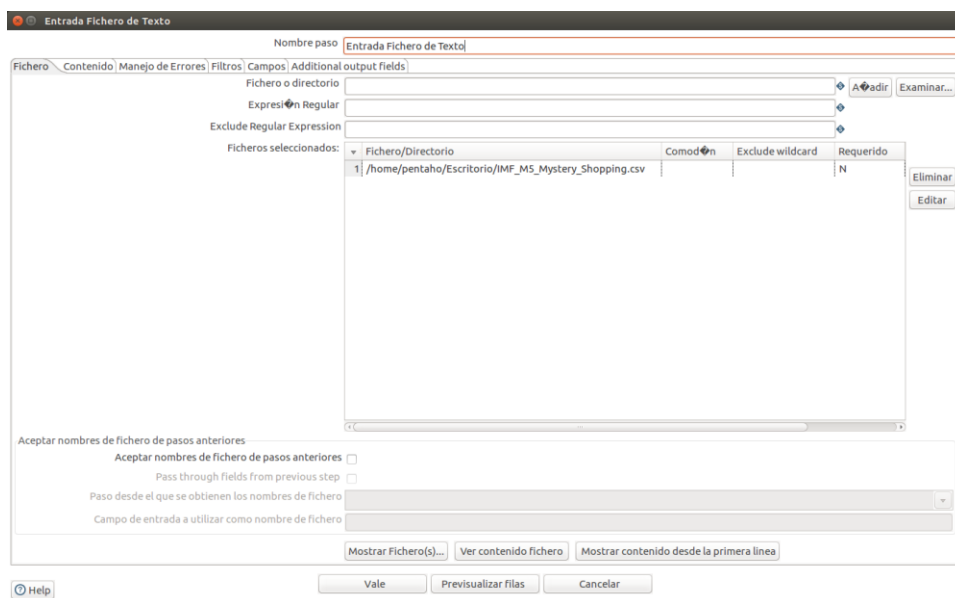
Dentro del job “Extracción Orígenes”, se encuentra la siguiente secuencia de steps:



Dentro del step “Extracción” del job “Extracción Orígenes” se encuentra la siguiente transformación:



- 1.1. El primer paso es la **entrada del fichero de texto**, se cambia el nombre del paso, se selecciona el archivo .csv con el botón **Examinar** y posteriormente se pulsa en **Añadir**.



En la pestaña Contenido, se escoge el **Formato Unix**, se vuelve a la pestaña **Fichero** y se pulsa sobre **Mostrar contenido desde la primera línea** para visualizar el contenido del archivo. Aquí se puede observar si hay cabecera, el separador de campos y el separador de textos. Una vez observado, se coloca en el separador de campos “;”.

Nombre paso: **Entrada Fichero de Texto**

Fichero | Contenido | Manejo de Errores | Filtros | Campos | Additional output fields

Tipo de fichero: **CSV**

Separador de campos: **;** Insert TAB

Separador de texto:

☐ Permitir saltos de linea en campos con separador de texto?

Escape:

Cabecera: ☒ Número de líneas de cabecera: **1**

Pie: ☐ Número de líneas de pie: **1**

☒ Líneas cortadas? ☐ Número de veces que se corta: **1**

Paginado (impresión): ☐ Número de líneas por página: **50**

Líneas en cabecera documento: **0**

Comprimido (Zip): **None**

Eliminar filas vacías: ☒

☒ Incluir nombre del fichero en salida? ☐ Campo con el nombre del fichero:

☒ Número de fila en salida? ☐ Campo con el número de fila:

☐ Número de fila por fichero? ☐

Formato: **Unix**

Codificación:

Límite: **0**

☒ Ser flexible al leer fechas?

Utilizar el formato de fecha local(e): **es\_ES**

Result filenames:

☒ Add filenames to result

Help Vale Previsualizar filas Cancelar

En la pestaña **Campo** se pulsa sobre **Traer campos**, dando como resultado 12 variables con sus características.

Nombre paso: **Entrada Fichero de Texto**

Fichero | Contenido | Manejo de Errores | Filtros | Campos | Additional output fields

Nº	Nombre	Tipo	Formato	Posición	Longitud	Precisión	Moneda	Decimal	Grupo	Nulo si	Por defecto	Tipo de poda	Repe
1	COD_LOC	String			15		€	.	-	-		ninguno	N
2	NOMBRE_LOC	String			52		€	.	-	-		ninguno	N
3	CP	Integer	#		15	0	€	.	-	-		ninguno	N
4	POBLACION	String			41		€	.	-	-		ninguno	N
5	OFICINA	String			3		€	.	-	-		ninguno	N
6	PROVINCIA	String			22		€	.	-	-		ninguno	N
7	COD_PROY	String			14		€	.	-	-		ninguno	N
8	ID_EVALUACION	Integer	#		15	0	€	.	-	-		ninguno	N
9	Fecha_de_ejecucion	Date	dd/MM/yyyy				€	.	-	-		ninguno	N
10	COD_AUDITOR	String			10		€	.	-	-		ninguno	N
11	RESULTADO	Number	##		6	4	€	.	-	-		ninguno	N
12	TITULO_CUESTIONARIO	String			50		€	.	-	-		ninguno	N

Help Vale Previsualizar filas Cancelar

Por último, se clics sobre **Previsualizar filas** para saber si los datos se leen correctamente:

Examine preview data

Rows of step: Entrada Fichero de Texto (1000 rows)

Nº	COD_LOC	NOMBRE_LOC	CP	POBLACION	OFICINA	PROVINCIA	COD_PROY	ID_EVALUACION	Fecha_de_ejecucion	COD_AUDITOR	RESULTADO	TITULO_CUESTIONARIO
12		MUCHAS BORIO	15930	Burio	910	A CORUÑA	0267_001	1938117	16/01/2014	829407		1 Nuevo cuestionario Mucha's
17		MUCHAS CARBALLIÑO	32500	Carballiño	910	ORENSE	0267_001	1938118	10/01/2014	33466	0,8	Nuevo cuestionario Mucha's
21		MUCHAS BUEU	36930	Bueu	910	PONTEVEDRA	0267_001	1938119	10/01/2014	29100	0,9	Nuevo cuestionario Mucha's
31		MUCHAS VIGO I	36202	Vigo	910	PONTEVEDRA	0267_001	1938120	21/01/2014	34161	0,8	Nuevo cuestionario Mucha's
34		MUCHAS REDONDELA	36800	Redondela	910	PONTEVEDRA	0267_001	1938121	23/01/2014	34161	0,6	Nuevo cuestionario Mucha's
37		MUCHAS VIGO II	36210	Vigo	910	PONTEVEDRA	0267_001	1938122	22/01/2014	800004	0,5	Nuevo cuestionario Mucha's
38		MUCHAS PONTEVEDRA II	36001	Pontevedra	910	PONTEVEDRA	0267_001	1938123	10/01/2014	29100	0,9	Nuevo cuestionario Mucha's
8		MUCHAS ORDENES	15680	Ordenes	910	A CORUÑA	0267_001	1938124	20/01/2014	33900	0,8	Nuevo cuestionario Mucha's
21		BCN-ROMA	8000	BARCELONA	901	BARCELONA	0377_001	1944559	13/01/2014	149	0,7	Vueling_Mystery Passenger 20
22		ROMA-BCN	8000	BARCELONA	901	BARCELONA	0377_001	1944560	13/01/2014	149	0,7	Vueling_Mystery Passenger 20
11		BARCELONA-FLORENCIA	8000	BARCELONA	901	BARCELONA	0377_001	1944561	07/01/2014	2299	0,8	Vueling_Mystery Passenger 20
12		FLORENCIA-BARCELONA	8000	BARCELONA	901	BARCELONA	0377_001	1944562	07/01/2014	2299	0,8	Vueling_Mystery Passenger 20
17		BARCELONA-LONDRES	8000	BARCELONA	901	BARCELONA	0377_001	1944563	30/01/2014	29667	0,9	Vueling_Mystery Passenger 20
14		LONDRES-BARCELONA	8000	BARCELONA	901	BARCELONA	0377_001	1944564	26/01/2014	29667	0,8	Vueling_Mystery Passenger 20
15		BARCELONA-MADRID	8000	BARCELONA	901	BARCELONA	0377_001	1944565	24/01/2014	149	0,8	Vueling_Mystery Passenger 20
16		MADRID-BARCELONA	8000	BARCELONA	901	BARCELONA	0377_001	1944566	24/01/2014	149	0,8	Vueling_Mystery Passenger 20
17		BARCELONA-MILAN	8000	BARCELONA	901	BARCELONA	0377_001	1944567	30/01/2014	33663	0,8	Vueling_Mystery Passenger 20
18		MILAN-BARCELONA	8000	BARCELONA	901	BARCELONA	0377_001	1944568	30/01/2014	33663	0,9	Vueling_Mystery Passenger 20
19		BARCELONA-LISBOA	8000	BARCELONA	901	BARCELONA	0377_001	1944569	29/01/2014	33663	0,9	Vueling_Mystery Passenger 20
20		LISBOA-BARCELONA	8000	BARCELONA	901	BARCELONA	0377_001	1944570	29/01/2014	33663	0,9	Vueling_Mystery Passenger 20
21		BCN-LA CORUÑA	8000	BARCELONA	901	BARCELONA	0377_001	1944600	15/01/2014	2299	0,8	Vueling_Mystery Passenger 20
22		LA CORUÑA-BCN	8000	BARCELONA	901	BARCELONA	0377_001	1944602	15/01/2014	2299	0,9	Vueling_Mystery Passenger 20
23	HM_300	IKEA ALCORCON	28925	Alcorcón	SGS	MADRID	1945178	16/01/2014	c_heras	0,9	AUDITORIA SEGURIDAD ALMI	
24	HM_301	IKEA ASTURIAS	33429	Paredes de Siero	SGS	ASTURIAS	1945179	29/01/2014	MAIDER_L	1	AUDITORIA SEGURIDAD ALMI	
25	HM_302	IKEA BADALONA	8917	Badalona	SGS	BARCELONA	1945180	15/01/2014	M_CASTANO	0,9	AUDITORIA SEGURIDAD ALMI	
26	HM_303	IKEA HOSPITALET	8907	Hospitalet de Llobregat	SGS	BARCELONA	1945181	20/01/2014	M_CASTANO	0,9	AUDITORIA SEGURIDAD ALMI	

Help

Vale

Previsualizar filas

Cancelar

Show Log

- 1.2. Se crea el paso **salida de tabla** y se utiliza la conexión Local\_mySql predeterminada en la máquina virtual master\_imf para almacenar la base de datos obtenida en el primer paso del ETL. Se comprueba que la conexión a la base de datos es correcta, se habilita la opción **vaciar tabla** y se clic en **SQL** para crear la tabla de destino; se clic en **Execute** para ejecutar la instrucción y por último en **Vale** para terminar la modificación del paso.

Salida de Tabla

Nombre paso: Salida Tabla

Conexión: Local\_MySql

Esquema destino:

Tabla destino: dataset\_extraccion

Tamaño de transacción (commit): 1000

Vaciar tabla: ☒

Ignorar errores de inserción: ☐

Specify database fields: ☐

Main options / Database fields

Repartir información en varias tablas: ☐

Campo de partición:

Particionar información por mes: ☒

Particionar información por días: ☐

Utilizar actualización por lotes para inserciones: ☒

El nombre de la tabla está definido en un campo?: ☐

Campo que contiene el nombre de la tabla:

Almacena el campo con el nombre de tabla: ☒

Incluye clave auto-generada: ☐

Nombre del campo clave auto-generada:

Help Vale Cancelar SQL

Simple SQL editor

SQL statements, separated by semicolon ';'

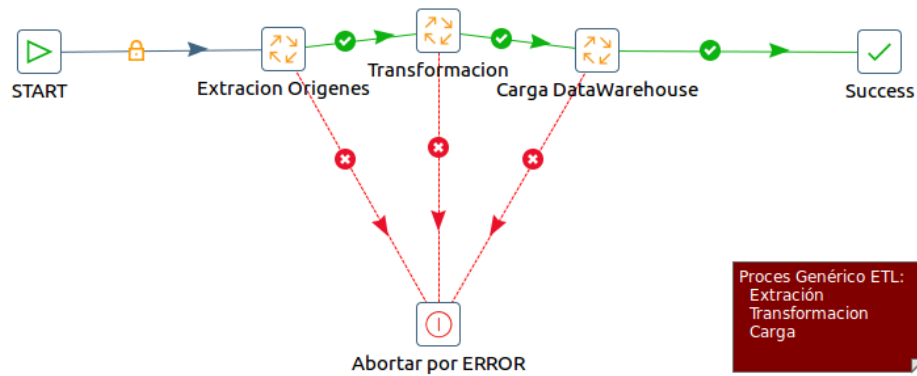
```
CREATE TABLE dataset_extraccion
(
  COD_LOC VARCHAR(15)
, NOMBRE_LOC VARCHAR(52)
, CP BIGINT
, POBLACION VARCHAR(41)
, OFICINA VARCHAR(3)
, PROVINCIA VARCHAR(22)
, COD_PROY VARCHAR(14)
, ID_EVALUACION BIGINT
, Fecha_de_ejecucion DATETIME
, COD_AUDITOR VARCHAR(10)
, RESULTADO DOUBLE
, TITULO_CUESTIONARIO VARCHAR(50)
)
;
```

Línea 1 column 0

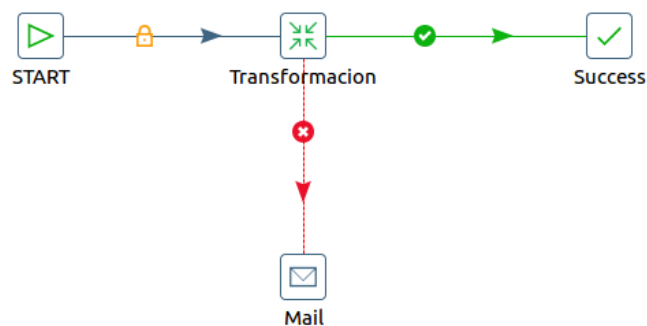
Execute Clear cache Cerrar

## 2. Transformación:

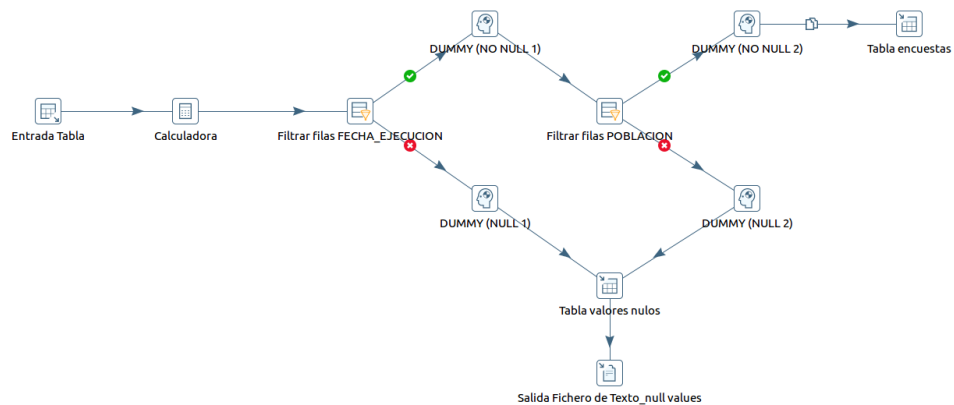
Volviendo al Job principal, se observa que el siguiente step es “Transformación”:



Job del step “Transformación”:



Transformación del step “Transformación”:



2.1. Dentro de la transformación, lo primero que se crea es el paso **Entrada Tabla** con el fin de recuperar la tabla del job “Extracción”.

Se escoge la conexión donde está guardada la base de datos y se comprueba que dicha conexión sea correcta. Clicando sobre **Obtener consulta SQL** para seleccionar la base de datos (también se puede escribir a mano la búsqueda).



**Entrada Tabla**

Nombre paso: Entrada Tabla

Conexión: Local\_MySql [Editar...] [Nuevo...] [Wizard...]

SQL: `SELECT *  
FROM dataset_extraccion` [Obtener consulta SQL...]

Line 1 Column 0

Enable lazy conversion ☐

Reemplazar variables en script? ☐

Insertar datos del paso: [v]

Ejecutar para cada fila? ☐

Limitar tamaño: 0

[Help] [Vale] [Previsualizar] [Cancelar]

Se previsualiza para ver si los datos son correctos y finalmente se clic en **Vale** para dar por terminada la modificación de este paso.

2.2. Con el step **Calculadora** se divide la variable “Fecha\_de\_ejecucion” en otras variables nuevas: AÑO, TRIMESTRE, MES y DÍA; escogiendo el cálculo necesario y la variable sobre la que se realiza el cálculo. Con este step no se pierde la variable inicial “Fecha\_de\_ejecucion”.

**Calculadora**

Nombre paso: calculadora

Campos:

	Nuevo campo	Cálculo	Campo A	Campo B	Campo C	Tipo de valor	Longitud	Precisión	Eliminar	C
1	ANNO	Año de fecha A	Fecha_de_ejecucion			String			N	
2	TRIMESTRE	Quarter of date A	Fecha_de_ejecucion			String			N	
3	MES	Mes de fecha A	Fecha_de_ejecucion			String			N	
4	DIA	Día del mes de fecha A	Fecha_de_ejecucion			String			N	

[Help] [Vale] [Cancelar]

2.3. Lo siguiente es filtrar las filas sin valores, primero escojo la variable **FECHA\_EJECUCION** enviando las filas a una u otra dummy dependiendo de si tiene valores nulos o no.

**Filtrar filas**

Nombre de paso: Filtrar filas FECHA\_EJECUCION

Enviar 'verdadero' a paso: DUMMY (NO NULL 1)

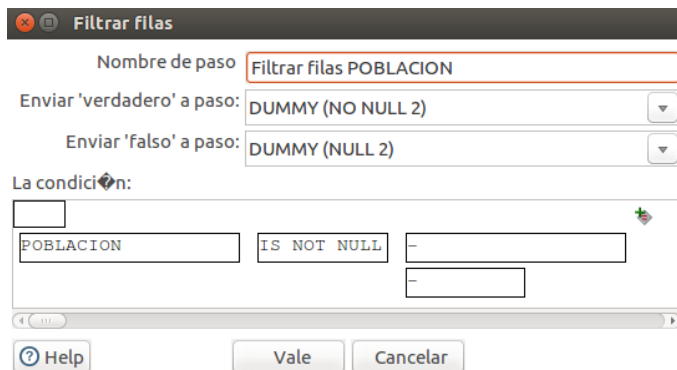
Enviar 'falso' a paso: DUMMY (NULL 1)

La condición:

[Fecha\_de\_ejecucion] IS NOT NULL [ ]

[Help] [Vale] [Cancelar]

2.4. A partir del paso **DUMMY (NO NULL 1)** se hace el filtrado de la variable **POBLACION**.



Nombre de paso: Filtrar filas POBLACION

Enviar 'verdadero' a paso: DUMMY (NO NULL 2)

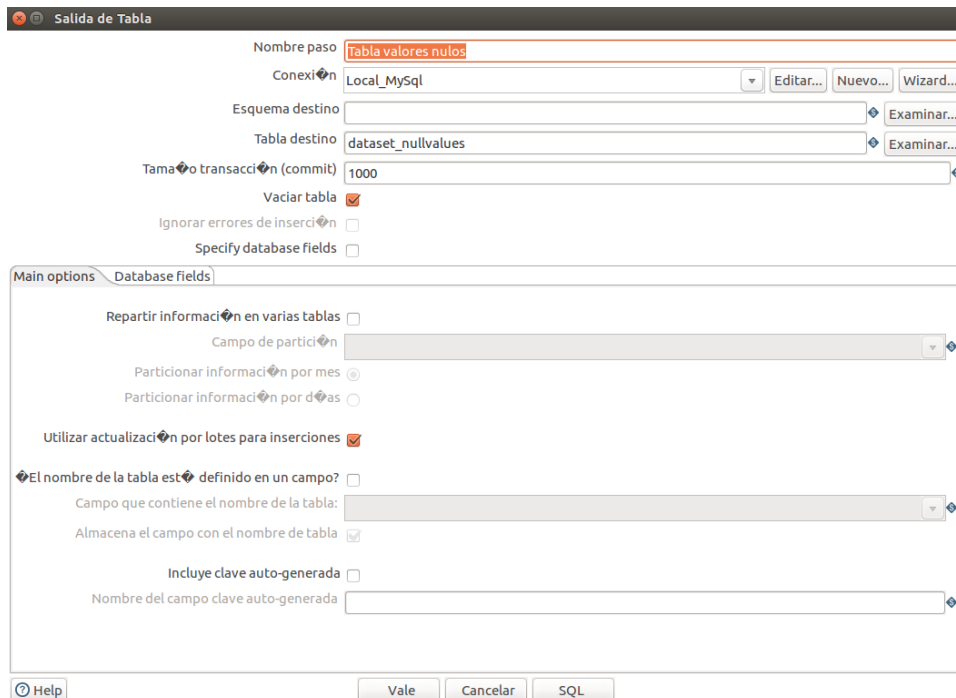
Enviar 'falso' a paso: DUMMY (NULL 2)

La condición:

POBLACION IS NOT NULL

Help Vale Cancelar

2.5. Los steps **DUMMY (NULL 1)** y **DUMMY (NULL 2)** incluyen las filas con valores nulos. Estas filas se envían al paso **Tabla valores nulos**, guardando los datos en la conexión **MySQL**.



Nombre paso: Tabla valores nulos

Conexión: Local\_MySql

Esquema destino:

Tabla destino: dataset\_nullvalues

Tamaño transacción (commit): 1000

Vaciar tabla: ☒

Ignorar errores de inserción: ☐

Specify database fields: ☐

Main options Database fields

Repartir información en varias tablas: ☐

Campo de partición:

Particionar información por mes: ☒

Particionar información por días: ☐

Utilizar actualización por lotes para inserciones: ☒

El nombre de la tabla está definido en un campo?: ☐

Campo que contiene el nombre de la tabla:

Almacena el campo con el nombre de tabla: ☒

Incluye clave auto-generada: ☐

Nombre del campo clave auto-generada:

Help Vale Cancelar SQL

Posteriormente, se guarda en un archivo .txt

Nombre paso: **Salida Fichero de Texto: null values**

Fichero | Contenido | Campos

Nombre Fichero: /home/pentaho/Escritorio/dataset\_nullvalues [Examinar...]

☐ Ejectuar como comando?

☐ Pass output to servlet

☒ Create Parent folder

☐ Do not create file at start

☐ Accept file name from field?

File name field: [ ]

Extensión: txt

☐ Incluir stepnr en nombre fichero?

☐ Incluir número partición en nombre fichero?

☐ Incluir fecha en nombre fichero?

☐ Incluir hora en nombre fichero?

☐ Specify Date time format

Date time format: [ ]

[Mostrar nombre fichero(s)...]

☒ Add filenames to result

[Help] [Vale] [Cancelar]

2.6. Las filas sin valores nulos (**DUMMY (NO NULL 2)**) se guardan en otra tabla:

Nombre paso: **Tabla encuestas**

Conexión: Local\_MySql [Editar... Nuevo... Wizard...]

Esquema destino: [ ] [Examinar...]

Tabla destino: dataset\_transformacion [Examinar...]

Tamaño transacción (commit): 1000

☒ Vaciar tabla

☐ Ignorar errores de inserción

☐ Specify database fields

Main options | Database fields

☐ Repartir información en varias tablas

Campo de partición: [ ]

☐ Particionar información por mes

☐ Particionar información por días

☒ Utilizar actualización por lotes para inserciones

☐ El nombre de la tabla está definido en un campo?

Campo que contiene el nombre de la tabla: [ ]

☒ Almacena el campo con el nombre de tabla

☐ Incluye clave auto-generada

Nombre del campo clave auto-generada: [ ]

[Help] [Vale] [Cancelar] [SQL]

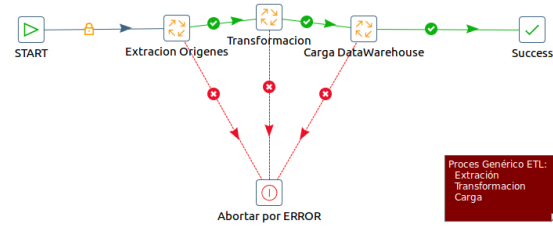
Igual que con las tablas anteriores, se verifica que la conexión con MySQL sea correcta y se clicla en la opción **vaciar tabla** y en el botón **SQL**.

El resultado del step “Transformación” del job principal es una tabla plana con los datos “limpios”, es decir, sin valores nulos y con la fecha fragmentada para poder realizar análisis en base a mes, día, año y trimestre.

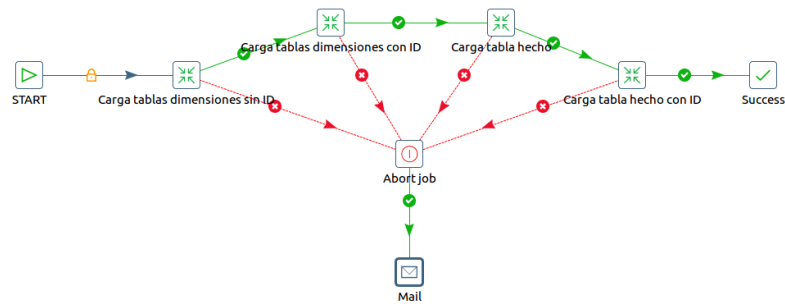
También se obtiene un archivo .txt con las filas que incluyen valores nulos para que la empresa decida qué hacer con ellos: completar la información o desecharlo.

### 3. Carga:

Recordando el job principal, el siguiente step corresponde a “Carga DataWarehouse”:

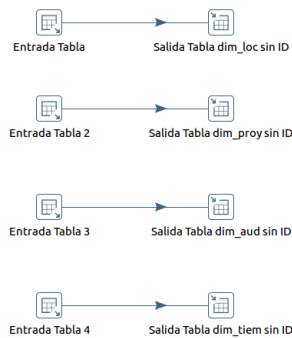


Dentro del step “Carga DataWarehouse” está el job que se detalla a continuación:

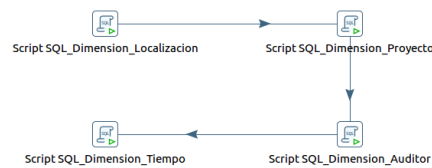


A su vez, este job se divide en cuatro steps principales:

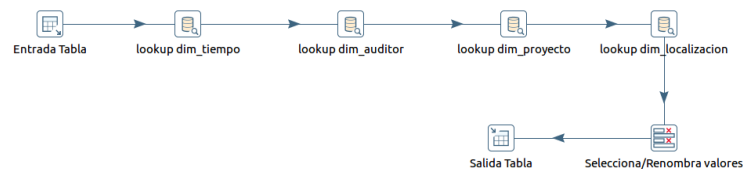
- Carga tablas dimensiones sin ID**



- Carga tablas dimensiones con ID**



- Carga tabla hecho**



- Carga tabla hecho con ID**

### 3.1. Carga tabla dimensiones sin ID: se utiliza el step “Entrada tabla” cuatro veces para elegir las variables de la tabla plana en función de su correspondiente dimensión:

The figure displays four screenshots of the 'Entrada Tabla' (Table Input) step configuration in a data tool, showing the process of loading dimension tables without primary keys. Each screenshot shows a different step (Entrada Tabla, Entrada Tabla 2, Entrada Tabla 3, Entrada Tabla 4) with specific SQL queries and settings.

**Entrada Tabla (Step 1):** The 'Nombre paso' is 'Entrada Tabla'. The 'Conexión' is 'Local\_MySql'. The SQL query is:

```
SELECT
  COD_LOC
  , NOMBRE_LOC
  , CP
  , POBLACION
  , PROVINCIA
FROM dataset_transformacion
GROUP BY
  PROVINCIA
  , CP
  , NOMBRE_LOC
  , COD_LOC
ORDER BY
  PROVINCIA
```

**Entrada Tabla 2 (Step 2):** The 'Nombre paso' is 'Entrada Tabla 2'. The 'Conexión' is 'Local\_MySql'. The SQL query is:

```
SELECT
  COD_PROY
  , ID_EVALUACION
  , TITULO_CUESTIONARIO
FROM dataset_transformacion
GROUP BY
  COD_PROY
  , TITULO_CUESTIONARIO
  , ID_EVALUACION
ORDER BY
  COD_PROY
```

**Entrada Tabla 3 (Step 3):** The 'Nombre paso' is 'Entrada Tabla 3'. The 'Conexión' is 'Local\_MySql'. The SQL query is:

```
SELECT
  OFICINA
  , COD_AUDITOR
FROM dataset_transformacion
GROUP BY
  OFICINA
  , COD_AUDITOR
ORDER BY
  OFICINA
```

**Entrada Tabla 4 (Step 4):** The 'Nombre paso' is 'Entrada Tabla 4'. The 'Conexión' is 'Local\_MySql'. The SQL query is:

```
SELECT
  ANNO
  , Fecha_de_ejecucion
  , TRIMESTRE
  , MES
  , DIA
FROM dataset_transformacion
GROUP BY
  Fecha_de_ejecucion
  , ANNO
  , TRIMESTRE
  , MES
  , DIA
ORDER BY
  Fecha_de_ejecucion
```

Each screenshot also shows the 'Line 1 Column 0' section with options like 'Enable lazy conversion', 'Reemplazar variables en', 'Insertar datos del paso', 'Ejecutar para cada fila?', and 'Limitar tamaño'.

Después de la entrada de las tablas se procede a usar el step “salida tabla”. Las tablas se almacenan sin clave primaria, siendo necesario ejecutar un script sobre cada tabla para colocar esa clave primaria y poder conectar las tablas de dimensiones con la tabla de hechos.

### 3.2. Carga tablas dimensiones con ID: En este step se ejecuta una secuencia de steps del tipo “Ejecutar sentencias SQL”, donde se ejecuta un script para cada tabla de dimensión especificando:

- Si la tabla definitiva de la dimensión existe, se debe borrar, así evitamos la duplicidad de datos al ejecutar el job del proceso ETL.
- Crear la tabla de dimensión correspondiente, tomando las variables de cada tabla de dimensión, creada en el punto 3.1.
- Crear una llave primaria para la tabla de dimensión creada.

**Ejecutar Sentencias SQL**

Nombre de paso: **Script SQL Dimension Localizacion**

Conexión: **Local\_MySql** [Editar...] [Nuevo...] [Wizard...]

Programa SQL a ejecutar. (sentencias separadas por ;) Los símbolos de interrogación serán sustituidos por argumentos.

```

DROP TABLE IF EXISTS dimension_localizacion;
CREATE TABLE dimension_localizacion AS SELECT PROVINCIA, POBLACION, CP, NOMBRE_LOC, COD_LOC FROM dim_loc;
ALTER TABLE dimension_localizacion ADD COLUMN ID_DIM_LOC SERIAL PRIMARY KEY;
SELECT ID_DIM_LOC, PROVINCIA, POBLACION, CP, NOMBRE_LOC, COD_LOC
FROM dimension_localizacion
GROUP BY ID_DIM_LOC, PROVINCIA, POBLACION, CP, NOMBRE_LOC, COD_LOC
ORDER BY ID_DIM_LOC;

```

Line 1 Column 0

☒ Ejecutar para cada fila? ☐

☐ Execute as a single statement

☐ Variable substitution

☐ Bind parameters?

☐ Quote Strings?

Parámetros:

Nombre de campo a utilizar como argumento

1

Campo con resultado de Inserción

Campo con resultado de Actualización

Campo con resultado de Eliminación

Campo con resultado de Lectura

[?] Help [Vale] [Cancelar] [Obtener campos]

**Ejecutar Sentencias SQL**

Nombre de paso: **Script SQL Dimension Proyecto**

Conexión: **Local\_MySql** [Editar...] [Nuevo...] [Wizard...]

Programa SQL a ejecutar. (sentencias separadas por ;) Los símbolos de interrogación serán sustituidos por argumentos.

```

DROP TABLE IF EXISTS dimension_proyecto;
CREATE TABLE dimension_proyecto AS SELECT COD_PROY, TITULO_CUESTIONARIO, ID_EVALUACION FROM dim_proy;
ALTER TABLE dimension_proyecto ADD COLUMN ID_DIM_PROY SERIAL PRIMARY KEY;
SELECT ID_DIM_PROY, COD_PROY, TITULO_CUESTIONARIO, ID_EVALUACION
FROM dimension_proyecto
GROUP BY ID_DIM_PROY, COD_PROY, TITULO_CUESTIONARIO, ID_EVALUACION
ORDER BY ID_DIM_PROY;

```

Line 1 Column 0

☒ Ejecutar para cada fila? ☐

☐ Execute as a single statement

☐ Variable substitution

☐ Bind parameters?

☐ Quote Strings?

Parámetros:

Nombre de campo a utilizar como argumento

1

Campo con resultado de Inserción

Campo con resultado de Actualización

Campo con resultado de Eliminación

Campo con resultado de Lectura

[?] Help [Vale] [Cancelar] [Obtener campos]

**Ejecutar Sentencias SQL**

Nombre de paso: **Script SQL Dimension Auditor**

Conexión: **Local\_MySql** [Editar...] [Nuevo...] [Wizard...]

Programa SQL a ejecutar. (sentencias separadas por ;) Los símbolos de interrogación serán sustituidos por

```

DROP TABLE IF EXISTS dimension_auditor;
CREATE TABLE dimension_auditor AS SELECT OFICINA, COD_AUDITOR FROM dim_aud;
ALTER TABLE dimension_auditor ADD COLUMN ID_DIM_AUD SERIAL PRIMARY KEY;
SELECT ID_DIM_AUD, OFICINA, COD_AUDITOR
FROM dimension_auditor
GROUP BY ID_DIM_AUD, OFICINA, COD_AUDITOR
ORDER BY ID_DIM_AUD;

```

Line 1 Column 0

☒ Ejecutar para cada fila? ☐

☐ Execute as a single statement

☐ Variable substitution

☐ Bind parameters? ☐

☐ Quote Strings? ☐

Parámetros:

Nombre de campo a utilizar como

1

Campo con resultado de Inserción

Campo con resultado de Actualización

Campo con resultado de Eliminación

Campo con resultado de Lectura

[?] Help [Vale] [Cancelar] [Obtener campos]

**Ejecutar Sentencias SQL**

Nombre de paso: **Script SQL Dimension Tiempo**

Conexión: **Local\_MySql** [Editar...] [Nuevo...] [Wizard...]

Programa SQL a ejecutar. (sentencias separadas por ;) Los símbolos de interrogación serán sustituidos por argumentos.

```

DROP TABLE IF EXISTS dimension_tiempo;
CREATE TABLE dimension_tiempo AS SELECT Fecha_de_ejecucion, ANNO, TRIMESTRE, MES, DIA FROM dim_tiem;
ALTER TABLE dimension_tiempo ADD COLUMN ID_DIM_TIEM SERIAL PRIMARY KEY;
SELECT ID_DIM_TIEM, Fecha_de_ejecucion, ANNO, TRIMESTRE, MES, DIA
FROM dimension_tiempo
GROUP BY ID_DIM_TIEM, Fecha_de_ejecucion, ANNO, TRIMESTRE, MES, DIA
ORDER BY ID_DIM_TIEM;

```

Line 1 Column 0

☒ Ejecutar para cada fila? ☐

☐ Execute as a single statement

☐ Variable substitution

☐ Bind parameters? ☐

☐ Quote Strings? ☐

Parámetros:

Nombre de campo a utilizar como argumento

1

Campo con resultado de Inserción

Campo con resultado de Actualización

Campo con resultado de Eliminación

Campo con resultado de Lectura

[?] Help [Vale] [Cancelar] [Obtener campos]

**3.3. Carga tabla hecho:** aquí se procede a crear la tabla de hechos sin ID. En resumen, se da de entrada a la tabla plana de datos “limpios” para compararlo con las tablas de dimensiones, buscando las variables que corresponden tanto en la tabla de dimensión como en la tabla plana y dando como resultado el ID de la tabla de dimensión; esto se hará con todas las tablas de dimensiones.

El objetivo de esta secuencia de steps es establecer una relación entre los ID de las tablas de dimensiones para obtener una única tabla donde aparezca dicha relación.

El primer paso es traer la tabla plana de datos “limpios”.

Nombre paso: **Entrada Tabla**

Conexión: **Local\_MySql** [Editar...] [Nuevo...] [Wizard...]

SQL: `SELECT  
COD_LOC  
, NOMBRE_LOC  
, CP  
, POBLACION  
, OFICINA  
, PROVINCIA  
, COD_PROY  
, ID_EVALUACION  
, COD_AUDITOR  
, RESULTADO  
, TITULO_CUESTIONARIO  
, ANNO  
, Fecha_de_ejecucion  
, TRIMESTRE  
, MES  
, DIA  
FROM dataset_transformacion`

Line 1 Column 0

Enable lazy conversion ☐

Reemplazar variables en script? ☒

Insertar datos del paso [dropdown]

Ejecutar para cada fila? ☐

Limitar tamaño: 0

[Help] [Vale] [Previsualizar] [Cancelar]

Lo siguiente es insertar el paso “**Búsqueda de valor en Base de Datos**”, será así con cada dimensión, conectando con Local\_MySql, insertando las variables que coinciden en la tabla de dimensiones y en la tabla plana, y devolviendo la variable ID correspondiente a cada tabla de dimensión.

Nombre paso: **lookup dim tiempo**

Conexión: **Local\_MySql** [Editar...] [Nuevo...] [Wizard...]

Esquema de búsqueda: [dropdown] [Examinar...]

Tabla de búsqueda: **dimension\_tiempo** [Examinar...]

Habilitar cache? ☐

Tamaño de cache en filas (0=todas): 0

Load all data from table ☐

La clave(s) para realizar búsqueda de valor(es):

	Campo de tabla	Comparador	Campo1	Campo2
1	Fecha_de_ejecucion	=	Fecha_de_ejecucion	
2	ANNO	=	ANNO	
3	MES	=	MES	
4	DIA	=	DIA	

Valores a devolver de la tabla de búsqueda:

	Campo	Nuevo nombre	Defecto	Tipo
1	ID_DIM_TIEM			Integer

No procesar la fila si la búsqueda falla ☐

Producir error si se obtienen múltiples resultados? ☐

Ordenar por: [dropdown]

[Help] [Vale] [Cancelar] [Obtener Campos] [Obtener Campos Búsqueda]



**Búsqueda de Valor en Base de Datos**

Nombre paso

Conexión

Esquema de búsqueda

Tabla de búsqueda

☒ Habilitar cache? ☐

Tamaño de cache en filas (0=todas)

Load all data from table ☐

La clave(s) para realizar búsqueda de valor(es):

	Campo de tabla	Comparador	Campo1	Campo2
1	OFICINA	=	OFICINA	
2	COD_AUDITOR	=	COD_AUDITOR	

Valores a devolver de la tabla de búsqueda:

	Campo	Nuevo nombre	Defecto	Tipo
1	ID_DIM_AUD			Integer

☐ No procesar la fila si la búsqueda falla

☒ Producir error si se obtienen múltiples resultados?

Ordenar por

**Búsqueda de Valor en Base de Datos**

Nombre paso

Conexión

Esquema de búsqueda

Tabla de búsqueda

☒ Habilitar cache? ☐

Tamaño de cache en filas (0=todas)

Load all data from table ☐

La clave(s) para realizar búsqueda de valor(es):

	Campo de tabla	Comparador	Campo1	Campo2
1	ID_EVALUACION	=	ID_EVALUACION	
2	COD_PROY	=	COD_PROY	
3	TITULO_CUESTIONARIO		TITULO_CUESTIONARIO	

Valores a devolver de la tabla de búsqueda:

	Campo	Nuevo nombre	Defecto	Tipo
1	ID_DIM_PROY			Integer

☐ No procesar la fila si la búsqueda falla

☒ Producir error si se obtienen múltiples resultados?

Ordenar por

**Búsqueda de Valor en Base de Datos**

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☐ Habilitar cache?

Tamaño de cache en filas (0=todas):

☐ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

	Campo de tabla	Comparador	Campo1	Campo2
1	PROVINCIA	=	PROVINCIA	
2	POBLACION	=	POBLACION	
3	CP	=	CP	
4	NOMBRE_LOC	=	NOMBRE_LOC	
5	COD_LOC	=	COD_LOC	

Valores a devolver de la tabla de búsqueda:

	Campo	Nuevo nombre	Defecto	Tipo
1	ID_DIM_LOC			Integer

☐ No procesar la fila si la búsqueda falla

☐ Producir error si se obtienen múltiples resultados?

Ordenar por:

Después de realizar la búsqueda, se utiliza el step “selecciona/renombra valores” para seleccionar las variables ID de las dimensiones, porque en caso de no usar este step, la tabla de salida sería igual que la tabla plana, y lo que queremos es la tabla de hechos con los ID de las dimensiones.

**Selecciona/Renombrar valores**

Nombre paso:

Selección & Modificar

Campos:

	Nombre campo	Renombrar a	Longitud	Precisión
1	ID_DIM_TIEM			
2	ID_DIM_AUD			
3	ID_DIM_PROY			
4	ID_DIM_LOC			
5	RESULTADO			

☐ Include unspecified fields, ordered by name

Por último, se selecciona el step “salida tabla” para almacenar la tabla resultante, se escoge la conexión, la tabla destino, seleccionar variar tabla para que no se añadan datos repetidos, se clic en SQL para ejecutar la sentencia que permita crear la tabla con los ID de las dimensiones y la variable “RESULTADO”, y se clic en Vale.

**Salida de Tabla**

Nombre paso:

Conexión:

Esquema destino:

Tabla destino:

Tamaño transacción (commit):

Vaciar tabla: ☒

Ignorar errores de inserción: ☐

Specify database fields: ☐

---

Main options Database fields

Repartir información en varias tablas: ☐

Campo de partición:

Particionar información por mes: ☐

Particionar información por días: ☐

Utilizar actualización por lotes para inserciones: ☒

El nombre de la tabla está definido en un campo: ☐

Campo que contiene el nombre de la tabla:

Almacena el campo con el nombre de tabla: ☒

Incluye clave auto-generada: ☐

Nombre del campo clave auto-generada:

#### 3.4. Carga tabla hecho con ID: en esta transformación hay un único step “Ejecutar Sentencia SQL”, especificando:

- Eliminar la tabla de hechos si ya está creada, para evitar duplicidad de datos.
- Crear la tabla de hechos, seleccionando la tabla resultante en el punto 3.4.
- Crear una llave primaria en la tabla de hechos.

**Ejecutar Sentencias SQL**

Nombre de paso:

Conexión:

Programa SQL a ejecutar. (sentencias separadas por ; ) Los símbolos de interrogación serán sustituidos por argumentos.

```

DROP TABLE IF EXISTS hecho_resultado;
CREATE TABLE hecho_resultado AS SELECT ID_DIM_TIEM, ID_DIM_AUD, ID_DIM_PROY, ID_DIM_LOC, RESULTADO FROM h_result;

ALTER TABLE hecho_resultado
ADD COLUMN ID_H_RESULTADO SERIAL PRIMARY KEY;

```

Line 1 Column 0

Ejecutar para cada fila? ☐

Execute as a single statement: ☐

Variable substitution: ☐

Bind parameters? ☐

Quote Strings? ☐

Parámetros:

Nombre de campo a utilizar como argumento

1

Campo con resultado de Inserción:

Campo con resultado de Actualización:

Campo con resultado de Eliminación:

Campo con resultado de Lectura:

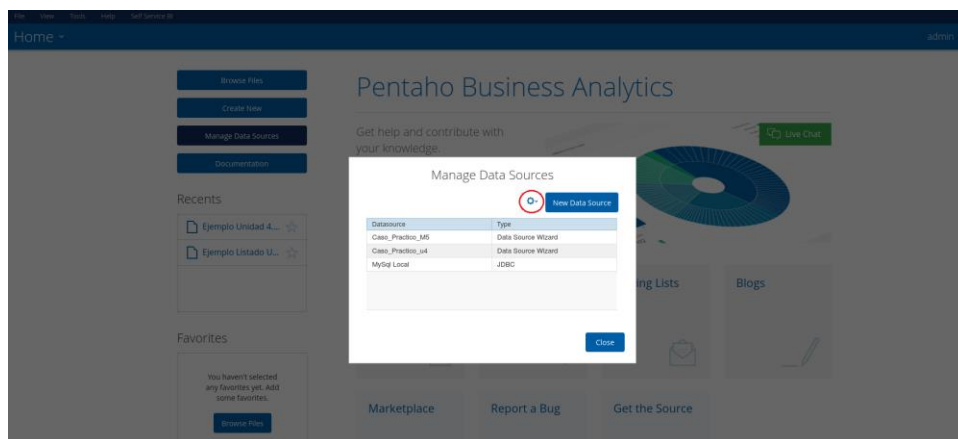
El resultado del step “Carga” del job principal es crear las tablas de dimensiones y la tabla de hechos. A pesar de que se han realizado modificaciones de tablas (al crear llaves primarias), es un proceso que, según mi entender, sería más apropiado realizarlo de esta manera.

11. Implementación del modelo multidimensional diseñado mediante los puntos anteriores. Se debe realizar con la herramienta wizard facilitada y mostrada en videos anteriores.

- 1) Se accede a la PUC de Pentaho. Una vez dentro, para crear el modelo multidimensional hay que buscar la fuente de datos, para ello, hay que acceder a **Manage Data Sources**.



- 2) Una vez dentro, se clicla sobre **New Data Source**. (cambiar captura pantalla, quitando CASO PRACTICO FINAL)



- 3) Ahora, hay que darle un nombre a la nueva fuente de datos, escoger el tipo fuente que será **Database Table(s)**, la conexión será **MySQL Local** porque es la única que se encuentra disponible y, donde están almacenadas las tablas de dimensiones y la tabla de hecho. Clicar en Next.

Data Source Wizard

Select Source Type

Select Tables

Define Joins

Data Source Name:  
Caso\_Practico\_M5

Source Type:  
Database Table(s)

Select a database connection and click Next to choose from a list of the available database tables.

Connection:  
MySQL Local

Create data source for:

☐ Reporting only

☒ Reporting and Analysis (Requires Star Schema)

- 4) En la siguiente pantalla hay que elegir dónde están almacenadas las tablas, y pasar a la derecha las tablas de dimensiones y la tabla de hechos. En la parte inferior hay que designar la tabla de hechos, y clicar en Next.

Data Source Wizard

Select Source Type

Select Tables

Define Joins

Select one table to finish or select multiple tables and click Next to define their joins.

Schema:  
master\_imf

Available Tables:

- 'master\_imf'. 'data\_quality\_ok'
- 'master\_imf'. 'dataset\_carga'
- 'master\_imf'. 'dataset\_extraccion'
- 'master\_imf'. 'dataset\_nullvalues'
- 'master\_imf'. 'dataset\_transform'
- 'master\_imf'. 'dataset\_transformacion2'
- 'master\_imf'. 'dataset\_transformacion'
- 'master\_imf'. 'dim\_aud'
- 'master\_imf'. 'dim\_cuestionario'
- 'master\_imf'. 'dim\_evaluador'
- 'master\_imf'. 'dim\_loc'
- 'master\_imf'. 'dim\_proy'
- 'master\_imf'. 'dim\_tiem'
- 'master\_imf'. 'dim\_tiempo\_full'
- 'master\_imf'. 'dim\_ubicacion'
- 'master\_imf'. 'ds\_origen\_ms'
- 'master\_imf'. 'dv\_datoslimpios\_ms'
- 'master\_imf'. 'h\_result'
- 'master\_imf'. 'h\_resultados'
- 'master\_imf'. 'municipios'

Selected Tables:

- 'master\_imf'. 'dimension\_localizacion'
- 'master\_imf'. 'dimension\_proyecto'
- 'master\_imf'. 'dimension\_auditor'
- 'master\_imf'. 'dimension\_tiempo'
- 'master\_imf'. 'hecho\_resultado'

Fact Table:  
'master\_imf'. 'hecho\_resultado'

- 5) En la última pantalla de la creación de la fuente de datos hay que determinar la relación entre la tabla de hechos y las tablas de dimensiones a partir de joins. A la izquierda está la tabla de hecho, a la derecha la tabla de dimensión, para cada tabla de izquierda y derecha

hay que elegir la variable con la que se relacionarán las distintas tablas, siendo los IDs que se crearon en la fase de carga. Una vez localizados los IDs, se clicla en **Create join** para crear el punto de unión entre ambas tablas; con todos los joins realizados, se clicla en **Finish**.

Data Source Wizard

Select Source Type Define how the tables join to each other. All tables must have at least one join defined.

Select Tables

Define Joins

Left Table: 'master\_imf'. 'hecho\_resultado'

Right Table: 'master\_imf'. 'dimension\_tiempo'

Key Field:

ID\_DIM\_TIEM

Fecha\_de\_ejecucion

ANNO

TRIMESTRE

MES

DIA

ID\_DIM\_TIEM

Join(s):

'master\_imf'. 'hecho\_resultado'. ID\_DIM\_TIEM - INNER JOIN - 'master\_imf'. 'dimension\_tie

Create Join

Delete Join

< Back Next >

Finish Cancel

- 6) Una vez creada la fuente de datos, se clicla en **Clear Model** porque el modelo multidimensional que creó Pentaho no se ajusta a lo que necesito.

Data Source Model Editor

Available

Analysis Reporting Properties

Measures

Dimensions

Dimension auditor

Dimension localizacion

Dimension proyecto

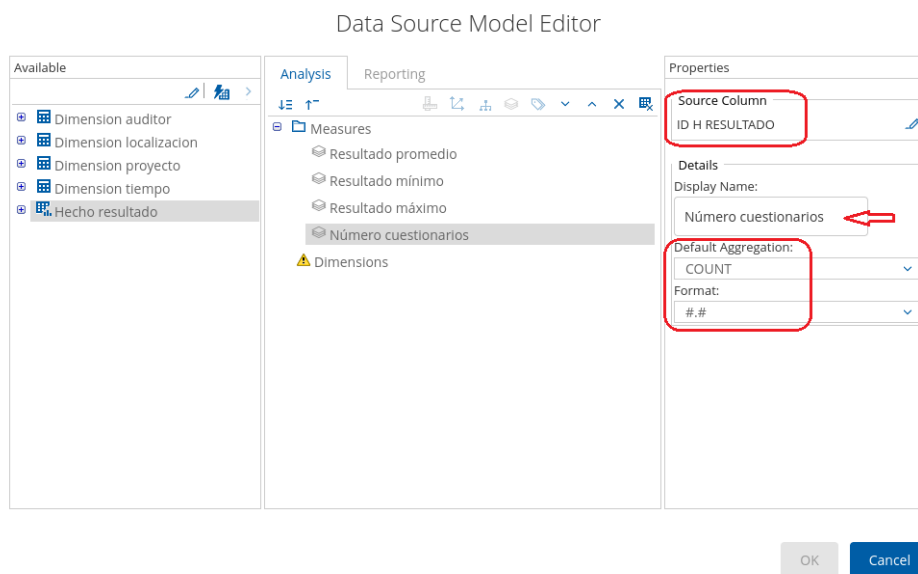
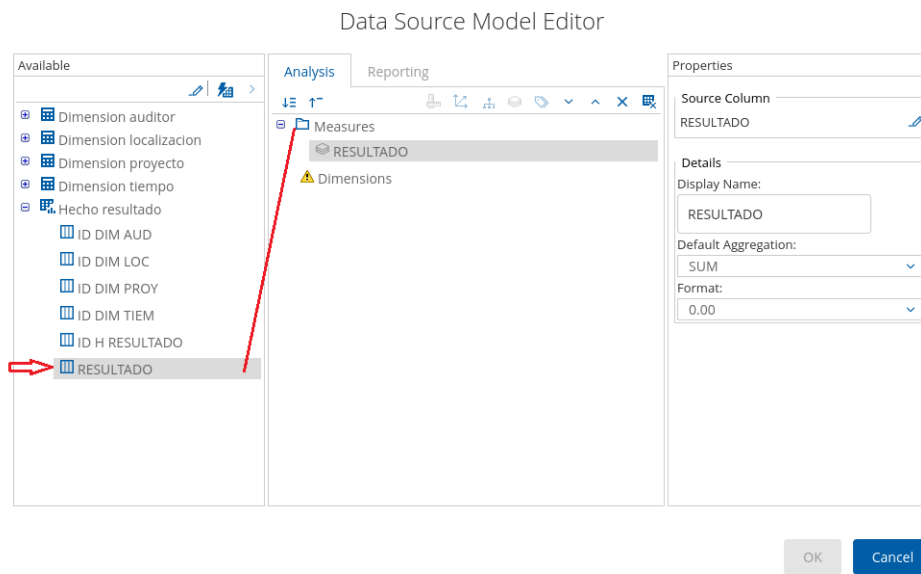
Dimension tiempo

Clear Model

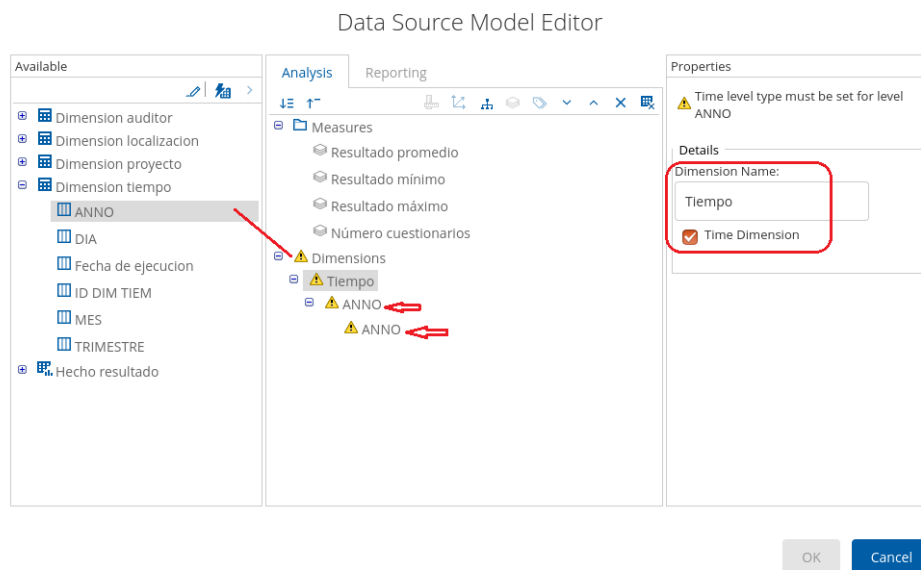
OK Cancel

- 7) Después de borrar el modelo preestablecido, comenzamos llevando la variable RESULTADO al apartado MEASURES, se cambia el nombre, la función de agregación que queramos y el

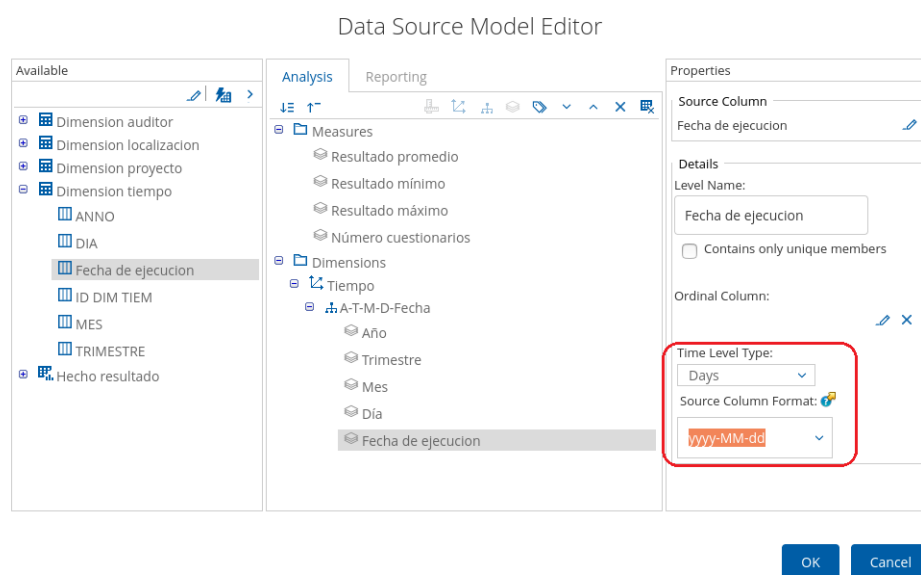
formato. Se repite el proceso hasta obtener todas las funciones de agregación establecidas en el modelo lógico.



- 8) Ahora es el momento de las dimensiones. La primera es Tiempo, se toma la variable AÑO, se especifica como dimensión temporal (**Time Dimension**), se renombra la dimensión como TIEMPO, se crea una jerarquía 5 niveles y se renombra como A-T-M-D-Fecha.



Para cada nivel, se llevan las variables en orden de jerarquía, especificando el nombre de nivel, el tipo de nivel y el formato.



- Se repite el proceso del punto anterior con el resto de dimensiones, con la diferencia de que no se marcará como una dimensión temporal.



## Data Source Model Editor

Available	Analysis	Reporting	Properties
<ul style="list-style-type: none"><li>Dimension auditor</li><li>Dimension localizacion<ul style="list-style-type: none"><li>COD LOC</li><li>CP</li><li>ID DIM LOC</li><li>NOMBRE LOC</li><li>POBLACION</li><li>PROVINCIA</li></ul></li><li>Dimension proyecto</li><li>Dimension tiempo</li><li>Hecho resultado</li></ul>	<ul style="list-style-type: none"><li>Measures<ul style="list-style-type: none"><li>Resultado promedio</li><li>Resultado mínimo</li><li>Resultado máximo</li><li>Número cuestionarios</li></ul></li><li>Dimensions<ul style="list-style-type: none"><li>Tiempo</li><li>Auditor<ul style="list-style-type: none"><li>Oficina-Auditor<ul style="list-style-type: none"><li>Oficina</li></ul></li><li>Auditor</li></ul></li><li>Localización</li></ul></li></ul>	<div>Source Column OFICINA</div> <div>Details Level Name: Oficina <input type="checkbox"/> Contains only unique members</div> <div>Ordinal Column:</div> <div>Geography Type: None</div>	

OK

Cancel

## Data Source Model Editor

Available	Analysis	Reporting	Properties
<ul style="list-style-type: none"><li>Dimension auditor</li><li>Dimension localizacion<ul style="list-style-type: none"><li>COD LOC</li><li>CP</li><li>ID DIM LOC</li><li>NOMBRE LOC</li><li>POBLACION</li><li>PROVINCIA</li></ul></li><li>Dimension proyecto</li><li>Dimension tiempo</li><li>Hecho resultado</li></ul>	<ul style="list-style-type: none"><li>Measures</li><li>Dimensions<ul style="list-style-type: none"><li>Tiempo</li><li>Auditor</li><li>Localización<ul style="list-style-type: none"><li>P-P-CP-L-CODL<ul style="list-style-type: none"><li>Provincia</li><li>Población</li></ul></li><li>CP</li><li>Nombre Localización</li><li>Código Localización</li></ul></li></ul></li></ul>	<div>Source Column POBLACION</div> <div>Details Level Name: Población <input type="checkbox"/> Contains only unique members</div> <div>Ordinal Column:</div> <div>Geography Type: City</div>	

OK

Cancel

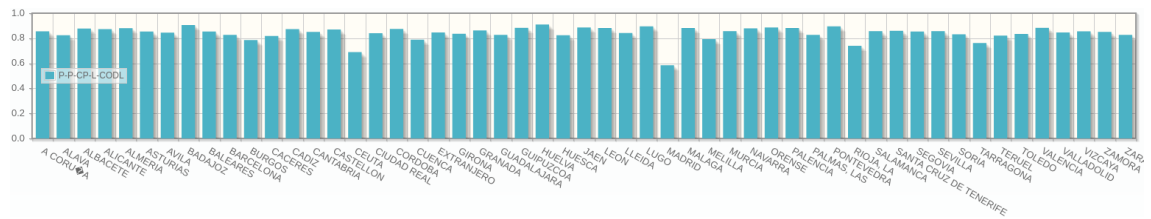
## Data Source Model Editor

Available	Analysis	Reporting	Properties
<ul style="list-style-type: none"><li>Dimension auditor</li><li>Dimension localizacion</li><li>Dimension proyecto</li><li>Dimension tiempo</li><li>Hecho resultado</li></ul>	<ul style="list-style-type: none"><li>Measures<ul style="list-style-type: none"><li>Resultado promedio</li><li>Resultado mínimo</li><li>Resultado máximo</li><li>Número cuestionarios</li></ul></li><li>Dimensions<ul style="list-style-type: none"><li>Tiempo<ul style="list-style-type: none"><li>A-T-M-D-Fecha</li></ul></li><li>Auditor<ul style="list-style-type: none"><li>Oficina-Auditor</li></ul></li><li>Localización<ul style="list-style-type: none"><li>P-P-CP-L-CODL</li></ul></li><li>Proyecto<ul style="list-style-type: none"><li>CODP-T-IDEV</li></ul></li></ul></li></ul>	<div>Details Hierarchy Name: CODP-T-IDEV</div>	

OK

Cancel

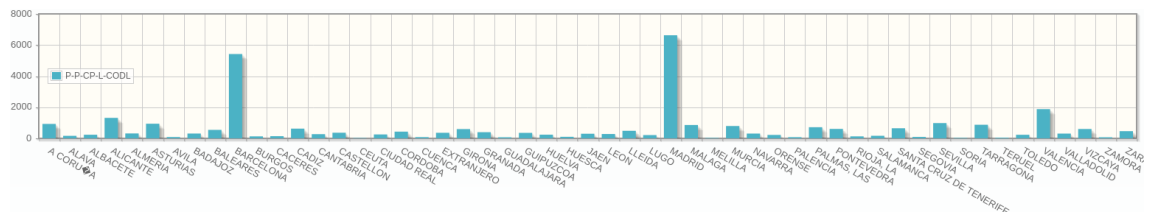
Resultado promedio por año (2014):



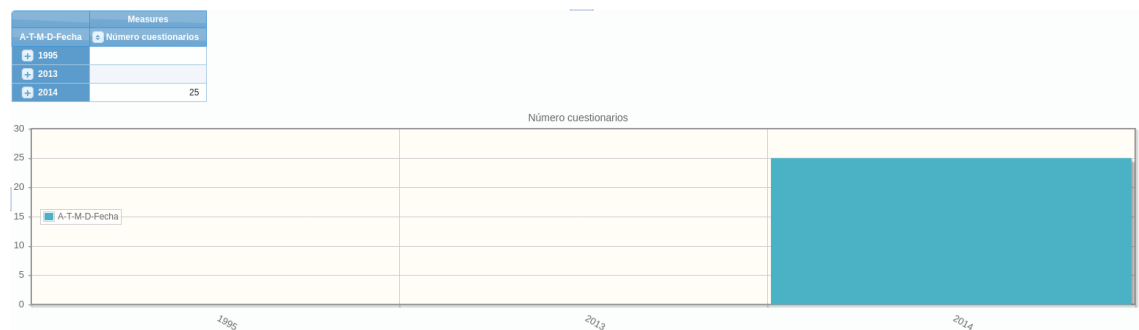
### Número de cuestionarios por año:



### Número de cuestionarios por provincia:



### Número de cuestionarios de la provincia de Ceuta por año (1995, 2013 y 2014):



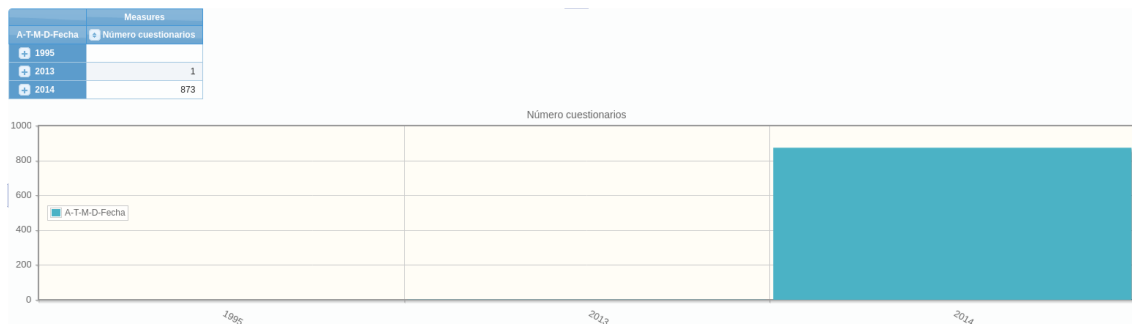
### Número de cuestionarios de la provincia de Madrid por año (1995, 2013 y 2014):



### Número de cuestionarios de la provincia de La Rioja (1995, 2013 y 2014):



Número de cuestionarios de la provincia de Tarragona (1995, 2013 y 2014):



Resultado promedio de la provincia de Madrid por año (1995, 2013 y 2014):



En la gráfica resultado promedio por año, se observa el pico más alto de resultado en las encuestas en el año 2013, en cambio, en 2014 la disminución en el resultado promedio destacable; por ello procedo a elaborar más gráficas.

El resultado promedio por provincia se mantiene más o menos estable por encima de 0,8, a excepción de Ceuta, Madrid, La Rioja y Tarragona. Para saber a qué se debe esa disminución, elaboro más gráficas, esta vez relacionadas con el número de cuestionarios por año, número de cuestionarios por provincia, número de cuestionario por año y provincia con menor resultado promedio, y resultado promedio por año y provincia con menor resultado.

El número de cuestionarios asciende significativamente en el año 2014, llegando a 32.696 cuestionarios ese año.

El número de cuestionarios por provincia se mantiene por debajo de los 2.000 en cada una de ellas, a excepción de Barcelona y Madrid, con más de 4.000 y 6.000 cuestionarios cada una.

El resultado promedio de las encuestas en Ceuta, La Rioja y Tarragona no se debe por una variación del resultado promedio entre los tres años, sino que se debe al promedio de las encuestas realizadas únicamente en el 2014; en cambio, Madrid presenta diferencias.

En Madrid, el número de cuestionarios pasa de 1 en 2013 a 6.623 en el año 2014.

Observando la gráfica de resultado promedio de la provincia de Madrid por año, el resultado promedio tan bajo se debe a que en 2013 el resultado promedio fue de 1 y en 2014 de 0.6.