
Práctica-7: Proyecto PDI/*Mondrian*.



UNIVERSIDAD DE GRANADA

Sistemas Multidimensionales (2019-2020)

Daniel Bolaños Martínez
danibolanos@correo.ugr.es
Grupo 2 - Viernes 15:30h

Índice

1. Introducción.	3
2. Selección de los datos.	3
3. Diseño del esquema multidimensional.	4
3.1. Diseño lógico y conceptual.	4
3.2. Estimación de instancias y hechos.	5
4. Preparación de los datos con Power Query.	6
5. Implementación del esquema multidimensional: ETL.	25
6. Componente OLAP.	60
7. Obtén un informe inicial libre y explica su contenido.	79
7.1. Mediante Slice&Dice obtén un nuevo informe y explica su contenido. . .	80
7.2. Mediante Drill-down obtén un nuevo informe y explica su contenido. . .	81
7.3. Mediante Roll-up obtén un nuevo informe y explica su contenido. . .	82
8. Bibliografía.	83

1. Introducción.

En esta práctica, desarrollaremos un proyecto basándonos en herramientas destinadas a profesionales de la informática relacionadas con los sistemas multidimensionales y que hemos estudiado durante las prácticas de la asignatura.

He elegido la combinación de PDI/*Mondrian* para realizar el proyecto, ya que me parecieron herramientas más intuitivas a la hora de trabajar con ellas. Además, realizaré el proyecto desde cero, haciendo tanto la selección de los datos como el diseño del esquema multidimensional.

2. Selección de los datos.

He realizado un proceso de búsqueda para obtener un conjunto de datos que se adecúe en complejidad y estructura al utilizado en las prácticas de la asignatura.

La página que he utilizado para buscar los datos ha sido *datos.gob.es* que recoge conjuntos de datos a nivel nacional de diversas instituciones.

El *dataset* elegido está basado en una *Encuesta de Ocupación en Campings* del **INE** y lleva un registro de clientes y estancias en campings por municipio en España.[2]

En el registro contamos con dos mediciones:

- **Pernoctaciones:** cada estancia de una noche registrada en el camping.
- **Viajeros:** persona que realiza una o más pernoctaciones seguidas en el mismo camping.

Además de esta información, contamos con datos relativos al mes y año de recolección de los datos, municipio donde se encuentran los campings y nacionalidad de los viajeros. Completaremos estos datos con los relativos a los archivos de comunidad autónoma y provincia que utilizamos en la segunda práctica.

En el archivo *2084-INE.xlsx* encontraremos los datos originales recogidos por el **INE**. En *códigos.xlsx*, hemos almacenado tanto los códigos usados en la práctica 2 como unos códigos relativos al mes que nos serán útiles para asignar el nombre del mes según el número del mismo.

3. Diseño del esquema multidimensional.

3.1. Diseño lógico y conceptual.

Mostraremos el diseño lógico y conceptual que tendremos que implementar haciendo uso de las herramientas escogidas y que será el primer paso para diseñar nuestro modelo.

Véase que son los que se presentaron como ejercicio 5 de teoría.

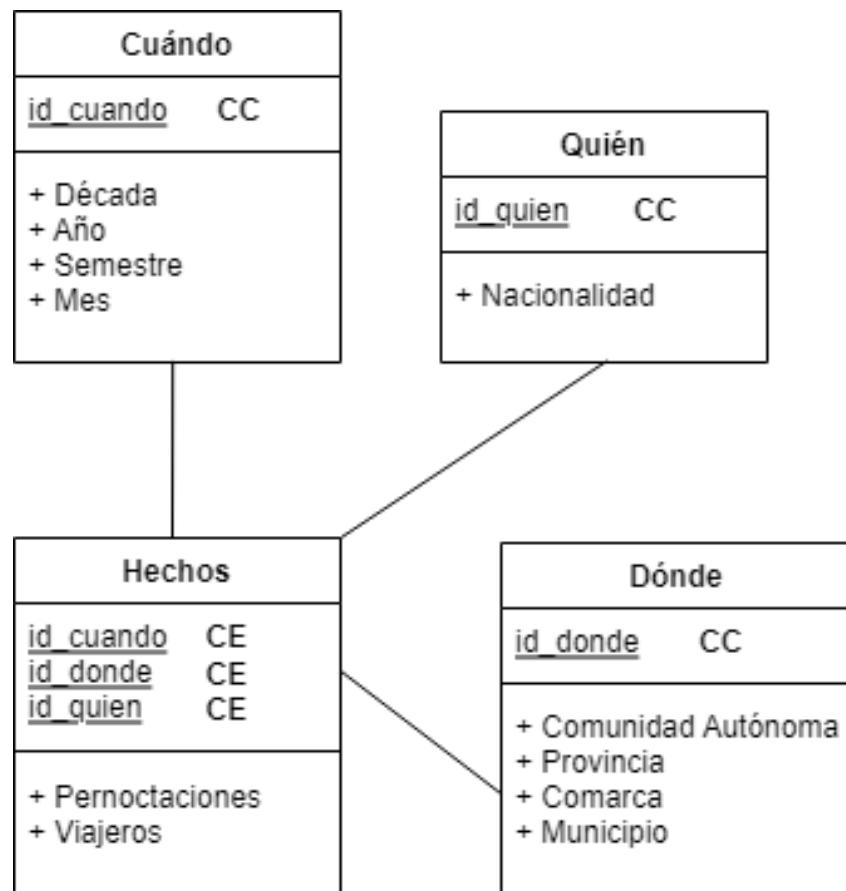


Figura 1: Diseño lógico de los datos.

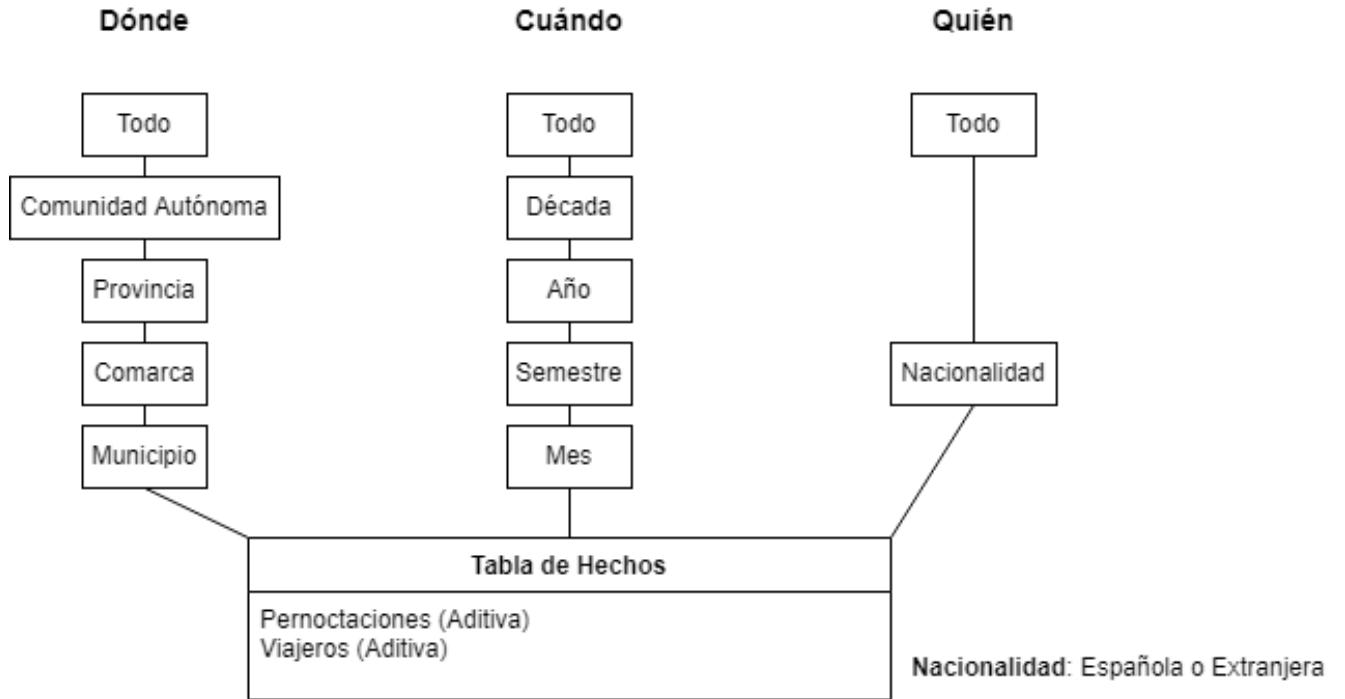


Figura 2: Diseño conceptual de los datos.

3.2. Estimación de instancias y hechos.

Estimaremos el número de instancias y hechos que tendrá nuestra base de datos.

Hay que tener en cuenta que el número podrá ser inferior debido a que eliminaremos las filas con valores *nulos* y vacíos.

- **Cuándo:** $n^{\circ}\text{años} \cdot n^{\circ} \text{meses por año} = (15 \text{ años} \cdot 12 \text{ meses}) + 2 \text{ meses 2020} = 182 \text{ instancias}$
- **Dónde:** $n^{\circ}\text{municipios} = 28 \text{ municipios} = 28 \text{ instancias}$
- **Quién:** $n^{\circ}\text{nacionalidades} = 2 \text{ nacionalidades} = 2 \text{ instancias}$
- **Hechos:** $n^{\circ}\text{municipios} \cdot n^{\circ}\text{nacionalidades} \cdot n^{\circ}\text{meses totales} = 28 \cdot 2 \cdot 182 = 10.192 \text{ hechos}$

4. Preparación de los datos con Power Query.

En primera instancia, realizaremos una preparación de los datos en *Power Query* para definir la estructura de nuestro esquema multidimensional pero sin generar las tablas de dimensión y de hechos.

A continuación, se muestra el archivo original que contiene los datos recogidos con la estructura definida por el **INE** y a partir del cual definiremos nuestra consulta en *Excel*.

Encuesta de ocupación en campings																						
Ocupación en Campings																						
Viajeros y pernoctaciones por puntos turísticos																						
Unidades:																						
Viajero																						
10	03031 Benidorm	2020M03	436	2020M02	582	2019M12	1.079	2019M11	668	2019M10	1.312	2019M09	3.675	2019M08	3.391	2019M07	7.594	2019M06	6.420			
11	03050 Campello, el				286		151		237		141		383		655		2.278	1.744		1.587		
12	03065 Elche/Eix																					
13	03139 Villajoyosa/Vila Joiosa, la				154		86		326		145		733		896			2.849	2.131			
14	04064 Mojácar						63		65		32		114		120		1.795	331				
15	11014 Conil de la Frontera																6.342	24.194	20.471	9.596		
16	11035 Tarifa																6.661	19.805	15.801	8.663		
17	12028 Benicasim/Benicàssim																1.725	7.177	4.658	3.320	1.488	
18	12085 Oropesa del Mar/Oropesa																					
19	12089 Peñíscola/Peñíscola																519	4.661				
20	17023 Blanes																414	2.853	10.103	6.575	3.099	
21	17047 Castelló d'Empúries																890	9.467	11.132	9.763	15.221	
22	17048 Castell-Platja d'Aro																	1.375	7.116	2.587	5.337	
23	17178 Sant Pere Pescador																	8.874	12.783	9.403	12.143	
24	17199 Torroella de Montgrí																	856	2.618	9.469	4.922	6.914
																		3.826	14.378	5.786	3.218	

Figura 3: Datos del archivo original *2084-INE.xlsx*.

Importamos a *Power Query* la hoja con los datos anteriores y eliminamos el paso *Tipo cambiado* (lo eliminaremos siempre que se añada automáticamente).

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table with 7 columns and 47 rows. The columns are labeled Column1 through Column7. The data includes various campsite counts and dates. The Power Query ribbon is visible at the top, and the 'PASOS APLICADOS' pane on the right shows steps like 'Navegación' and 'Filas superiores quitadas'.

	Column1	Column2	Column3	Column4	Column5	Column6	Column7
1	Encuesta de ocupación en campings	null	null	null	null	null	
2	Ocupación en Campings	null	null	null	null	null	
3		null	null	null	null	null	
4	Viajeros y pernoctaciones por puntos turísticos	null	null	null	null	null	
5	Unidades:	null	null	null	null	null	
6		null	null	null	null	null	
7	Viajero	null	null	null	null	null	
8	Residentes en España	null	null	null	null	null	
9	2020M03	2020M02	2020M01	2019M12	2019M11	2019M10	
10	03031 Benidorm	.	436	582	1079	668	
11	03050 Campello, el	.	286	151	297	141	
12	03065 Elche/Eix	.	-	-	-	-	
13	03139 Villajoyosa/Vila Joiosa, Ia	.	154	86	326	145	
14	04064 Mojácar	.	-	63	65	92	
15	11014 Conil de la Frontera	.	-	-	-	-	
16	11035 Tarifa	.	271	157	378	117	
17	12028 Benicasim/Benicàssim	.	154	180	-	370	
18	12085 Oropesa del Mar/Oropesa	.	-	-	-	-	
19	12089 Peñíscola/Peñíscola	.	414	408	709	1686	
20	17029 Blanes		600	1272	2141	1141	

Figura 4: Datos importados a *Power Query*.

Eliminamos las filas de cabecera y pie de página de la tabla dinámica.

Podemos eliminar las primeras o últimas filas de una consulta pulsando sobre [«Inicio», «Reducir filas», «Quitar filas», «Quitar filas superiores»] e [«Inicio», «Reducir filas», «Quitar filas», «Quitar filas inferiores»], respectivamente.

	Column1	Column2	Column3	Column4	Column5	Column6	Column7
1	Viajero	null	null	null	null		
2	Residentes en España	null	null	null	null		
3	2020M03	2020M02	2020M01	2019M12	2019M11	2019M10	
4	03031 Benidorm	.	436	582	1079	668	
5	03050 Campello, el	.	286	151	297	141	
6	03065 Elche/Elx	
7	03139 Villajoyosa/Vila Joiosa, la	.	154	86	326	145	
8	04064 Mojácar	.	.	63	65	92	
9	11014 Conil de la Frontera	
10	11035 Tarifa	.	271	157	378	117	
11	12028 Benicasim/Benicàssim	.	154	180	.	370	
12	12085 Oropesa del Mar/Oropesa	
13	12089 Peníscola/Pefíscula	.	414	408	709	1686	
14	17023 Blanes	.	890	.	1376	3141	
15	17047 Castelló d'Empúries	
16	17048 Castell-Platja d'Aro	
17	17178 Sant Pere Pescador	
18	17199 Torroella de Montgrí	.	.	721	318	.	
19	18140 Motril	
20	20000 Valencia	

Figura 5: Eliminamos la cabecera y pie de página.

Transponemos la tabla dinámica.

La operación que permite transformar las filas en columnas y viceversa es [«Transformar», «Tabla», «Transponer»].

The screenshot shows the Microsoft Power Query Editor interface. The ribbon at the top has tabs for Archivo, Inicio, Transformar, Agregar columna, and Vista. The Transformar tab is selected. The main area displays a table with 20 rows and 7 columns. The columns are labeled ABC Column1 through ABC Column7. The first two rows show 'Viajero' and 'Residentes en España' respectively. The data includes dates like '2020M03' and numerical values like '436'. On the right side, there's a 'Configuración de la consulta' pane with sections for 'PROPIEDADES' (Nombre: Camping) and 'PASOS APLICADOS' (Table transpuesta). The bottom status bar indicates '31 COLUMNAS, 733 FILAS' and 'VISTA PREVIA DESCARGADA EL JUEVES, 7 DE MAYO DE 2020'.

Figura 6: Transponemos la tabla de los datos.

Rellenamos los datos que faltan en la primera y segunda columna de la tabla.

Esta operación se realiza automáticamente seleccionando las columnas que queremos completar y pulsando sobre [«Transformar», «Cualquier columna», «Rellenar», «Abajo»].

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table with 31 columns and 733 rows. The first two columns are highlighted in green, indicating they are selected for transformation. The third column contains data like 'Residentes en España' and '2020M03'. The fourth column contains numerical values like 436, 582, 1079, etc. The fifth column contains codes like '03031 Benidorm'. The sixth column contains codes like '03050 Campello, el'. The seventh column contains codes like '03065 Elche/Eix'. The eighth column contains codes like '03139 Villajoyosa'. The ninth column contains a large number of zeros. The tenth column contains a large number of ones. The eleventh column contains a large number of twos. The twelfth column contains a large number of threes. The thirteenth column contains a large number of fours. The fourteenth column contains a large number of fives. The fifteenth column contains a large number of sixes. The sixteenth column contains a large number of sevens. The seventeenth column contains a large number of eights. The eighteenth column contains a large number of nines. The nineteenth column contains a large number of zeros. The twentieth column contains a large number of ones. The twenty-first column contains a large number of twos. The twenty-second column contains a large number of threes. The twenty-third column contains a large number of fours. The twenty-fourth column contains a large number of fives. The twenty-fifth column contains a large number of sixes. The twenty-sixth column contains a large number of sevens. The twenty-seventh column contains a large number of eights. The twenty-eighth column contains a large number of nines. The twenty-ninth column contains a large number of zeros. The thirtieth column contains a large number of ones. The thirty-first column contains a large number of twos. The bottom status bar indicates '31 COLUMNAS, 733 FILAS' and 'VISTA PREVIA DESCARGADA EL JUEVES, 7 DE MAYO DE 2020'.

Figura 7: Completamos con los valores que faltan.

Usaremos la primera fila de la tabla como encabezado de la misma.

Esta operación se puede llevar a cabo pulsando sobre [«Transformar», «Tabla», «Usar la primera fila como encabezado»].

The screenshot shows the Microsoft Power Query Editor interface. The ribbon at the top has 'Inicio' selected. The main area displays a table with 31 columns and 732 rows. The first row of the table is highlighted in green, indicating it is set as the header. The 'Transformar' tab is active in the ribbon. On the right side, there is a 'Configuración de origen de datos' pane where 'Camping' is listed under 'Orígenes de datos'. The status bar at the bottom indicates 'VISTA PREVIA DESCARGADA EL JUEVES, 7 DE MAYO DE 2020' and shows the date '14/05/2020' and time '16:36'.

Figura 8: Usamos la primera fila como encabezado.

Eliminamos la dinamización de las columnas de la tabla dinámica que contienen las mediciones del registro.

Podemos eliminar la dinamización seleccionando las columnas que contienen los datos que definen el contenido de las filas y pulsando sobre [«Transformar», «Cualquier columna», «Anular dinamización de columnas», «Anular dinamización de otras columnas»].

Viajero	Residencias en España	Valor
1 Viajero	Residentes en España	03031 Benidorm
2 Viajero	Residentes en España	03050 Campello, el
3 Viajero	Residentes en España	03139 Villajoyosa/Vila Joiosa, la
4 Viajero	Residentes en España	11035 Tarifa
5 Viajero	Residentes en España	12028 Benicasim/Benicàssim
6 Viajero	Residentes en España	12089 Peníscola/Peñíscola
7 Viajero	Residentes en España	17023 Blanes
8 Viajero	Residentes en España	18140 Motril
9 Viajero	Residentes en España	29069 Marbella
10 Viajero	Residentes en España	30016 Cartagena
11 Viajero	Residentes en España	46181 Oliva
12 Viajero	Residentes en España	03031 Benidorm
13 Viajero	Residentes en España	03050 Campello, el
14 Viajero	Residentes en España	03139 Villajoyosa/Vila Joiosa, la
15 Viajero	Residentes en España	04064 Mojacar
16 Viajero	Residentes en España	11035 Tarifa
17 Viajero	Residentes en España	12028 Benicasim/Benicàssim
18 Viajero	Residentes en España	12089 Peníscola/Peñíscola
19 Viajero	Residentes en España	18140 Motril
20 Viajero	Residentes en España	29069 Marbella

Figura 9: Anulamos la dinamización de las columnas seleccionadas.

Dividimos en dos campos distintos el código y nombre de municipio y el código de mes y número de año.

Podemos separar los valores en dos campos seleccionando la columna correspondiente y pulsando sobre [«Transformar», «Columna de texto», «Dividir columna», «Por delimitador»].

The screenshot shows the Microsoft Power Query Editor interface. The ribbon at the top has tabs for Archivo, Inicio, Transformar, Agregar columna, and Vista. The Transformar tab is selected. The main area displays a table with 20 rows of data. The first column is labeled 'Viajero' and contains values like 'Residentes en España'. The last column is labeled 'Valor' and contains numerical values. Between these columns are two intermediate columns: 'ABC _2.1' and 'ABC _2.2'. The 'ABC _2.1' column contains codes like '02', '02', '02', etc., and the 'ABC _2.2' column contains names like 'Benidorm', 'Campello, el', 'Villajoyosa/Vila Joiosa, la', etc. On the right side of the screen, there is a 'Configuración de la consulta' pane with sections for 'PROPIEDADES' (Nombre: Camping) and 'PASOS APLICADOS' (which includes 'Dividir columna por posición'). The status bar at the bottom indicates '7 COLUMNAS, 999+ FILAS' and 'VISTA PREVIA DESCARGADA EL JUEVES, 7 DE MAYO DE 2020'.

	ABC _2.1	ABC _2.2	ABC Attributo.1	ABC Attributo.2	ABC Valor
1	Viajero	Residentes en España	2020	02	03031 Benidorm
2	Viajero	Residentes en España	2020	02	03050 Campello, el
3	Viajero	Residentes en España	2020	02	03139 Villajoyosa/Vila Joiosa, la
4	Viajero	Residentes en España	2020	02	11035 Tarifa
5	Viajero	Residentes en España	2020	02	12028 Benicasim/Benicàssim
6	Viajero	Residentes en España	2020	02	12089 Peníscola/Peñíscola
7	Viajero	Residentes en España	2020	02	17023 Blanes
8	Viajero	Residentes en España	2020	02	18140 Motril
9	Viajero	Residentes en España	2020	02	29069 Marbella
10	Viajero	Residentes en España	2020	02	30016 Cartagena
11	Viajero	Residentes en España	2020	02	46181 Oliva
12	Viajero	Residentes en España	2020	01	03031 Benidorm
13	Viajero	Residentes en España	2020	01	03050 Campello, el
14	Viajero	Residentes en España	2020	01	03139 Villajoyosa/Vila Joiosa, la
15	Viajero	Residentes en España	2020	01	04064 Mojácar
16	Viajero	Residentes en España	2020	01	11035 Tarifa
17	Viajero	Residentes en España	2020	01	12028 Benicasim/Benicàssim
18	Viajero	Residentes en España	2020	01	12089 Peníscola/Peñíscola
19	Viajero	Residentes en España	2020	01	18140 Motril
20	Viajero	Residentes en España	2020	01	29069 Marbella

Figura 10: Separamos en dos columnas los códigos y nombres.

Renombramos los campos **Tipo**, **Nacionalidad**, **Año**, **CodMes**, **CodMunicipio**, **Municipio** y **Clientes** y asignamos a cada uno un tipo especificado.

Asignaremos a todos los campos el tipo *Texto* excepto al campo **Clientes** que asignaremos el tipo *Número entero*.

	A ⁰ _c Tipo	A ⁰ _c Nacionalidad	A ⁰ _c Año	A ⁰ _c CodMes	A ⁰ _c CodMunicipio	A ⁰ _c Municipio	1 ² ₃ Clientes
1	Viajero	Residentes en España	2020	02	03031	Benidorm	436
2	Viajero	Residentes en España	2020	02	03050	Campello, el	286
3	Viajero	Residentes en España	2020	02	03139	Villajoyosa/Vila Joio...	154
4	Viajero	Residentes en España	2020	02	11035	Tarifa	271
5	Viajero	Residentes en España	2020	02	12028	Benicasim/Benicàssim	154
6	Viajero	Residentes en España	2020	02	12089	Peníscola/Peñíscola	414
7	Viajero	Residentes en España	2020	02	17023	Blanes	890
8	Viajero	Residentes en España	2020	02	18140	Motril	721
9	Viajero	Residentes en España	2020	02	29069	Marbella	549
10	Viajero	Residentes en España	2020	02	30016	Cartagena	379
11	Viajero	Residentes en España	2020	02	46181	Oliva	386
12	Viajero	Residentes en España	2020	01	03031	Benidorm	582
13	Viajero	Residentes en España	2020	01	03050	Campello, el	151
14	Viajero	Residentes en España	2020	01	03139	Villajoyosa/Vila Joio...	86
15	Viajero	Residentes en España	2020	01	04064	Mojácar	63
16	Viajero	Residentes en España	2020	01	11035	Tarifa	157
17	Viajero	Residentes en España	2020	01	12028	Benicasim/Benicàssim	180
18	Viajero	Residentes en España	2020	01	12089	Peníscola/Peñíscola	408
19	Viajero	Residentes en España	2020	01	18140	Motril	318
20	Viajero	Residentes en España	2020	01	29069	Marbella	298

Figura 11: Renombramos las columnas y asignamos los tipos definidos.

* El campo **Clientes** no será definitivo, es el nombre que le ponemos a esa columna de forma temporal hasta que dinamicemos por **Tipo**.

Definiremos una columna **Viajeros** y otra **Pernoctaciones** a partir de una columna dinámica para cada valor del campo **Tipo**.

Se puede definir directamente una columna para cada valor del campo **Tipo**, seleccionando la columna y pulsando sobre [«Transformar», «Cualquier columna», «Columna dinámica»].

Nacionalidad	Año	CodMes	CodMunicipio	Municipio	Viajeros	Pernoctaciones
Española	2005	01	03031	Benidorm	80	1020
Española	2005	01	03050	Campello, el	172	1024
Española	2005	01	03065	Elche/Eix	316	3216
Española	2005	01	03139	Villajoyosa/Vila Jolo...	830	1088
Española	2005	01	04064	Mojácar	159	143
Española	2005	01	11035	Tarifa	362	763
Española	2005	01	12028	Benicasim/Benicàssim	19	337
Española	2005	01	12085	Oropesa del Mar/Or...	149	262
Española	2005	01	12089	Peníscola/Pèñiscola	134	226
Española	2005	01	18140	Motril	null	null
Española	2005	01	21042	Isla Cristina	202	267
Española	2005	01	29069	Marbella	501	5364
Española	2005	01	29094	Vélez-Málaga	390	1014
Española	2005	01	30016	Cartagena	237	2074
Española	2005	01	30020	Fortuna	null	null
Española	2005	01	30026	Mazarrón	169	604
Española	2005	01	46181	Oliva	59	613
Española	2005	02	03031	Benidorm	42	929
Española	2005	02	03050	Campello, el	170	1528
Española	2005	02	03065	Elche/Eix	382	2571

Figura 12: Definimos las nuevas columnas **Viajeros** y **Pernoctaciones**.

Además, hemos cambiado los valores para el campo **Nacionalidad: Residentes en España y Residentes en el Extranjero** por *Española* y *Extranjera* respectivamente.

Eliminaremos las filas con valores *null* y filtraremos las filas que contienen valores vacíos indicados como '' o '..' en nuestro ejemplo.

The screenshot shows the Microsoft Power Query Editor interface. The ribbon at the top has tabs for Archivo, Inicio, Transformar, Agregar columna, and Vista. The 'Transformar' tab is selected. On the left, there's a sidebar labeled 'Consultas' with a list of queries. The main area displays a table with 20 rows of data. The columns are: Año (Year), Nacionalidad (Nationality), CodMes (Month Code), CodMunicipio (Municipality Code), Municipio (Municipality Name), Viajeros (Visitors), and Pernoctaciones (Overnights). The data includes various Spanish municipalities and their visitor statistics for the year 2005. To the right of the table, the 'Transformar' ribbon group is visible, containing tools like 'Elegir columnas', 'Quitar filas', 'Conservar filas', 'Dividir columna', 'Agrupar por', 'Usar la primera fila como encabezado', 'Reemplazar los valores', 'Combinar consultas', 'Anexar consultas', 'Combinar archivos', 'Administrar parámetros', 'Configuración de origen de datos', and 'Nuevo origen'. A configuration pane on the right shows 'Nombre: Camping' and a list of applied steps under 'PASOS APLICADOS', including 'Filas filtradas1' (highlighted in green).

Año	Nacionalidad	CodMes	CodMunicipio	Municipio	Viajeros	Pernoctaciones
2005	Española	01	08031	Benidorm	80	1020
2005	Española	01	08050	Campello, el	172	1024
2005	Española	01	08065	Elche/Eix	316	3216
2005	Española	01	08139	Villajoyosa/Vila Joia...	830	1088
2005	Española	01	04064	Mojácar	159	143
2005	Española	01	11035	Tarifa	362	763
2005	Española	01	12028	Benicasim/Benicàssim	19	337
2005	Española	01	12085	Oropesa del Mar/Or...	149	262
2005	Española	01	12089	Peníscola/Peníscola	134	226
2005	Española	01	21042	Isla Cristina	202	267
2005	Española	01	29069	Marbella	501	5364
2005	Española	01	29094	Vélez-Málaga	390	1014
2005	Española	01	30016	Cartagena	237	2074
2005	Española	01	30026	Mazarrón	169	604
2005	Española	01	46181	Oliva	59	613
2005	Española	02	08031	Benidorm	42	929
2005	Española	02	08050	Campello, el	170	1528
2005	Española	02	08065	Elche/Eix	382	2571
2005	Española	02	08139	Villajoyosa/Vila Joia...	613	1602
2005	Española	02	04064	Mojácar	856	803

Figura 13: Eliminamos filas que contengan valores nulos o vacíos.

Cargaremos en *Power Query* el archivo *codigos.xlsx* que contiene las hojas relativas a: *codMes*, *codCA* y *codINE*.

En la hoja *codMes*, podemos ver la relación entre el código de mes y el formato de nombre que queremos darle al campo **Mes**.

Cod	Mes
01	01-enero
02	02-febrero
03	03-marzo
04	04-abril
05	05-mayo
06	06-junio
07	07-julio
08	08-agosto
09	09-septiembre
10	10-octubre
11	11-noviembre
12	12-diciembre

Figura 14: Información contenida en la hoja *codMes*.

Las hojas *codCA* y *codINE* fueron utilizadas en la práctica del censo y nos aportan información sobre otros campos relativos a la dimensión **Dónde** como la comarca o la comunidad autónoma.

codCA - Editor de Power Query

Consultas [4]

- Camping
- codCA**
- codINE
- codMes

	A _B C COD_PROV	A _B C PROVINCIA	A _B C COD_CA	A _B C COMUNIDAD_AUTONOMA	A _B C CAPITAL
1	01	Araba/Álava	16	País Vasco/Euskadi	Vitoria-Gasteiz
2	02	Albacete	08	Castilla-La Mancha	Albacete
3	03	Alacant/Alicante	10	Comunitat Valenciana	Alicante/Alacant
4	04	Almería	01	Andalucía	Almería
5	05	Ávila	07	Castilla y León	Ávila
6	06	Badajoz	11	Extremadura	Badajoz
7	07	Illes Balears	04	Illes Balears	Palma
8	08	Barcelona	09	Cataluña/Catalunya	Barcelona
9	09	Burgos	07	Castilla y León	Burgos
10	10	Cáceres	11	Extremadura	Cáceres
11	11	Cádiz	01	Andalucía	Cádiz
12	12	Castelló/Castellón	10	Comunitat Valenciana	Castelló de la Plana
13	13	Ciudad Real	08	Castilla-La Mancha	Ciudad Real
14	14	Córdoba	01	Andalucía	Córdoba
15	15	A Coruña	12	Galicia	A Coruña
16	16	Cuenca	08	Castilla-La Mancha	Cuenca
17	17	Girona	09	Cataluña/Catalunya	Girona
18	18	Granada	01	Andalucía	Granada
19	19	Guadalajara	08	Castilla-La Mancha	Guadalajara
20	20	Gipuzkoa	16	País Vasco/Euskadi	Donostia/San Sebastián

5 COLUMNAS, 52 FILAS

codINE - Editor de Power Query

Consultas [4]

- Camping
- codCA
- codINE**
- codMes

	A _B C COD_INE	A _B C COD_PROV	A _B C NOMBRE_ACTUAL	A _B C SUPERFICIEHa	A _B C CAPITAL	A _B C ALTITUD
1	01001000000	01	Alegria-Dulantzi	1994,5872	Alegria-Dulantzi	
2	01002000000	01	Amurrio	9635,289	Amurrio	
3	01003000000	01	Aramaiar	7327,4027	Ibarra	
4	01004000000	01	Artziniega	2728,73	Artziniega	
5	01006000000	01	Armiñón	1294,4326	Armiñón	
6	01008000000	01	Arratzua-Ubarrundia	5740,9889	Durana	
7	01009000000	01	Aspárrena	6533,64	Arala	
8	01010000000	01	Ayala/Alara	14074,15	Arespalditza/Respaldiza	
9	01011000000	01	Baños de Ebro/Mañueta	945,9752	Baños de Ebro/Mañueta	
10	01013000000	01	Barrundia	9742,34	Ozeta	
11	01014000000	01	Berantegi	3573,0031	Berantegi	
12	01016000000	01	Bernedo	13044,2902	Bernedo	
13	01017000000	01	Campezo/Kanpezu	8536,6875	Santa Cruz de Campezo/Santikurutze Kanpezu	
14	01018000000	01	Zigolita	10206,9376	Gopari	
15	01019000000	01	Kripian	1251,599	Kripian	
16	01020000000	01	Kuartango	8403,4771	Zuhatzu Kuartango	
17	01021000000	01	Elburgo/Burgelu	3209,3031	Elburgo/Burgelu	
18	01022000000	01	Etxelego	1632,3129	Etxelego	
19	01023000000	01	Elvillar/Bilar	1750,2228	Elvillar/Bilar	
20	01027000000	01	Txonta Ganea	1771,8239	Aitzorri	

13 COLUMNAS, 999+ FILAS

Figura 15: Información contenida en las hojas codCA y codINE.

Pulsando sobre [«Inicio», «Combinar», «Combinar consultas», «Combinar consultas»] podemos combinar la información que nos aportan las columnas y filtrar los campos que necesitemos para nuestra tabla principal.

Previamente hemos combinado las hojas *codCA* y *codINE* a partir del código de provincia. En la tabla principal, a través del campo **CodMunicipio**, extraeremos información de la hoja *codINE* relativa a los códigos y nombres de comarca, provincia y comunidad autónoma.

	Municipio	Viajeros	Pernocaciones	codINE.CO_COMARCA	codINE.DS_COMARCA	codINE.codCA.COD_PROV
1	Benidorm	80	1020	304	Central	03
2	Benidorm	42	929	304	Central	03
3	Benidorm	1081	4298	304	Central	03
4	Benidorm	594	2661	304	Central	03
5	Benidorm	1016	3562	304	Central	03
6	Benidorm	1275	5682	304	Central	03
7	Benidorm	3378	13360	304	Central	03
8	Benidorm	6627	29621	304	Central	03
9	Benidorm	1110	6843	304	Central	03
10	Benidorm	428	3283	304	Central	03
11	Benidorm	41	1879	304	Central	03
12	Benidorm	236	2326	304	Central	03
13	Benidorm	68	1634	304	Central	03
14	Benidorm	92	2201	304	Central	03
15	Benidorm	294	2427	304	Central	03
16	Benidorm	2351	8617	304	Central	03
17	Benidorm	771	3760	304	Central	03
18	Benidorm	1715	6619	304	Central	03
19	Benidorm	3048	15018	304	Central	03
20	Benidorm	5074	10753	304	Central	03

Figura 16: Expandimos información desde *codINE*.

Pulsando sobre [«Transformar», «Columna de texto», «Formato», «Poner En Mayúsculas Cada Palabra»], podemos poner en minúsculas los valores de comarca.

De forma similar realizamos una combinación a partir del campo **CodMes** y la hoja *codMes* para obtener los nombres para los meses.

The screenshot shows the Microsoft Power Query Editor interface. The top ribbon has tabs like Archivo, Inicio, Transformar, Agregar columna, and Vista. The main area displays a table with three columns: codINE.codCA.COD_PROV, codINE.codCA.PROVINCIA, and codINE.codCA.COMUNIDAD_AUTONOMA. The 'PASOS APLICADOS' pane on the right lists several steps, including 'Se expandió codMes.1' (which is highlighted in green). The bottom status bar shows '14 COLUMNAS, 999+ FILAS' and the date '14/05/2020'.

Figura 17: Expandimos información desde *codMes*.

Reordenaremos las columnas y les asignaremos un nuevo nombre y tipo *Texto* para cada una de los nuevos campos añadidos en el último paso.

A_C Mes	A_C Nacionalidad	A_C CodMunicipio	A_C Municipio	A_C CodComarca	A_C Comarca	A_C CodProvincia
01-enero	Española	03031	Benidorm	304	Central	03
01-enero	Española	03031	Benidorm	304	Central	03
02-febrero	Española	03031	Benidorm	304	Central	03
03-marzo	Española	03031	Benidorm	304	Central	03
04-abril	Española	03031	Benidorm	304	Central	03
05-mayo	Española	03031	Benidorm	304	Central	03
06-junio	Española	03031	Benidorm	304	Central	03
07-julio	Española	03031	Benidorm	304	Central	03
08-agosto	Española	03031	Benidorm	304	Central	03
09-septiembre	Española	03031	Benidorm	304	Central	03
10-octubre	Española	03031	Benidorm	304	Central	03
11-noviembre	Española	03031	Benidorm	304	Central	03
12-diciembre	Española	03031	Benidorm	304	Central	03
02-febrero	Española	03031	Benidorm	304	Central	03
03-marzo	Española	03031	Benidorm	304	Central	03
04-abril	Española	03031	Benidorm	304	Central	03
05-mayo	Española	03031	Benidorm	304	Central	03
06-junio	Española	03031	Benidorm	304	Central	03
07-julio	Española	03031	Benidorm	304	Central	03

Figura 18: Reordenamos y renombramos las columnas.

Los campos creados en orden de aparición son: **Año**, **CodMes**, **Mes**, **Nacionalidad**, **CodMunicipio**, **Municipio**, **CodComarca**, **Comarca**, **CodProvincia**, **Provincia**, **CodCA**, **CA**, **CodMunicipio**, **Municipio**, **Viajeros** y **Pernoctaciones**.

Añadiremos una columna condicional para crear el campo **Semestre** a partir del campo **CodMes**.

Pulsando sobre [«Aregar columna», «General», «Columna condicional»].

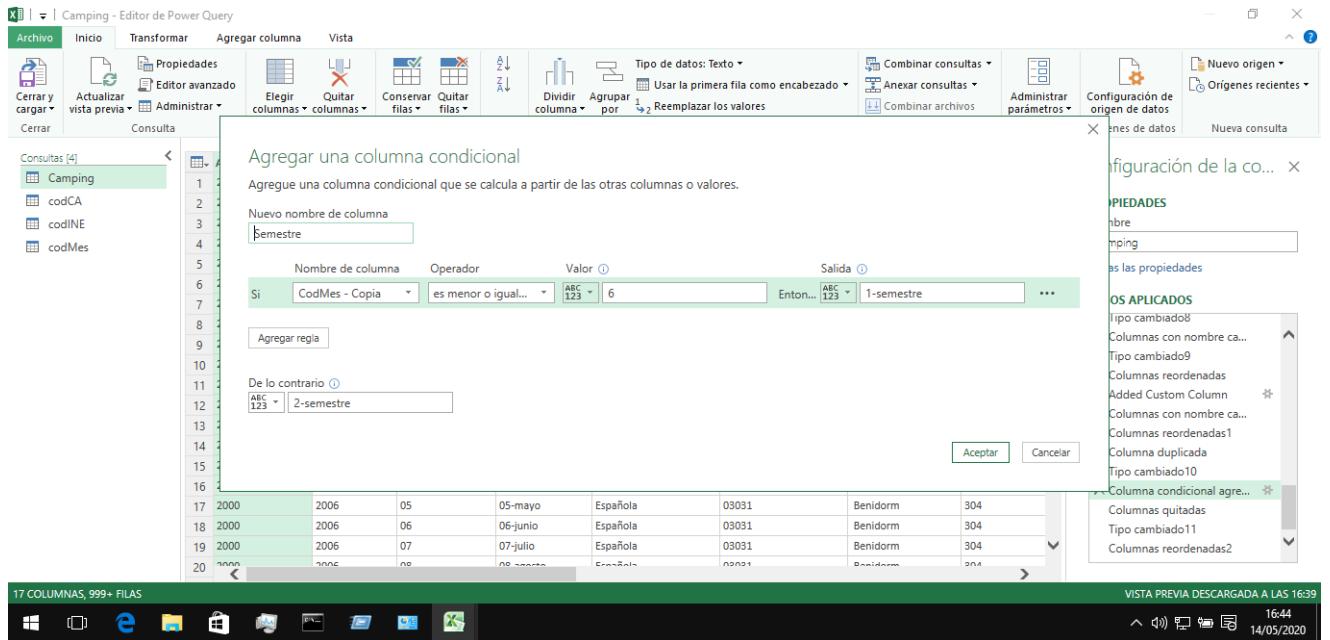


Figura 19: Añadimos una columna condicional para **Semestre**.

Crearemos el campo **Década** a partir de la información que nos aporta el campo **Año**.

Pulsando sobre [«Añadir columna», «Columna a partir de los ejemplos», «A partir de la selección»].

	Año	Década	Año	Año	Semestre	Año	CodMes	Año	Mes	Nacionalidad	Año	CodMunicipio	Año	Municipio
1	2000	2000			1-semestre	01		01-enero		Española	03031		Benidorm	
2	2000	2000			1-semestre	01		01-enero		Española	03031		Benidorm	
3	2000	2000			1-semestre	02		02-febrero		Española	03031		Benidorm	
4	2000	2000			1-semestre	03		03-marzo		Española	03031		Benidorm	
5	2000	2000			1-semestre	04		04-abril		Española	03031		Benidorm	
6	2000	2000			1-semestre	05		05-mayo		Española	03031		Benidorm	
7	2000	2000			1-semestre	06		06-junio		Española	03031		Benidorm	
8	2000	2000			2-semestre	07		07-julio		Española	03031		Benidorm	
9	2000	2000			2-semestre	08		08-agosto		Española	03031		Benidorm	
10	2000	2000			2-semestre	09		09-septiembre		Española	03031		Benidorm	
11	2000	2000			2-semestre	10		10-octubre		Española	03031		Benidorm	
12	2000	2000			2-semestre	11		11-noviembre		Española	03031		Benidorm	
13	2000	2000			2-semestre	12		12-diciembre		Española	03031		Benidorm	
14	2000	2000			1-semestre	02		02-febrero		Española	03031		Benidorm	
15	2000	2000			1-semestre	03		03-marzo		Española	03031		Benidorm	
16	2000	2000			1-semestre	04		04-abril		Española	03031		Benidorm	
17	2000	2000			1-semestre	05		05-mayo		Española	03031		Benidorm	
18	2000	2000			1-semestre	06		06-junio		Española	03031		Benidorm	
19	2000	2000			2-semestre	07		07-julio		Española	03031		Benidorm	
20	2000	2000			2-semestre	no		no		Española	no		Benidorm	

Figura 20: Añadimos una columna a partir de ejemplos para **Década**.

Crearemos una hoja con la nueva tabla a la que llamaremos **Registro** y será a partir de la cual generaremos las tablas de dimensiones y hechos con las otras herramientas.

Deshabilitaremos la carga para el resto de hojas y guardaremos el trabajo en el archivo *P7-danibolanos.xlsx*.

	Década	Año	Semestre	CodMes	Mes	Nacionalidad	CodMunicipio	Municipio	CodComarca	Comarca	CodProvincia	Provincia
1	2000	2005	1-semestre	01	01-enero	Española	03031	Benidorm	304	Central	03	Alacant/Alic
2	2000	2006	1-semestre	01	01-enero	Española	03031	Benidorm	304	Central	03	Alacant/Alic
3	2000	2005	1-semestre	02	02-febrero	Española	03031	Benidorm	304	Central	03	Alacant/Alic
4	2000	2005	1-semestre	03	03-marzo	Española	03031	Benidorm	304	Central	03	Alacant/Alic
5	2000	2005	1-semestre	04	04-abril	Española	03031	Benidorm	304	Central	03	Alacant/Alic
6	2000	2005	1-semestre	05	05-mayo	Española	03031	Benidorm	304	Central	03	Alacant/Alic
7	2000	2005	1-semestre	06	06-junio	Española	03031	Benidorm	304	Central	03	Alacant/Alic
8	2000	2005	2-semestre	07	07-julio	Española	03031	Benidorm	304	Central	03	Alacant/Alic
9	2000	2005	2-semestre	08	08-agosto	Española	03031	Benidorm	304	Central	03	Alacant/Alic
10	2000	2005	2-semestre	09	09-septiembre	Española	03031	Benidorm	304	Central	03	Alacant/Alic
11	2000	2005	2-semestre	10	10-octubre	Española	03031	Benidorm	304	Central	03	Alacant/Alic
12	2000	2005	2-semestre	11	11-noviembre	Española	03031	Benidorm	304	Central	03	Alacant/Alic
13	2000	2005	2-semestre	12	12-diciembre	Española	03031	Benidorm	304	Central	03	Alacant/Alic
14	2000	2006	1-semestre	02	02-febrero	Española	03031	Benidorm	304	Central	03	Alacant/Alic
15	2000	2006	1-semestre	03	03-marzo	Española	03031	Benidorm	304	Central	03	Alacant/Alic
16	2000	2006	1-semestre	04	04-abril	Española	03031	Benidorm	304	Central	03	Alacant/Alic
17	2000	2006	1-semestre	05	05-mayo	Española	03031	Benidorm	304	Central	03	Alacant/Alic
18	2000	2006	1-semestre	06	06-junio	Española	03031	Benidorm	304	Central	03	Alacant/Alic
19	2000	2006	2-semestre	07	07-julio	Española	03031	Benidorm	304	Central	03	Alacant/Alic
20	2000	2006	2-semestre	08	08-agosto	Española	03031	Benidorm	304	Central	03	Alacant/Alic

Figura 21: Guardamos la consulta como **Registro**.

5. Implementación del esquema multidimensional: ETL.

Realizaremos la implementación del esquema multidimensional sobre *PDI (Pentaho Data Integration)* también conocido como *Kettle* mediante su interfaz gráfica *Spoon*.

Haremos las operaciones pertinentes sobre los datos generados en *Excel* y definiremos la estructura de las tablas de dimensión y hechos en *Spoon*.

Finalmente lo almacenaremos en una base de datos que crearemos previamente en *PostgreSQL*.

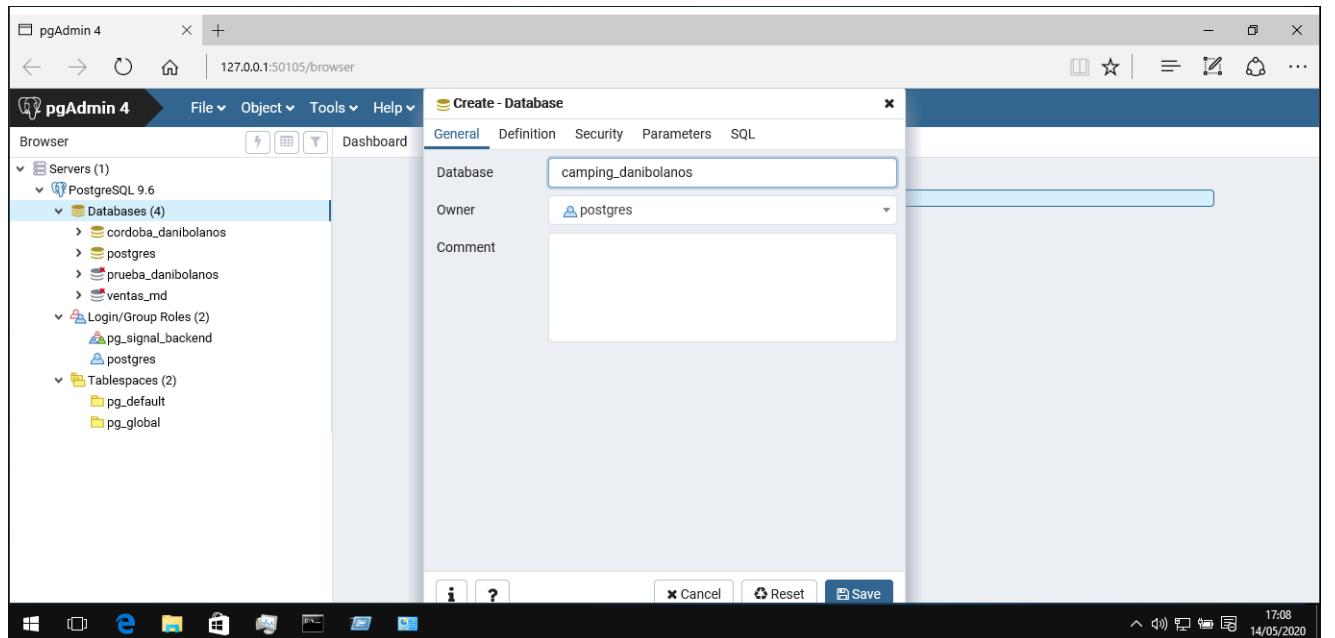


Figura 22: Creación de la BD **camping_danibolanos** en *pgAdmin*.

Creamos las tablas **cuando_danibolanos**, **donde_danibolanos**, **quien_danibolanos** y **registro_danibolanos** para la base de datos **camping_danibolanos** creada anteriormente.

The screenshot shows the pgAdmin 4 interface with the following details:

- Title Bar:** pgAdmin 4, 127.0.0.1:50105/browser
- Toolbar:** File, Object, Tools, Help
- Properties Tab:** Selected
- Dashboard:** Not selected
- SQL:** Not selected
- Statistics:** Not selected
- Dependencies:** Not selected
- Dependents:** Not selected
- Left Sidebar (Browser View):**
 - Tables (4) is selected
 - Sub-items under Tables: cuando_danibolanos, donde_danibolanos, quien_danibolanos, registro_danibolanos
 - Other categories: FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Sequences, Trigger Functions, Types, Views, cordoba_danibolanos, postgres, prueba_danibolanos, ventas_md, Login/Group Roles (2), pg_signal_backend, postgres, Tablespaces (2)
- Table View:**

Name	Owner	Partitioned table?	Comment
cuando_danibolanos	postgres	<input type="checkbox"/> False	
donde_danibolanos	postgres	<input type="checkbox"/> False	
quien_danibolanos	postgres	<input type="checkbox"/> False	
registro_danibolanos	postgres	<input type="checkbox"/> False	
- System Taskbar:** Windows icons, Start button, Task View, File Explorer, Taskbar search, Date/Time (14/05/2020), 17:32

Figura 23: Creación de las tablas de dimensión y hechos.

Configuramos la conexión a la BD que hemos creado en *pgAdmin* para las transformaciones en *Spoon*. Usaremos la configuración que se nos especificó en la práctica 3.

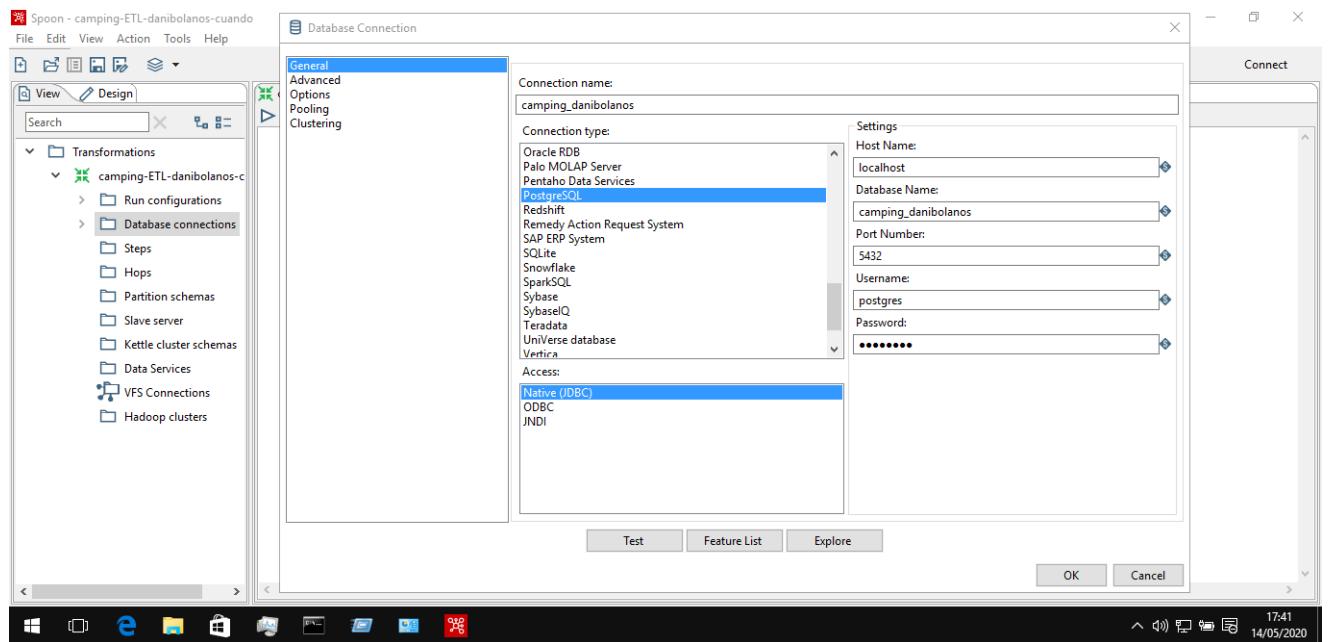


Figura 24: Creamos la conexión de la base de datos en *Spoon*.

Crearemos una transformación para cada tabla. Para la transformación **camping-ETL-danibolanos-cuando** crearemos los siguientes pasos para pasar de la hoja **Registro** a la tabla **cuando_danibolanos**.

Para ello necesitamos quedarnos con los campos **Década**, **Semestre**, **Año**, **Cod-Mes** y **Mes** y completar la tabla con el campo **id_cuando**.

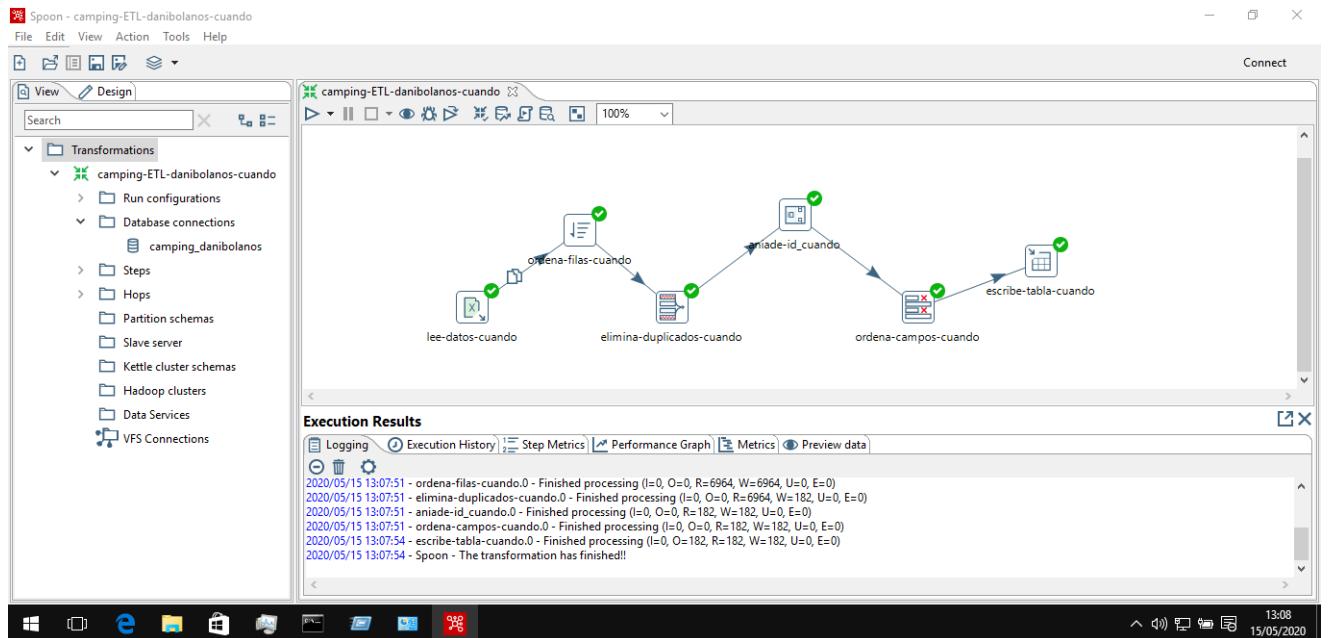


Figura 25: Configuramos los diferentes pasos en la transformación.

De izquierda a derecha se han usado los pasos: *Excel input file*, *Sort rows*, *Unique rows*, *Add sequence*, *Select values* y *Table output*.

Configuramos el paso *Input Excel* para la obtención de los datos del archivo **P7-danibolanos.xlsx**.

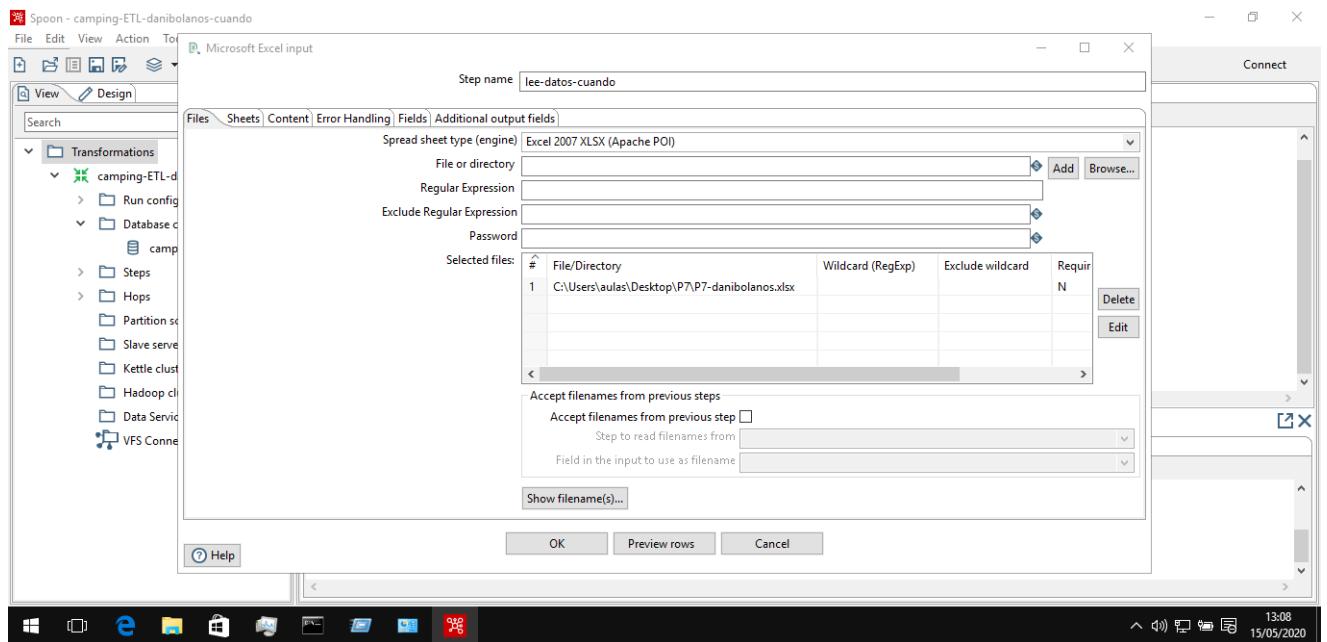


Figura 26: Extraemos los datos con el paso *Input Excel*.

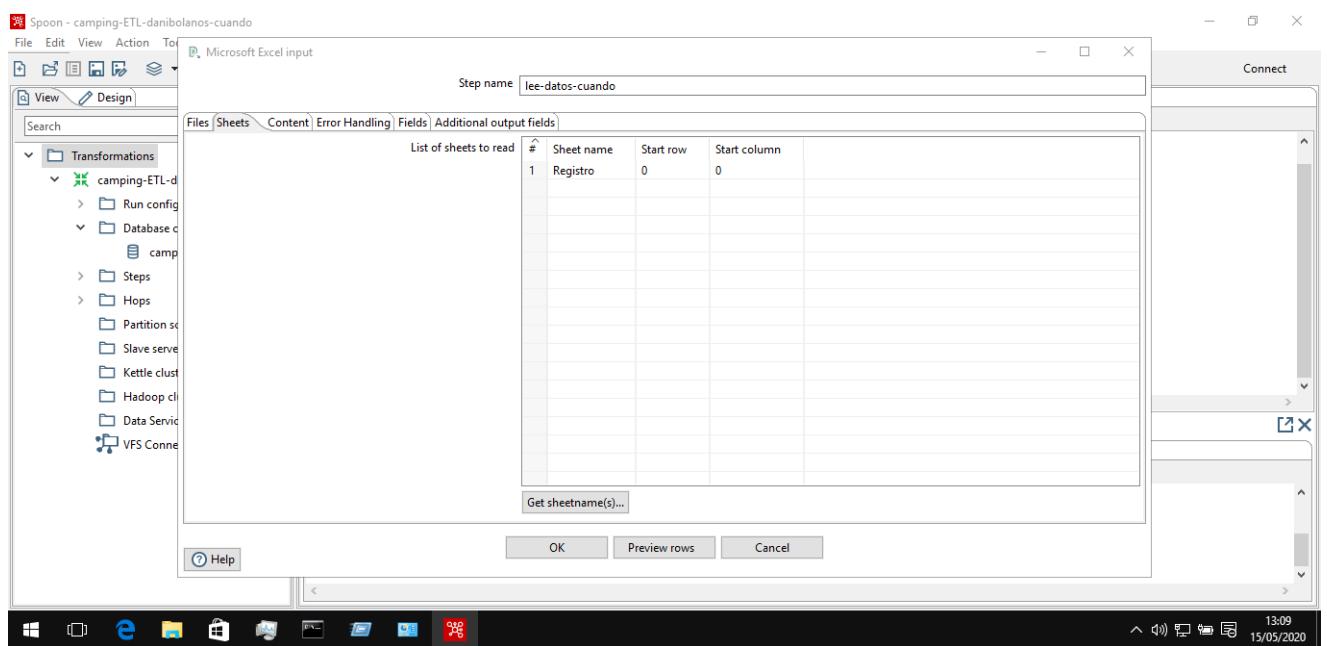


Figura 27: Extraemos los datos de la hoja **Registro**.

Seleccionamos los campos y ponemos los nombres en minúscula, sin tildes y utilizando el criterio *camel_case*.

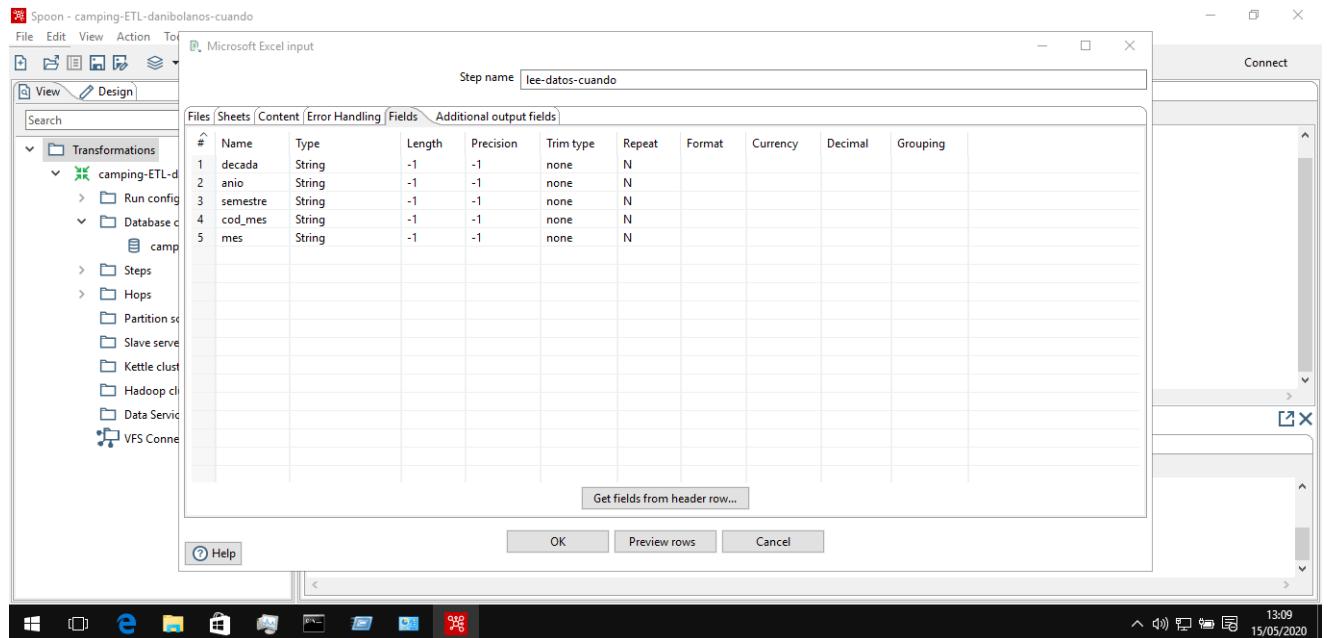


Figura 28: Campos de hoja **Registro** con los criterios especificados.

* Cabe destacar que este paso leerá las columnas en el mismo orden en que se especifique, así que si queremos extraer campos intermedios debemos guardar la lista completa y eliminar los que no queramos posteriormente.

Ordenamos los campos con el paso *Sort rows* antes de eliminar los duplicados. Esto es necesario ya que el paso *Unique rows* solo eliminará duplicados cuando aparezcan de forma consecutiva.

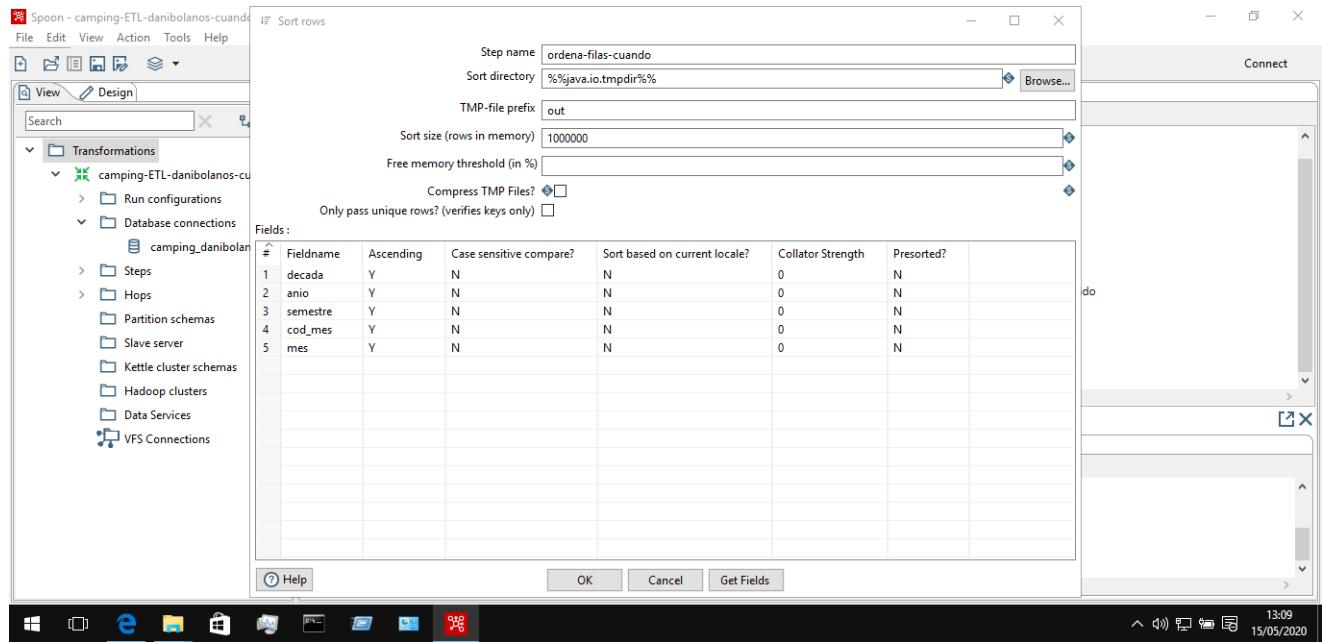


Figura 29: Ordenamos los valores de los campos con el paso *Sort rows*.

Eliminamos las filas duplicadas con el paso *Unique rows*. Si no especificamos nada elimina coincidencias de filas completas.

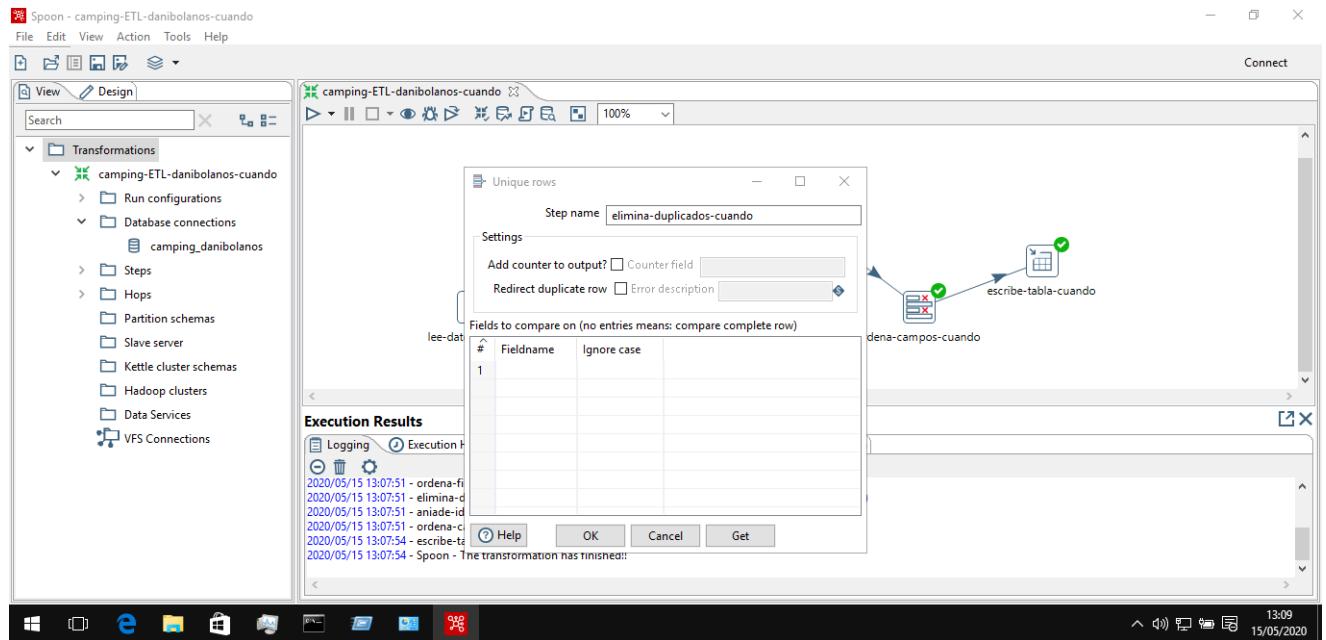


Figura 30: Eliminamos duplicados con el paso *Unique rows*.

Añadimos un campo **id_cuando** con valores del 1 al 182 que servirá como clave primaria para la tabla de dimensión **cuando_danibolanos**.

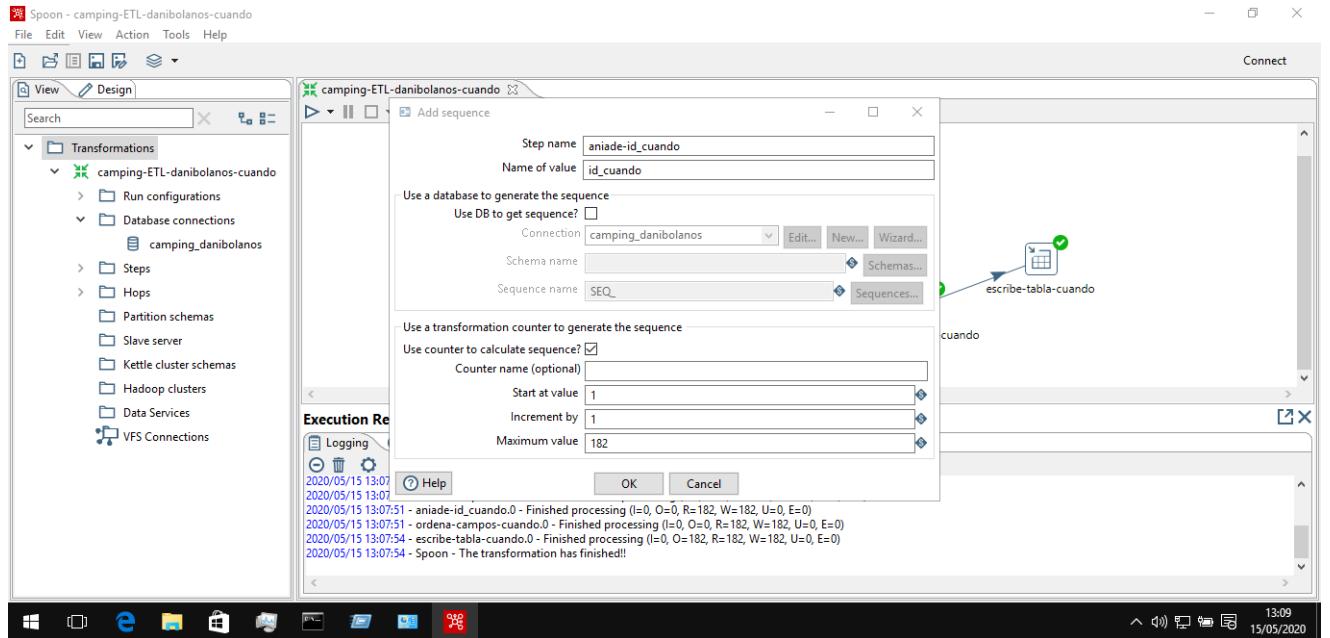


Figura 31: Añadimos el campo **id_cuando** con el paso *Add sequence*.

Reordenamos los campos para poner en primer lugar el campo **id_cuando**.

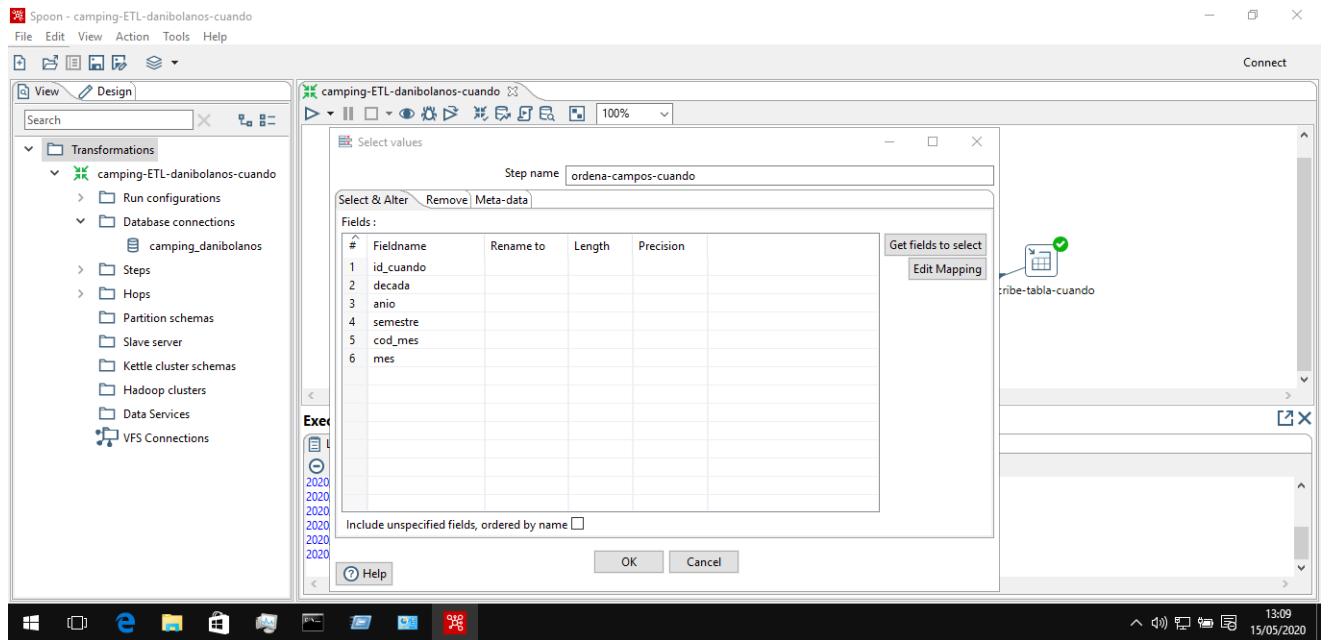


Figura 32: Reordenamos los campos con el paso *Select values*.

Pulsamos el botón *SQL* para crear la tabla **cuando_danibolanos**.

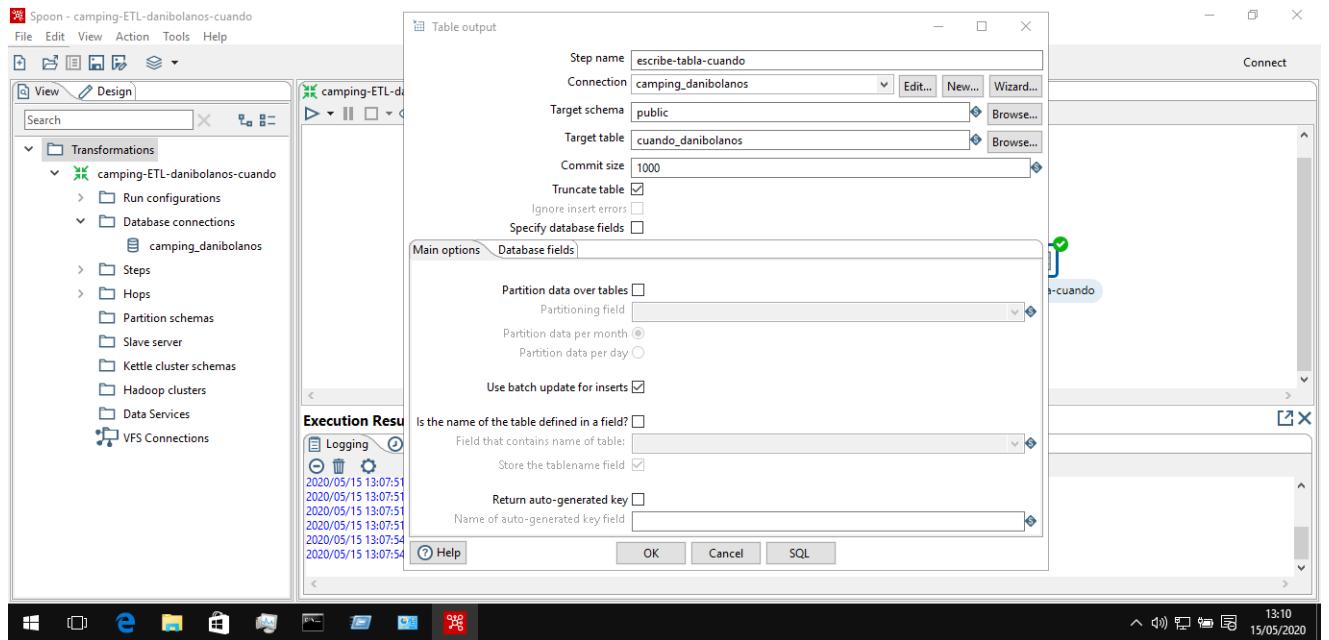


Figura 33: Escribimos los valores en la tabla.

Al pulsar sobre el botón *SQL* del paso *Table output*, generamos las sentencias *SQL* encargadas de crear las tablas en la BD.

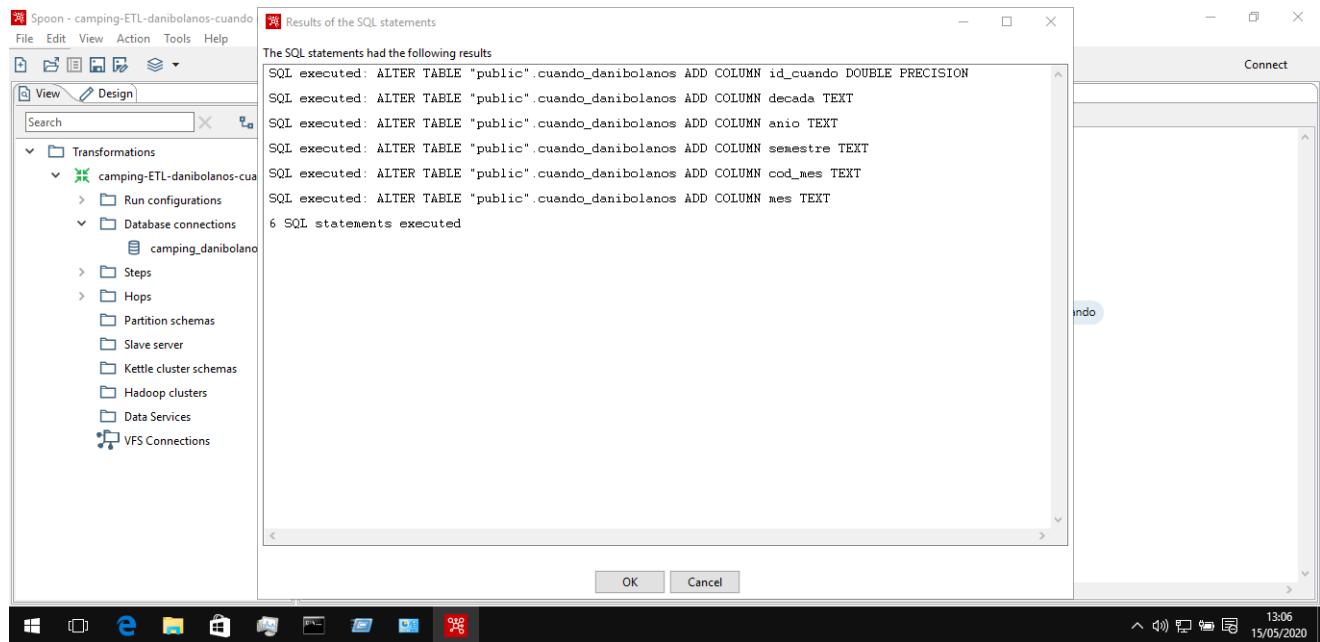


Figura 34: Generamos las sentencias *SQL*.

En *Spoon*, la llave primaria se genera con el tipo *double precision*. En *pgAdmin* especificaremos el tipo como *integer* y estableceremos la llave como clave primaria y no nula.

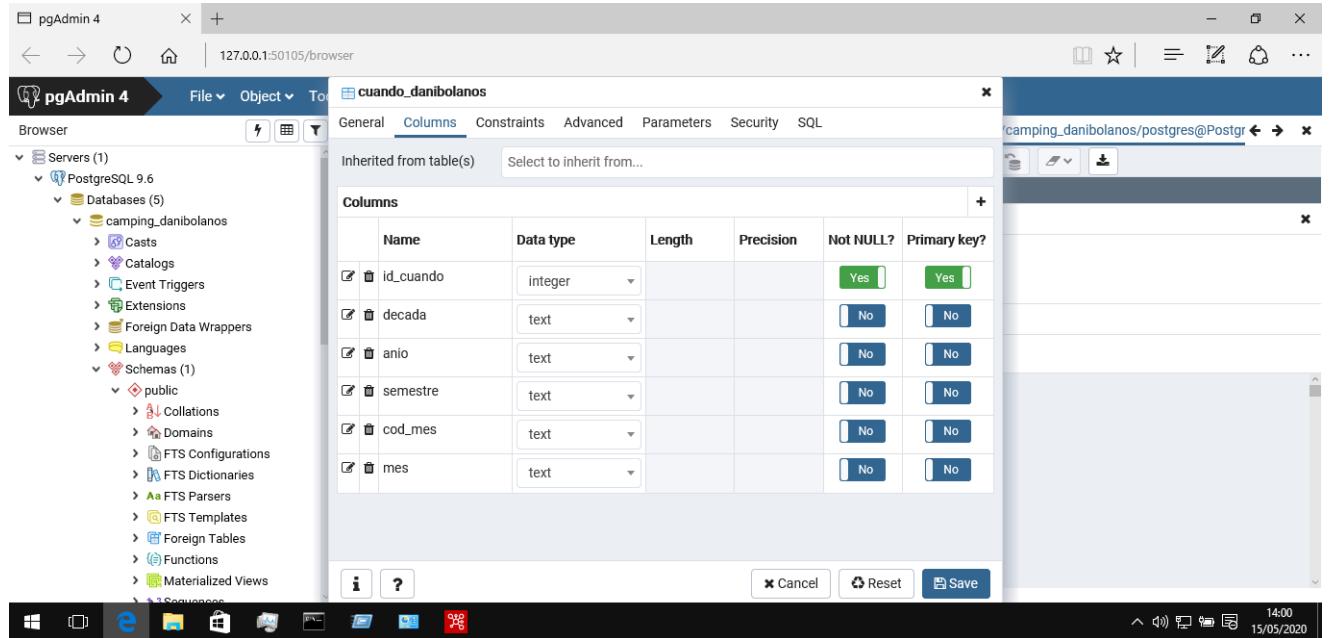
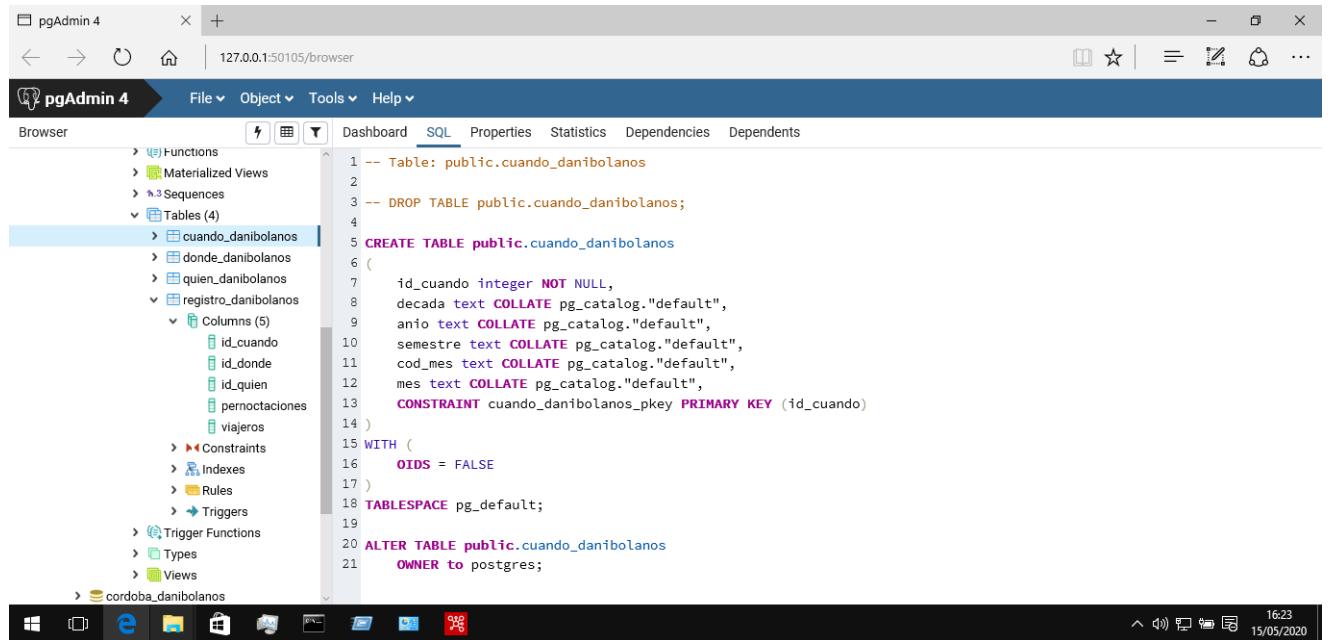


Figura 35: Cambiamos propiedades de **id_cuando**.

Una vez realizadas las modificaciones pertinentes, obtenemos la tabla definida como sentencias *SQL* de la siguiente forma.



```
1 -- Table: public.cuando_danibolanos
2
3 -- DROP TABLE public.cuando_danibolanos;
4
5 CREATE TABLE public.cuando_danibolanos
6 (
7     id_cuando integer NOT NULL,
8     decada text COLLATE pg_catalog."default",
9     anio text COLLATE pg_catalog."default",
10    semestre text COLLATE pg_catalog."default",
11    cod_mes text COLLATE pg_catalog."default",
12    mes text COLLATE pg_catalog."default",
13    CONSTRAINT cuando_danibolanos_pkey PRIMARY KEY (id_cuando)
14 )
15 WITH (
16     OIDS = FALSE
17 )
18 TABLESPACE pg_default;
19
20 ALTER TABLE public.cuando_danibolanos
21     OWNER to postgres;
```

Figura 36: Definición de la tabla **cuando_danibolanos** en sentencias *SQL*.

Para la transformación **camping-ETL-danibolanos-donde** crearemos los siguientes pasos para pasar de la hoja Registro a la tabla **donde_danibolanos**.

Para ello necesitamos quedarnos con los campos **CodMunicipio**, **Municipio**, **CodComarca**, **Comarca**, **CodProvincia**, **Provincia**, **CodCA** y **CA** y completar la tabla con el campo **id_donde**.

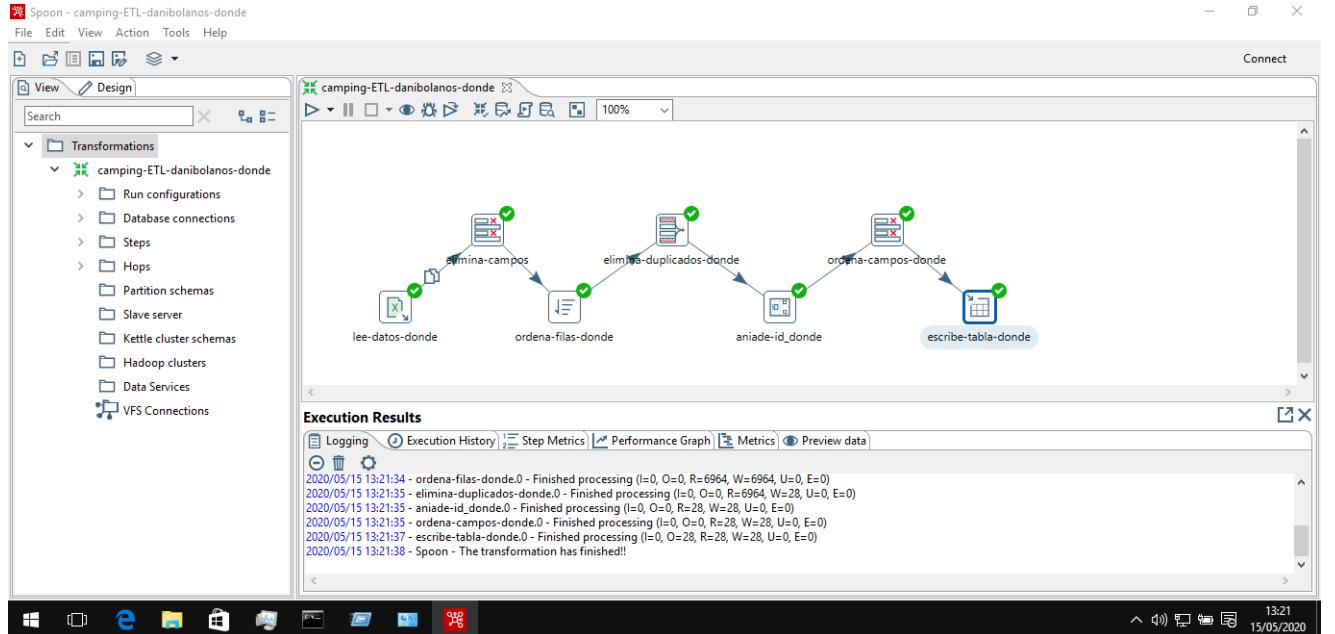


Figura 37: Configuramos los diferentes pasos en la transformación.

De izquierda a derecha se han usado los pasos: *Excel input file*, *Select values*, *Sort rows*, *Unique rows*, *Add sequence*, *Select values* y *Table output*.

* Eliminaremos algunas capturas para las dimensiones **Dónde** y **Quién** que sean redundantes a los pasos ya explicados para la dimensión **Cuándo**.

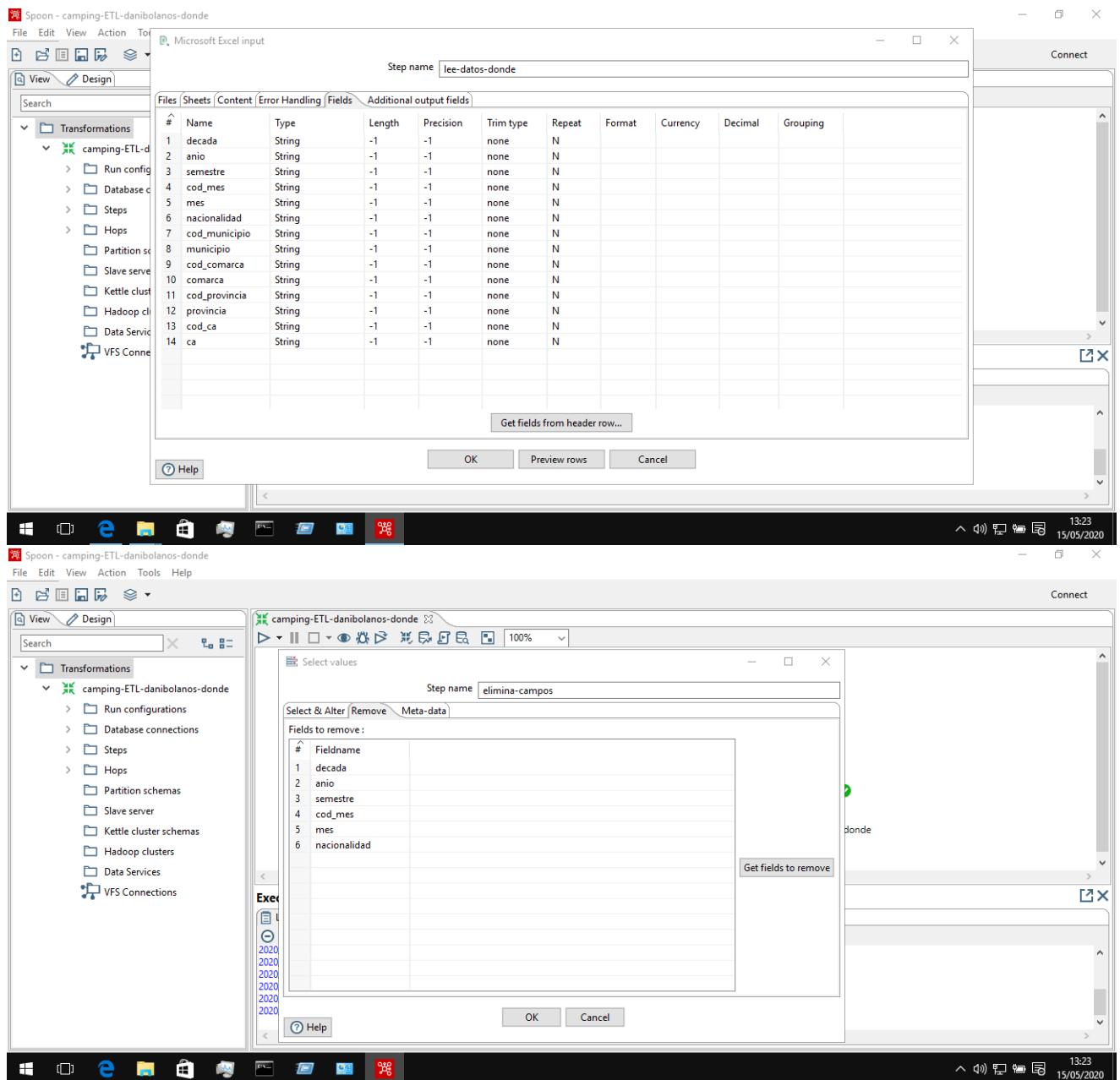


Figura 38: Eliminamos los campos que no nos sirven con el paso *Select values*.

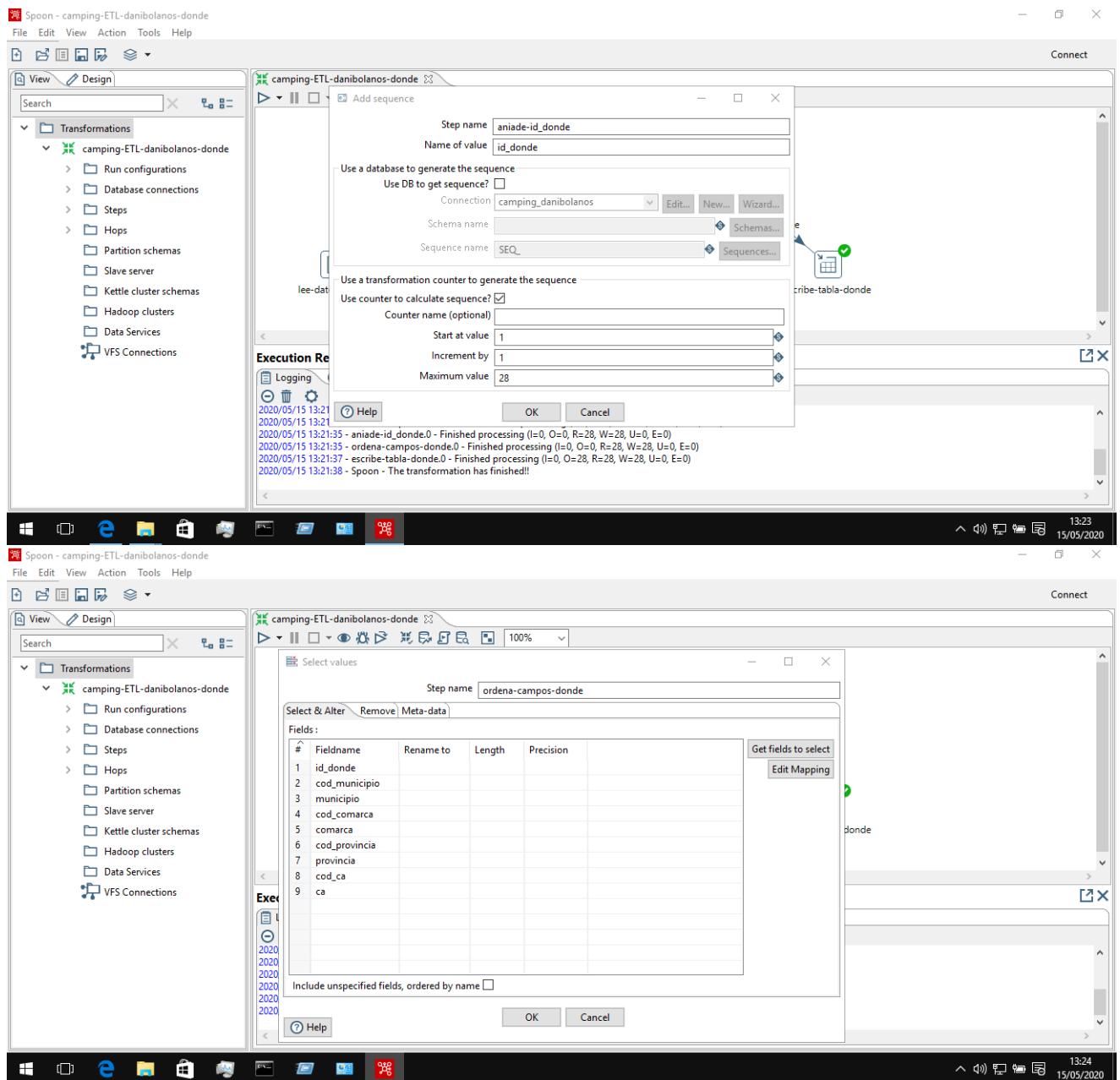


Figura 39: Añadimos el campo **id_donde** y reordenamos los campos.

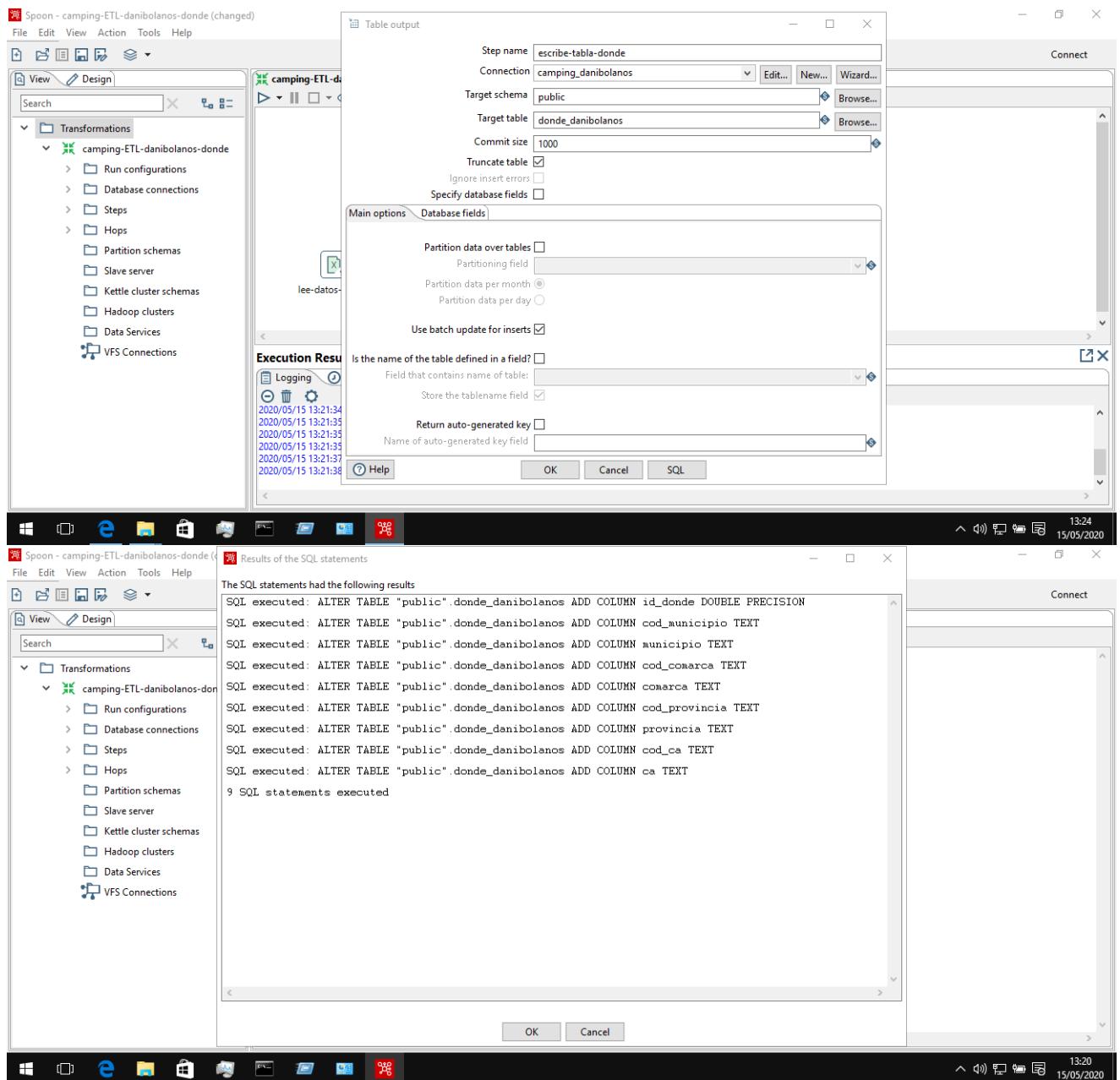


Figura 40: Generamos las sentencias *SQL*.

The screenshot shows two windows of pgAdmin 4. The top window displays the 'Columns' tab for the 'donde_danibolanos' table, listing columns such as id_donde, cod_municipio, municipio, cod_comarca, comarca, cod_provincia, and provincia. The bottom window shows the SQL tab with the CREATE TABLE command for the same table.

Top Window (Columns Tab):

Name	Data type	Length	Precision	Not NULL?	Primary key?
id_donde	integer			Yes	Yes
cod_municipio	text			No	No
municipio	text			No	No
cod_comarca	text			No	No
comarca	text			No	No
cod_provincia	text			No	No
provincia	text			No	No

Bottom Window (SQL Tab):

```

1 -- Table: public.donde_danibolanos
2
3 -- DROP TABLE public.donde_danibolanos;
4
5 CREATE TABLE public.donde_danibolanos
6 (
7     id_donde integer NOT NULL,
8     cod_municipio text COLLATE pg_catalog."default",
9     municipio text COLLATE pg_catalog."default",
10    cod_comarca text COLLATE pg_catalog."default",
11    comarca text COLLATE pg_catalog."default",
12    cod_provincia text COLLATE pg_catalog."default",
13    provincia text COLLATE pg_catalog."default",
14    cod_ca text COLLATE pg_catalog."default",
15    ca text COLLATE pg_catalog."default",
16    CONSTRAINT donde_danibolanos_pkey PRIMARY KEY (id_donde)
17 )
18 WITH (
19     OIDS = FALSE
20 )
21 TABLESPACE pg_default;
22
23 ALTER TABLE public.donde_danibolanos
  
```

Figura 41: Definición de la tabla **donde_danibolanos** en sentencias *SQL*.

Para la transformación **camping-ETL-danibolanos-quien** crearemos los siguientes pasos para pasar de la hoja **Registro** a la tabla **quien_danibolanos**.

Para ello necesitamos quedarnos con el campo **Nacionalidad** y completar la tabla con el campo **id_quien**.

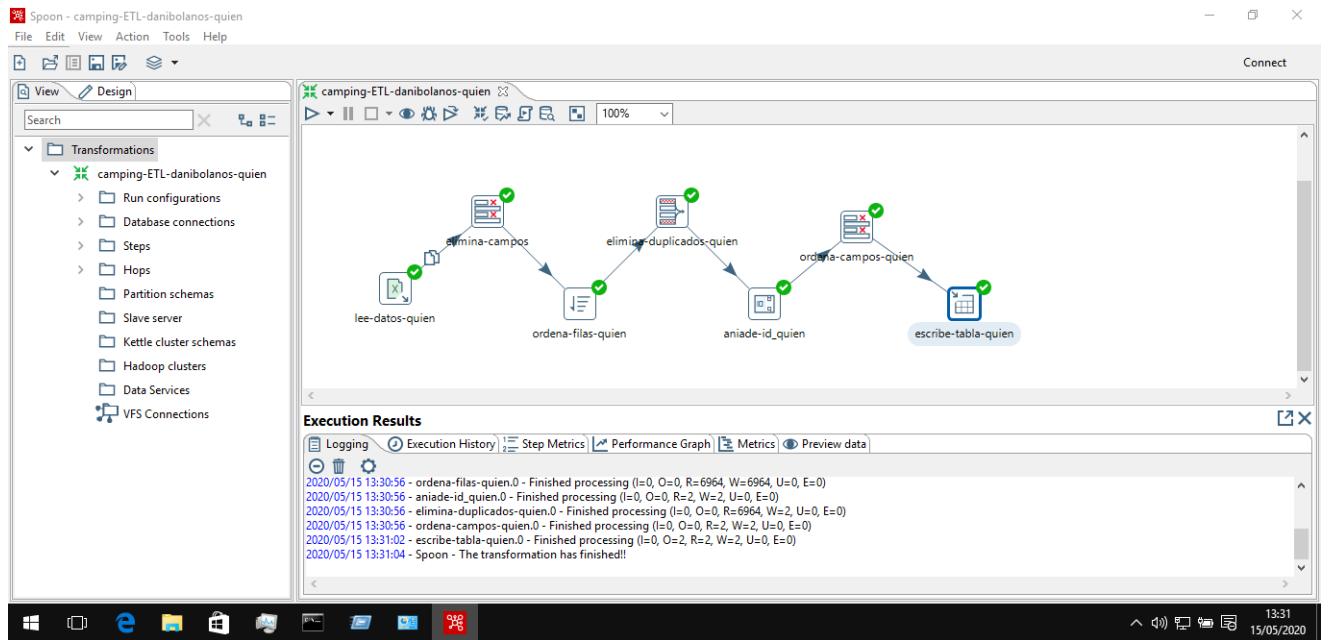


Figura 42: Configuramos los diferentes pasos en la transformación.

De izquierda a derecha se han usado los pasos: *Excel input file*, *Select values*, *Sort rows*, *Unique rows*, *Add sequence*, *Select values* y *Table output*.

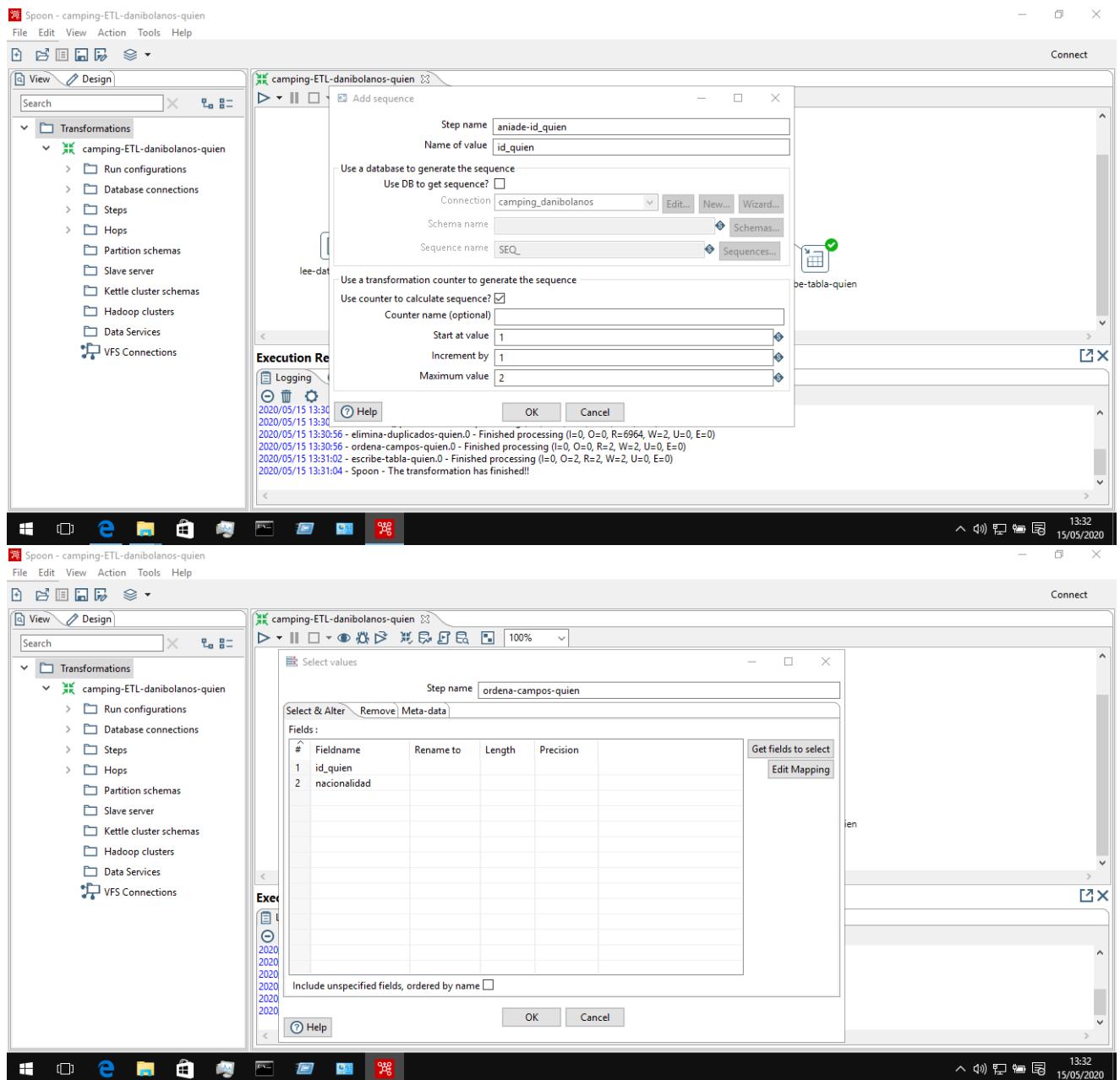


Figura 43: Añadimos el campo **id_quien** y reordenamos los campos.

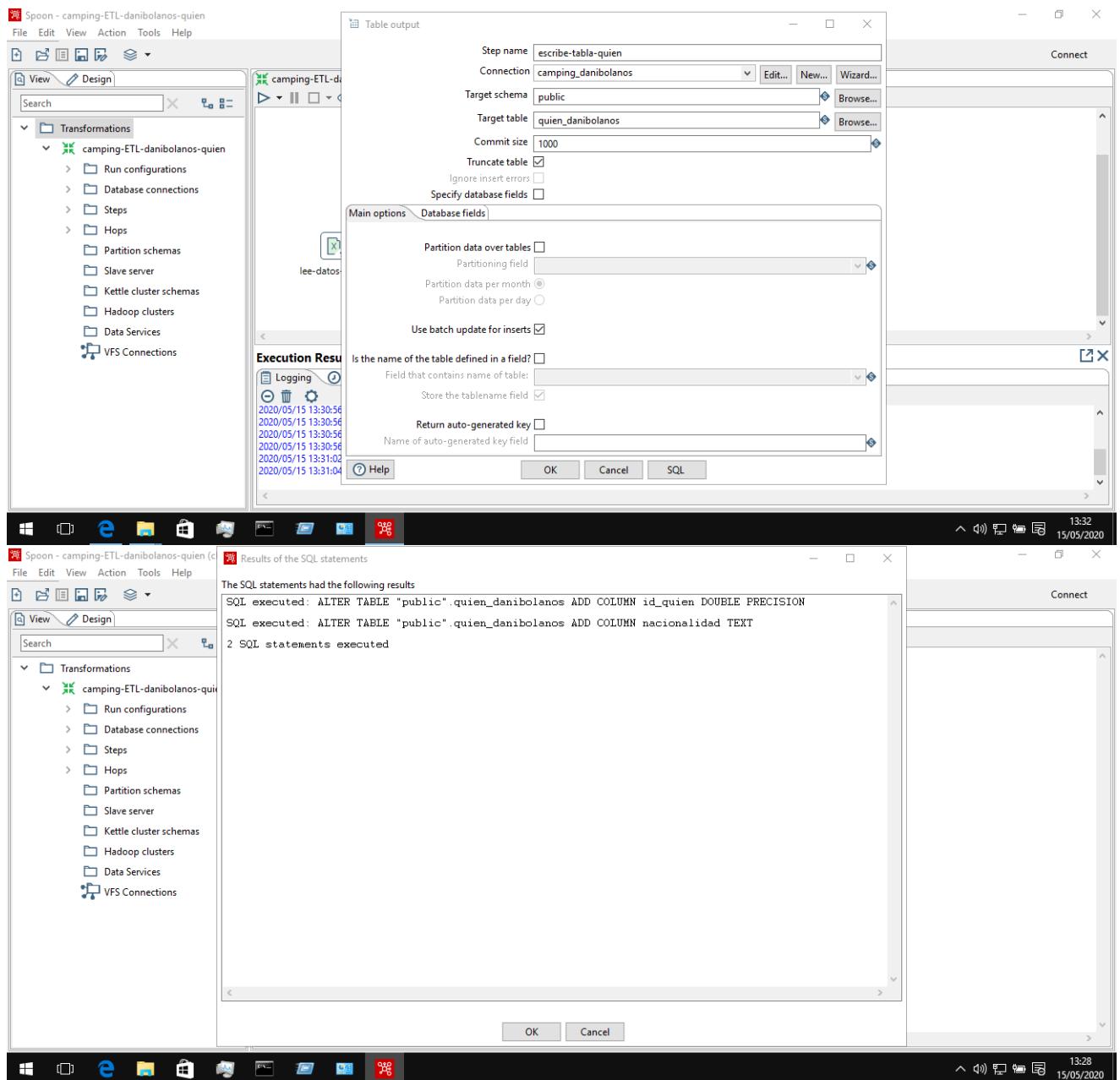


Figura 44: Generamos las sentencias *SQL*.

The screenshot shows the pgAdmin 4 interface with two main windows. The top window is a configuration dialog for the table 'quien_danibolanos'. It has tabs for General, Columns, Constraints, Advanced, Parameters, Security, and SQL. The 'Columns' tab is selected, displaying two columns: 'id_quien' (integer, primary key) and 'nacionalidad' (text). The bottom window shows the pgAdmin 4 browser interface with the SQL tab selected. The left sidebar lists databases, tables, and other objects. The main area displays the SQL code for creating the 'quien_danibolanos' table:

```

1 -- Table: public.quien_danibolanos
2
3 -- DROP TABLE public.quien_danibolanos;
4
5 CREATE TABLE public.quien_danibolanos
6 (
7     id_quien integer NOT NULL,
8     nacionalidad text COLLATE pg_catalog."default",
9     CONSTRAINT quien_danibolanos_pkey PRIMARY KEY (id_quien)
10 )
11 WITH (
12     OIDS = FALSE
13 )
14 TABLESPACE pg_default;
15
16 ALTER TABLE public.quien_danibolanos
17     OWNER to postgres;

```

Figura 45: Definición de la tabla **quien_danibolanos** en sentencias *SQL*.

Para la transformación **camping-ETL-danibolanos-registro** crearemos los siguientes pasos para pasar de la hoja **Registro** a la tabla de hechos **registro_danibolanos**.

Para ello necesitamos quedarnos con los campos **Viajeros** y **Pernoctaciones** y completar la tabla con los campos **id_cuando**, **id_donde** e **id_quien**.

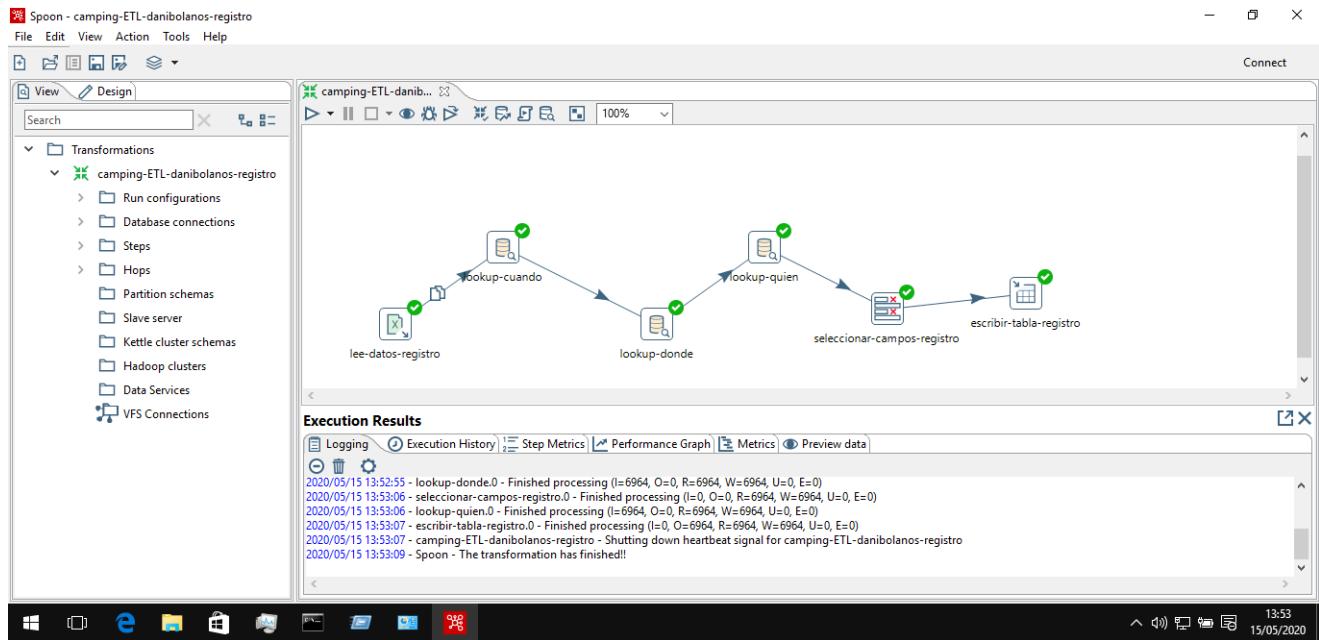


Figura 46: Configuramos los diferentes pasos en la transformación.

De izquierda a derecha se han usado los pasos: *Excel input file*, *Database lookup*, *Database lookup*, *Select values* y *Table output*.

Con el paso *Excel input file*, seleccionamos todos los campos y ponemos los nombres en minúscula, sin tildes y utilizando el criterio *camel_case*.

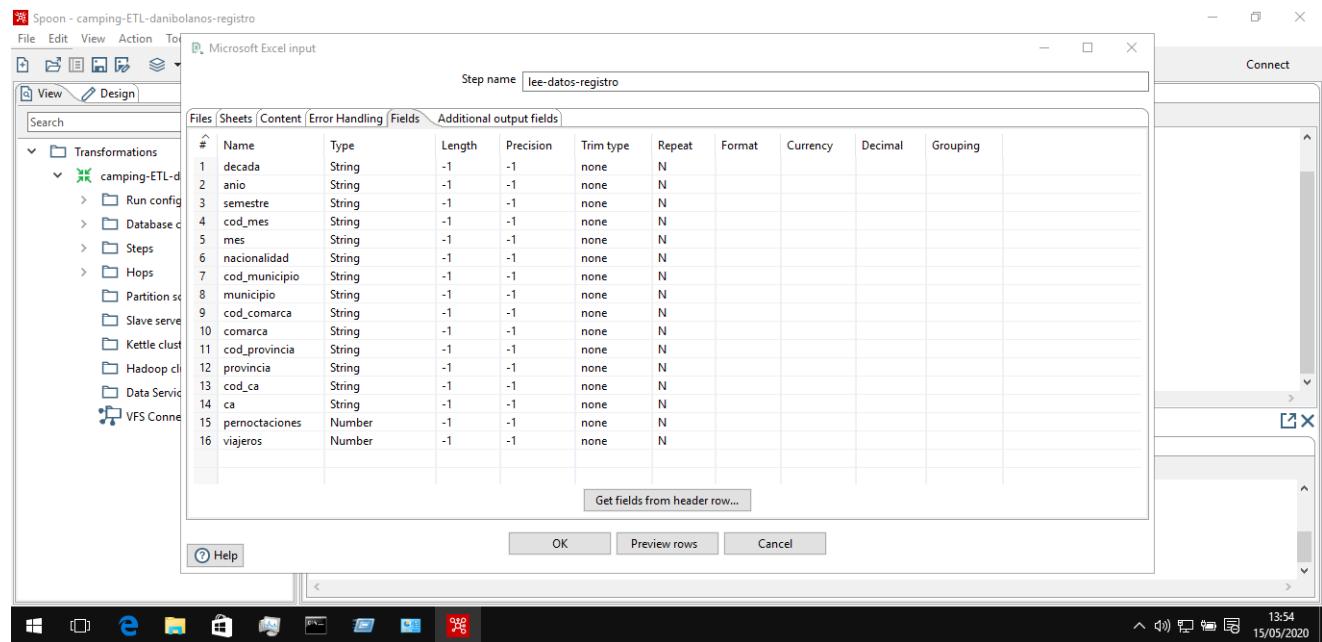


Figura 47: Campos de hoja **Registro** con los criterios especificados.

Haciendo uso del paso *Database lookup*, combinaremos las columnas especificadas en el primer recuadro comparando (*Comparator* '=') los campos de la tabla en *pgAdmin* (*Table field*) con los datos leídos en el paso anterior (*Field1*).

En el segundo recuadro, especificamos el campo que queremos añadir una vez reactualizada la reunión de las filas. En este caso, **id_cuando**.

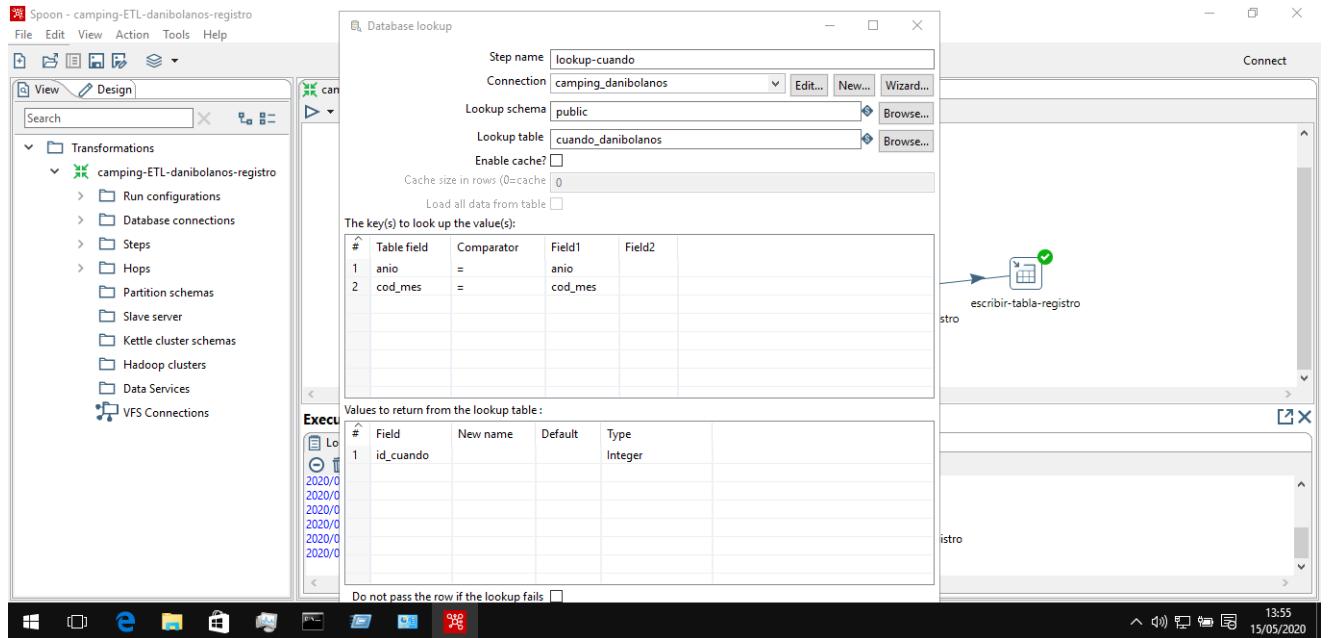


Figura 48: Paso *Database lookup* para la dimensión Cuándo.

* Se puede observar que no es necesario especificar que las filas a combinar coincidan en todos los campos, ya que algunos como los códigos determinan de forma única los nombres a los que se refieren (p.ej: CodMes → Mes).

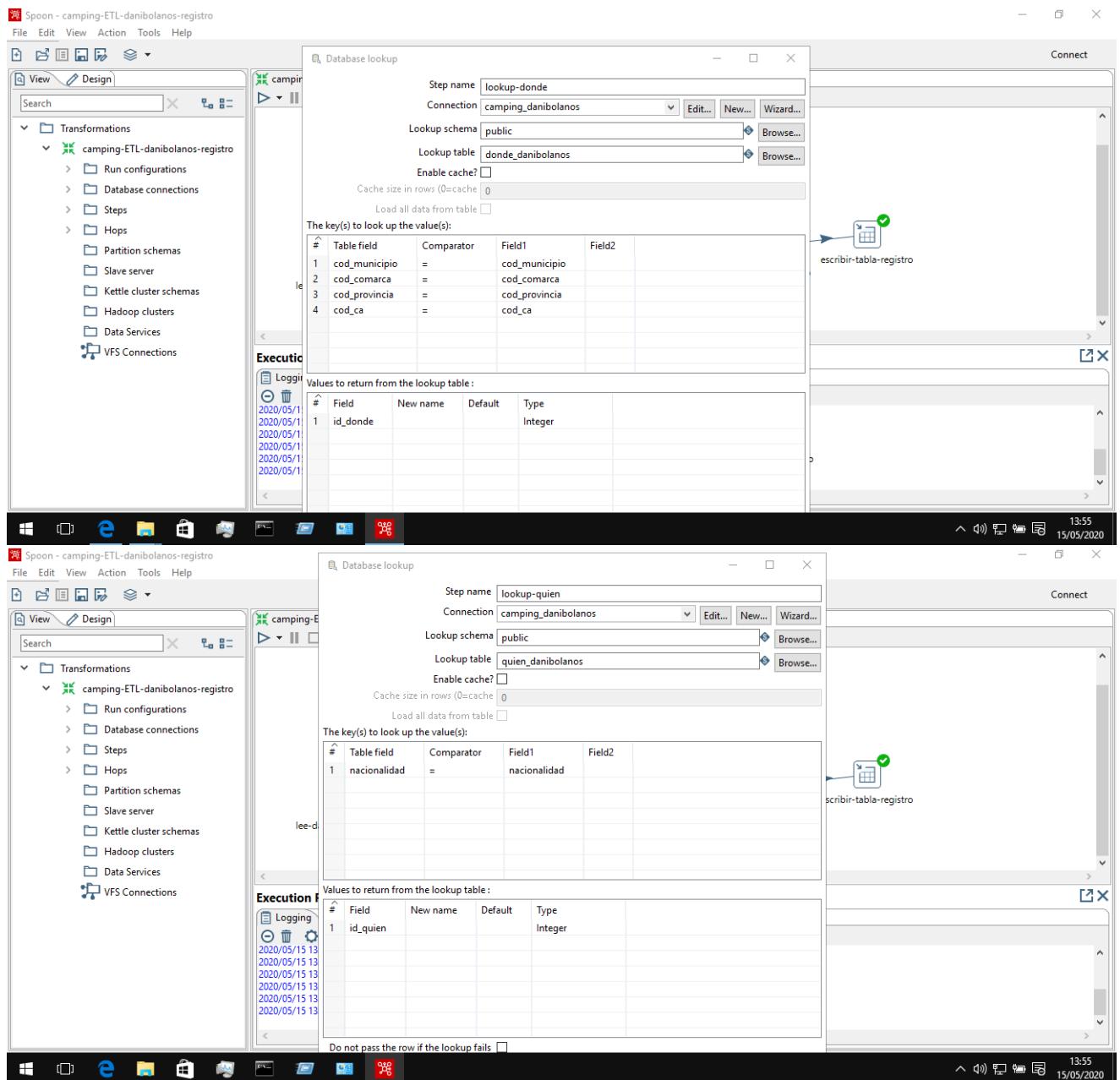


Figura 49: Paso *Database lookup* para las dimensiones **Dónde** y **Quién**.

Reordenamos y seleccionamos los campos que formarán la tabla de hechos.

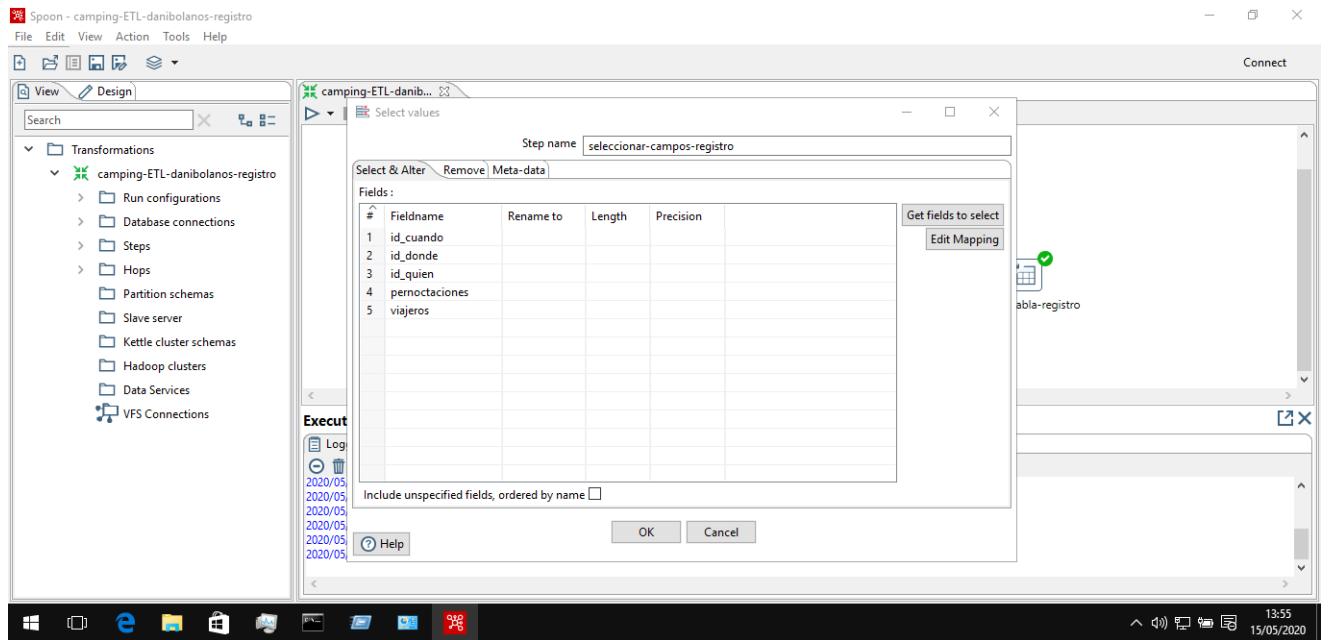


Figura 50: Reordenamos los campos con el paso *Select values*.

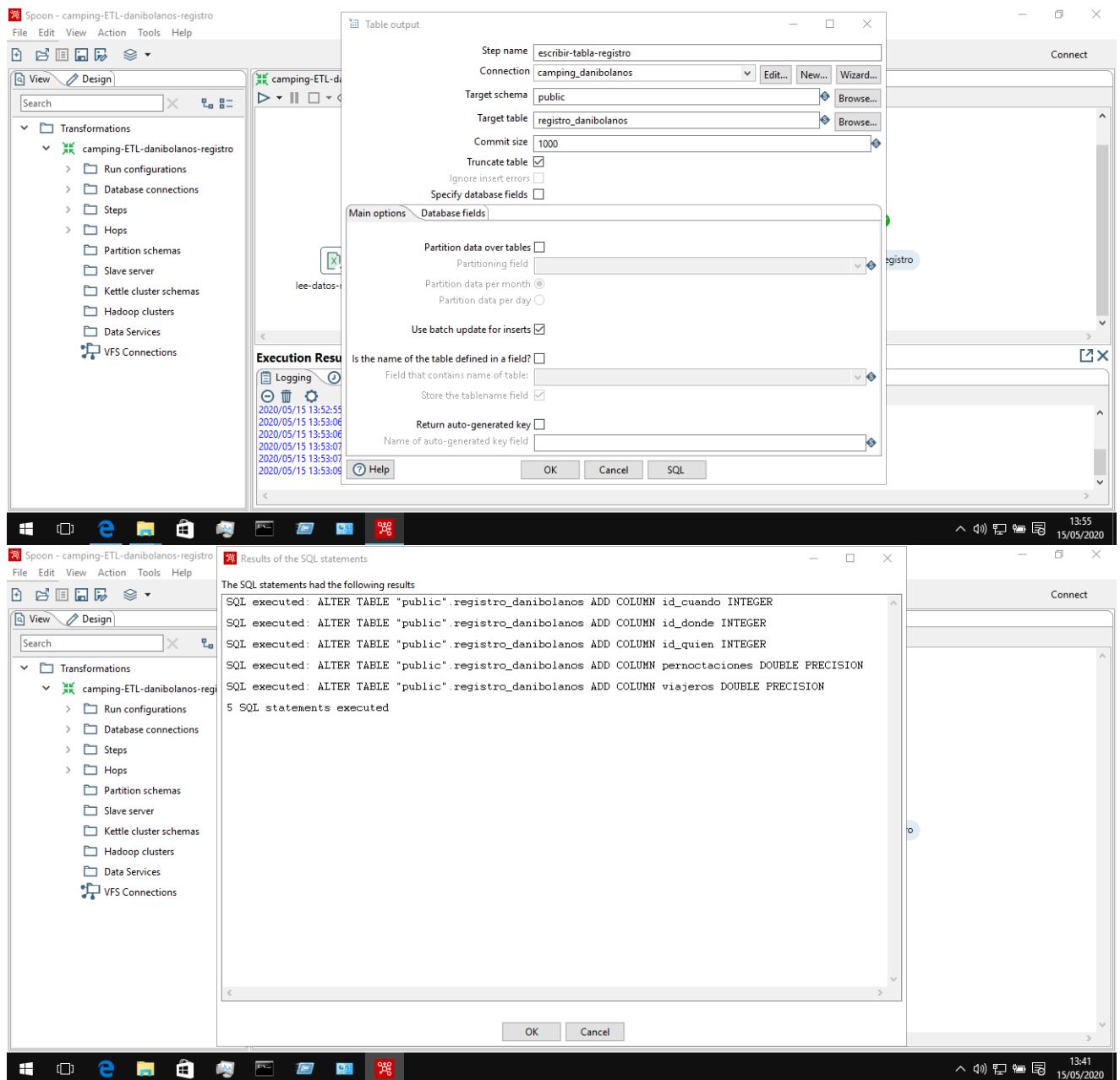


Figura 51: Generamos las sentencias *SQL*.

The figure consists of two vertically stacked screenshots of the pgAdmin 4 interface, version 4, running in a browser window.

Screenshot 1: Table Definition

This screenshot shows the 'registro_danibolanos' table being created. The 'Columns' tab is selected, displaying five columns:

Name	Data type	Length	Precision	Not NULL?	Primary key?
id_cuando	integer			Yes	Yes
id_donde	integer			Yes	Yes
id_quien	integer			Yes	Yes
pernoctaciones	integer			No	No
viajeros	integer			No	No

Screenshot 2: Constraint Definition

This screenshot shows the 'Constraints' tab for the 'registro_danibolanos' table. It lists three foreign key constraints:

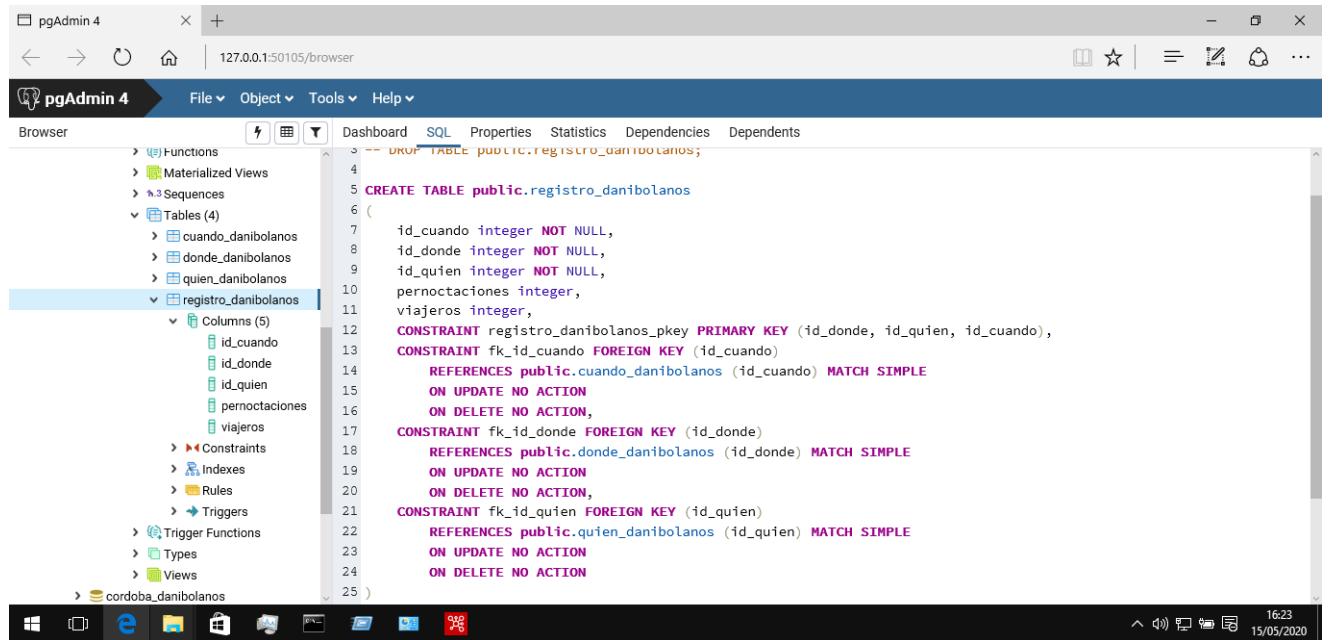
Name	Columns
fk_id_cuando	(id_cuando) -> (id_cuando)
fk_id_donde	(id_donde) -> (id_donde)
fk_id_quien	(id_quien) -> (id_quien)

The 'fk_id_quien' constraint is currently selected. Below it, the 'Columns' tab is selected, showing the details for the 'id_quien' column:

Local	Referenced
id_quien	id_quien

Figura 52: Definimos los tipos, las claves primarias y las claves externas.

Una vez realizadas las modificaciones pertinentes y definidas las claves primarias y externas, obtenemos la tabla de hechos definida como sentencias *SQL*.



The screenshot shows the pgAdmin 4 interface with the 'Browser' tab selected. In the left sidebar, under 'Tables (4)', the 'registro_danibolanos' table is expanded, showing its columns: id_cuando, id_donde, id_quien, pernoctaciones, and viajeros. The 'Properties' tab is active at the top, displaying the following SQL code:

```
-- DROP TABLE public.registro_danibolanos;
CREATE TABLE public.registro_danibolanos
(
    id_cuando integer NOT NULL,
    id_donde integer NOT NULL,
    id_quien integer NOT NULL,
    pernoctaciones integer,
    viajeros integer,
    CONSTRAINT registro_danibolanos_pkey PRIMARY KEY (id_donde, id_quien, id_cuando),
    CONSTRAINT fk_id_cuando FOREIGN KEY (id_cuando)
        REFERENCES public.cuando_danibolanos (id_cuando) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT fk_id_donde FOREIGN KEY (id_donde)
        REFERENCES public.donde_danibolanos (id_donde) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT fk_id_quien FOREIGN KEY (id_quien)
        REFERENCES public.quien_danibolanos (id_quien) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

Figura 53: Definición de la tabla de hechos en sentencias *SQL*.

Definimos un *job* con las cuatro *transformaciones* y lo ejecutaremos para cargar los datos del Excel con la estructura que hemos proporcionado a la BD creada.

Podemos ver como se ha ejecutado correctamente y sin mostrar ningún error.

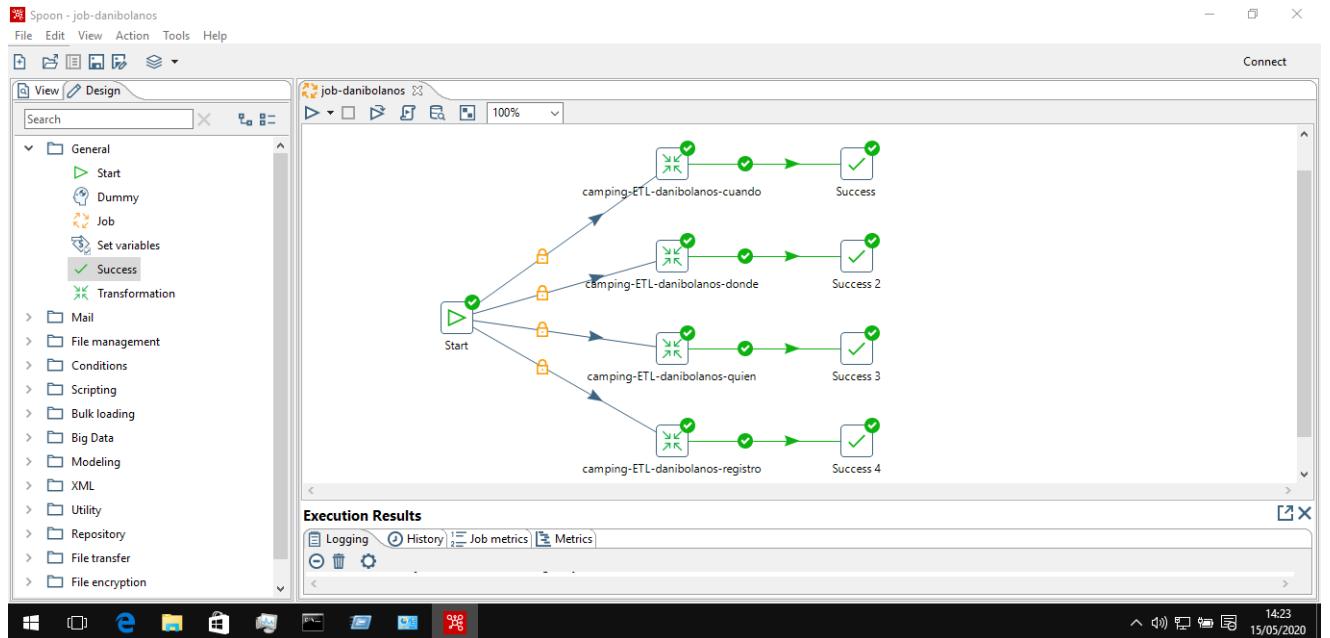


Figura 54: Definición del *job* para cargar los datos a la BD.

The figure consists of two vertically stacked screenshots of the pgAdmin 4 interface, both displaying the contents of PostgreSQL tables.

Screenshot 1 (Top): This screenshot shows the contents of the 'cuando_danibolanos' table. The table has columns: id_cuando, decada, anio, semestre, cod_mes, and mes. The data is as follows:

	id_cuando	decada	anio	semestre	cod_mes	mes
1	1	2000	2005	1-semestre	01	01-e...
2	2	2000	2005	1-semestre	02	02-fe...
3	3	2000	2005	1-semestre	03	03-m...
4	4	2000	2005	1-semestre	04	04-a...
5	5	2000	2005	1-semestre	05	05-m...
6	6	2000	2005	1-semestre	06	06-ju...
7	7	2000	2005	2-semestre	07	07-ju...
8	8	2000	2005	2-semestre	08	08-a...
9	9	2000	2005	2-semestre	09	09-s...

Screenshot 2 (Bottom): This screenshot shows the contents of the 'donde_danibolanos' table. The table has columns: id_donde, cod_municipio, municipio, cod_comarca, comarca, cod_provincia, provincia, cod_ca, and ca. The data is as follows:

	id_donde	cod_municipio	municipio	cod_comarca	comarca	cod_provincia	provincia	cod_ca	ca
1	1	03031	Benidorm	304	Central	03	Alacant/Ali...	10	Com...
2	2	03050	Campello, el	304	Central	03	Alacant/Ali...	10	Com...
3	3	03065	Eiche/Elx	305	Meridional	03	Alacant/Ali...	10	Com...
4	4	03139	Villajoyosa/...	304	Central	03	Alacant/Ali...	10	Com...
5	5	04064	Mojácar	403	Bajo Almería	04	Almería	01	Anda...
6	6	11014	Conil de la ...	1102	Costa Nor...	11	Cádiz	01	Anda...
7	7	11035	Tarifa	1105	Campo De ...	11	Cádiz	01	Anda...
8	8	12028	Benicasim/...	1206	La Plana	12	Castelló/C...	10	Com...
9	9	12085	Oropesa de...	1206	La Plana	12	Castelló/C...	10	Com...

Figura 55: Datos para las tablas **cuando_danibolanos** y **donde_danibolanos**.

Top Session: public.quien_danibolanos

```
1 SELECT * FROM public.quien_danibolanos
2
```

	id_quien	nacionalidad
1	1	Española
2	2	Extranjera

Bottom Session: public.registro_danibolanos

```
1 SELECT * FROM public.registro_danibolanos
2
```

	id_cuando	id_donde	id_quien	pernoctaciones	viajeros
1	1	1	1	1020	80
2	13	1	1	1634	68
3	2	1	1	929	42
4	3	1	1	4298	1081
5	4	1	1	2661	594
6	5	1	1	3562	1016
7	6	1	1	5682	1275
8	7	1	1	13360	3378
9	8	1	1	29621	6627

Figura 56: Datos para las tablas **quien_danibolanos** y **registro_danibolanos**.

6. Componente OLAP.

Mondrian es un motor OLAP escrito en *Java* que ejecuta consultas MDX sobre datos en una BD relacional y devuelve los resultados en formato multidimensional. Como BD relacional usaremos *PostgreSQL*.

Adicionalmente, usaremos *JRubik* para generar consultas MDX, enviárselas a *Mondrian* y presentar adecuadamente los resultados devueltos en formato multidimensional. Para generar el esquema multidimensional en XML usaremos *Schema Workbench*.

Partiremos de la BD **camping_danibolanos** que hemos definido y crearemos un nuevo esquema en *Schema Workbench*.

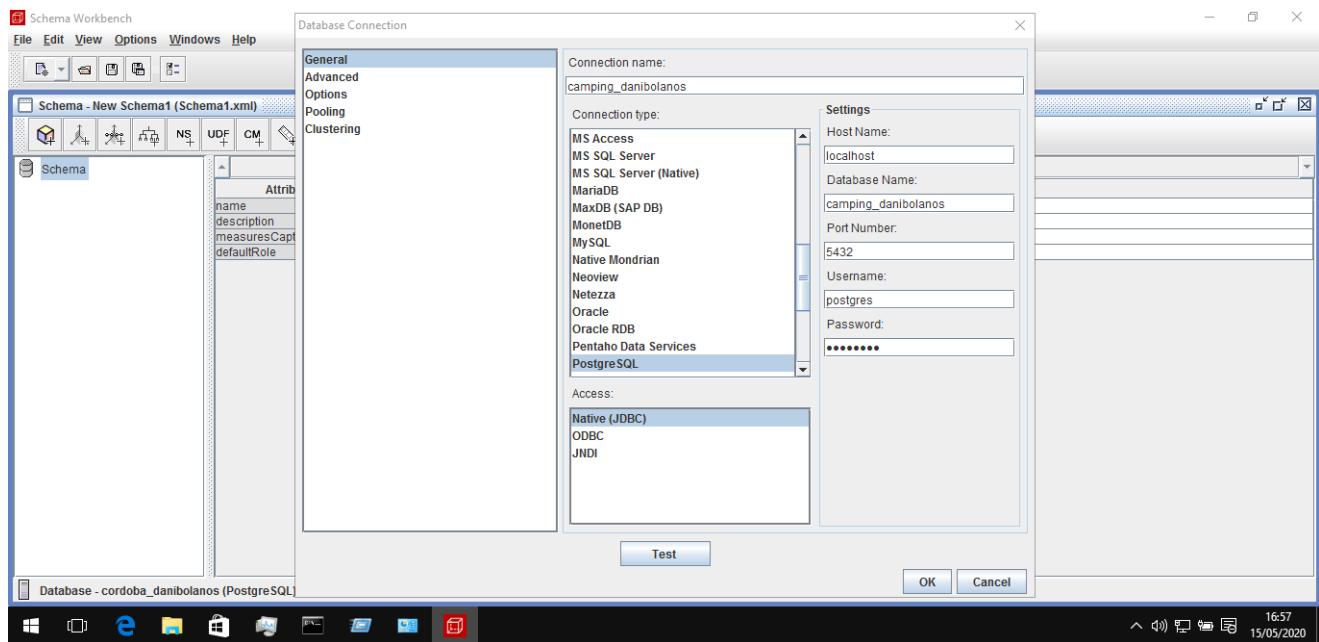


Figura 57: Conexión con la BD.

Pulsamos sobre [«Options», «Connection»] para definir la conexión con la BD que habíamos definido.

Creamos un nuevo esquema con [«File», «New», «Schema»] y definimos su campo 'name' como **camping_danibolanos**.

Una vez definido el esquema, dentro de este, definiremos el cubo con el mismo nombre.

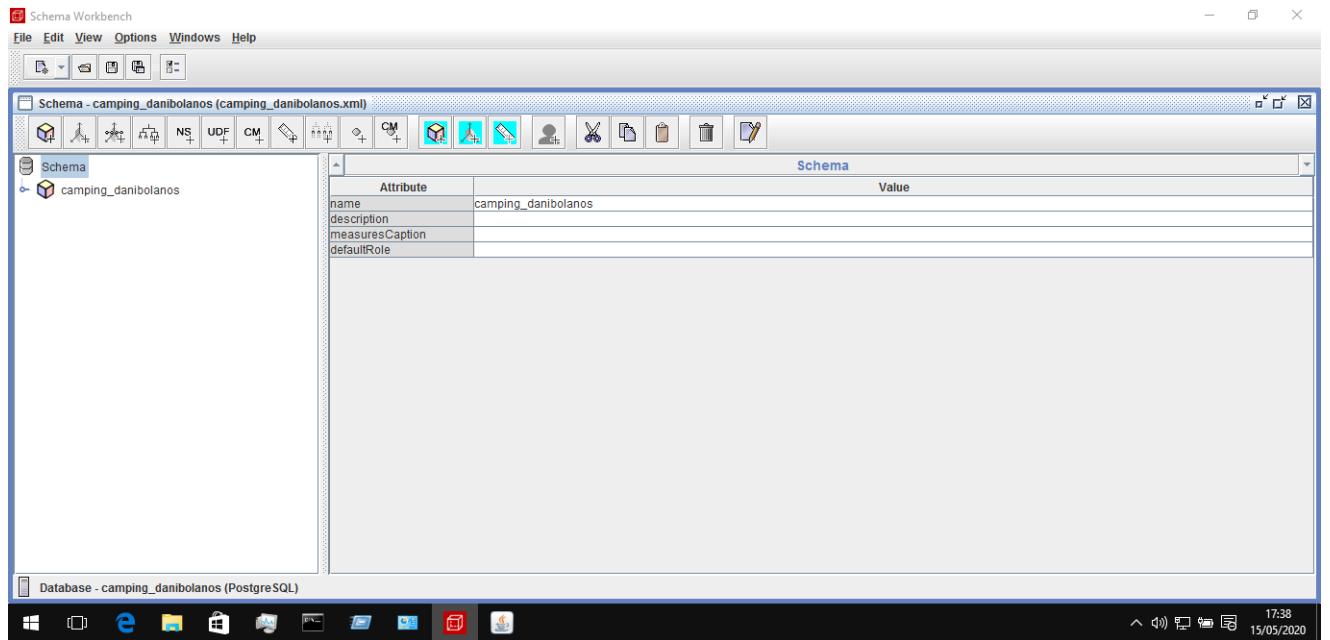


Figura 58: Definición del esquema y el cubo.

He realizado la definición completa del cubo utilizando las diferentes Herramientas proporcionadas por *Schema Workbench*.

A continuación se irán mostrando capturas de la definición de cada paso.

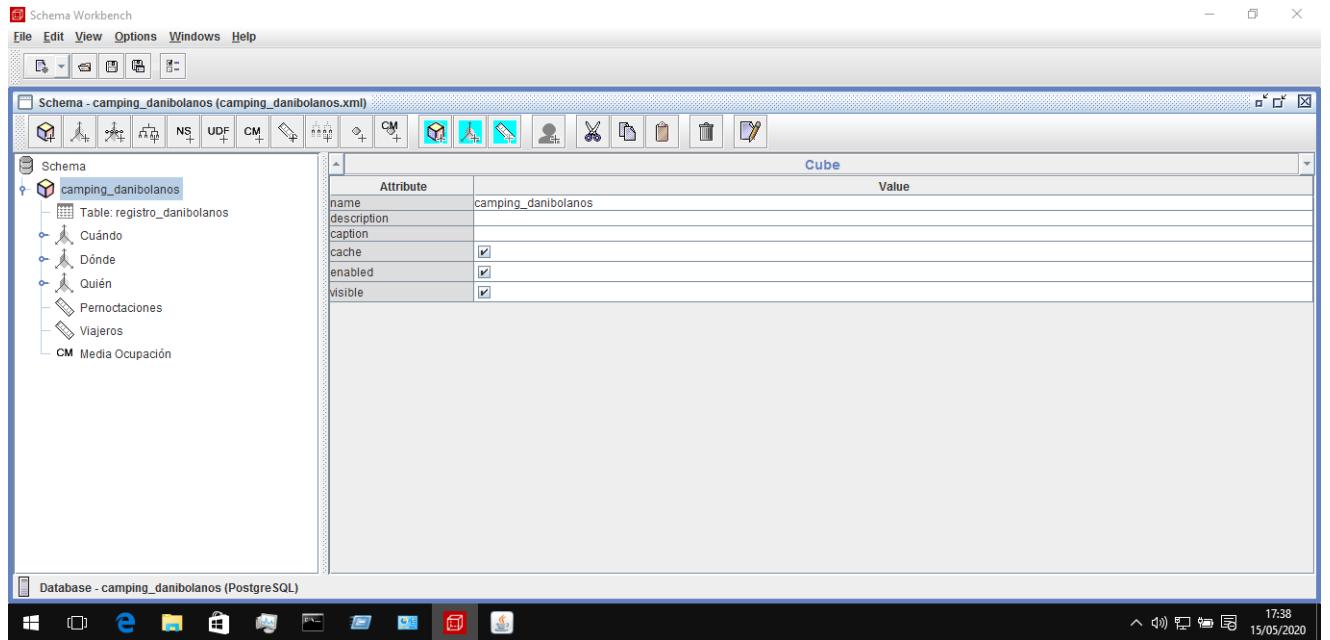


Figura 59: Definición del cubo completo.

En el menú de la izquierda se muestra el resultado final.

Seleccionamos la opción «Add table» para añadir al cubo la tabla de hechos.

En nuestro ejemplo, corresponde con la tabla **registro_danibolanos** contenida en el esquema **public** en la BD.

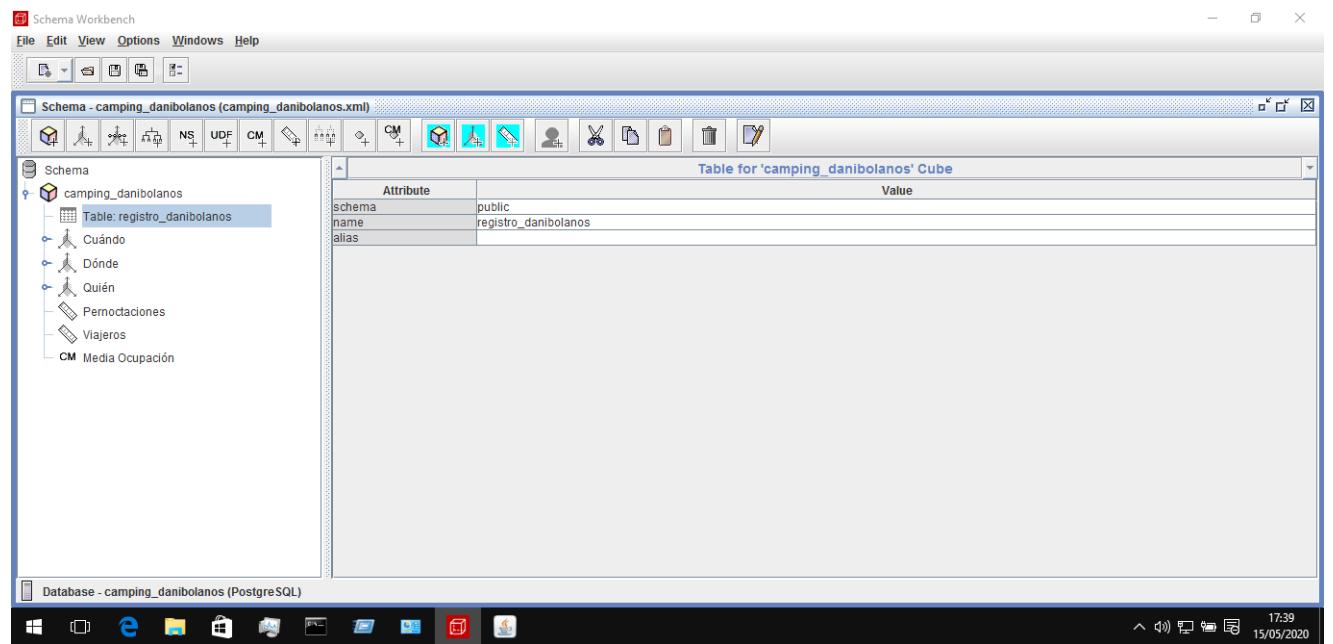


Figura 60: Definición de la tabla de hechos.

Estando situados en el cubo, definimos una nueva dimensión mediante la operación «Add dimension».

Aunque la dimensión es relativa al tiempo, como estamos utilizando campos como **Semestre** o **Mes** con un formato distinto al especificado por *Schema Workbench*, utilizaremos el tipo «StandardDimension».

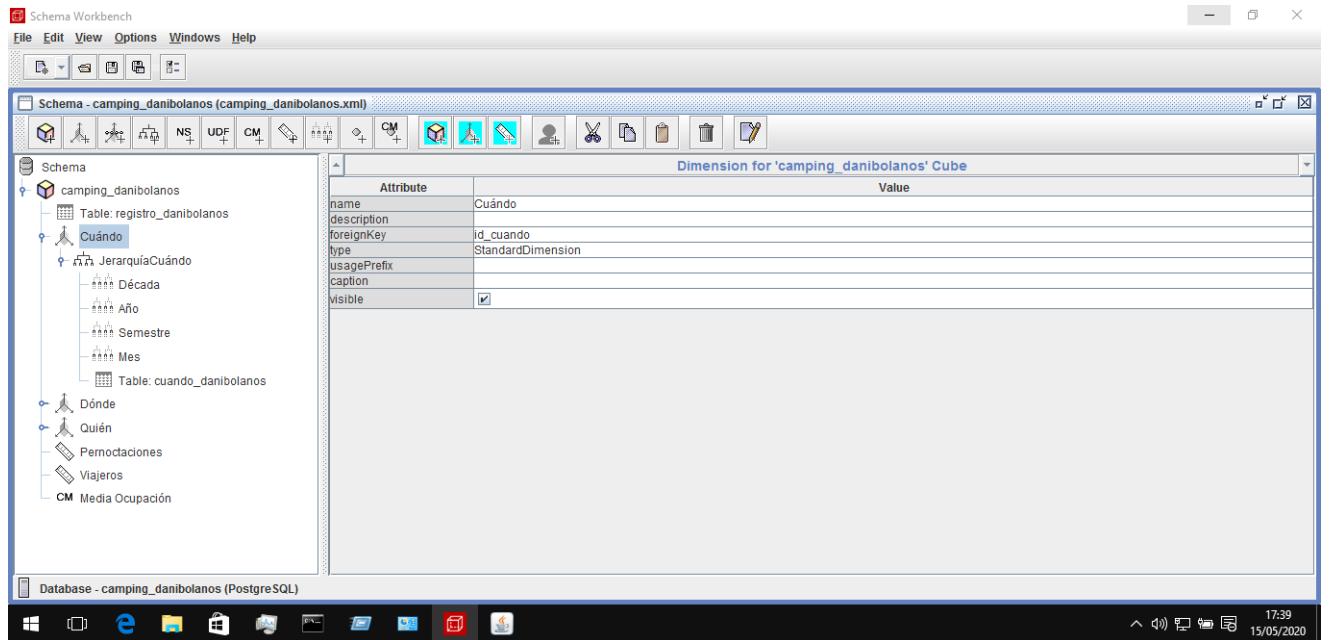


Figura 61: Definición de la dimensión Cuándo.

La jerarquía utilizada para esta dimensión será:

- **JerarquíaCuándo: Década, Año, Semestre, Mes.**

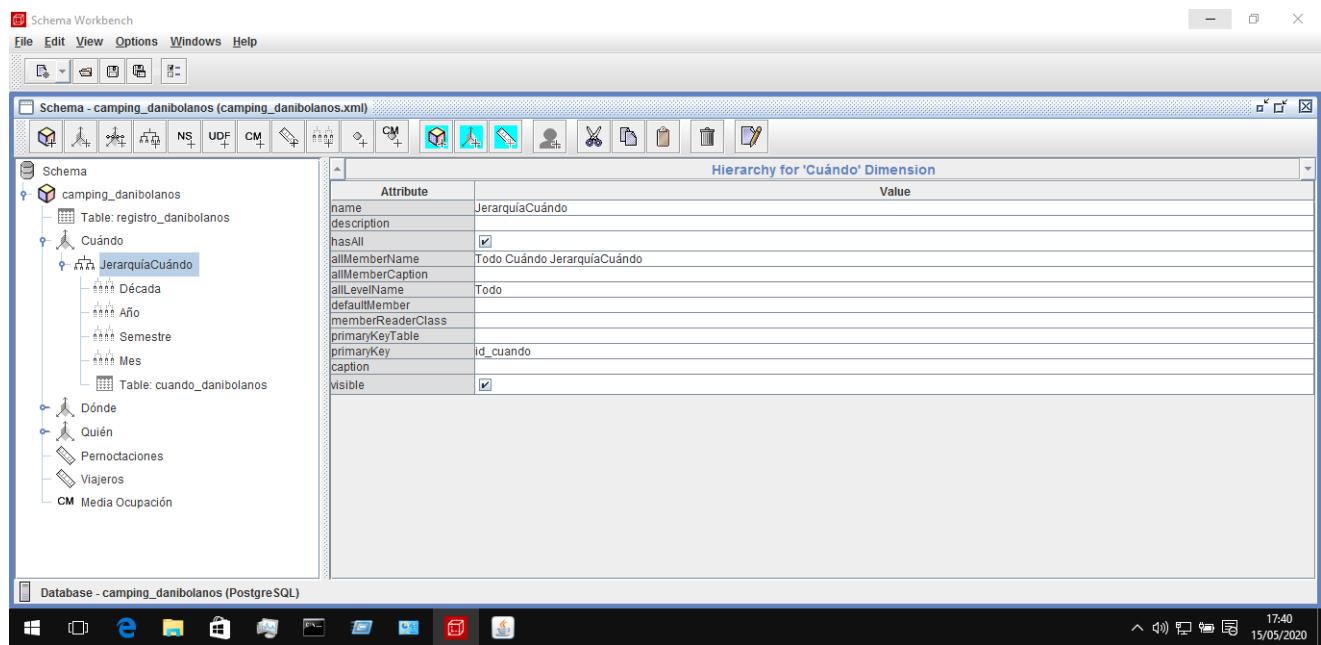


Figura 62: Definición de las jerarquías en la dimensión **Cuándo**.

En este caso vemos como se define el nivel **Década** donde especificamos sus valores teniendo en cuenta su tipo, columna a la que hace referencia...

Su **levelType** será *Regular* y marcaremos la casilla de **uniqueMembers** al ser el nivel más alto de la jerarquía.

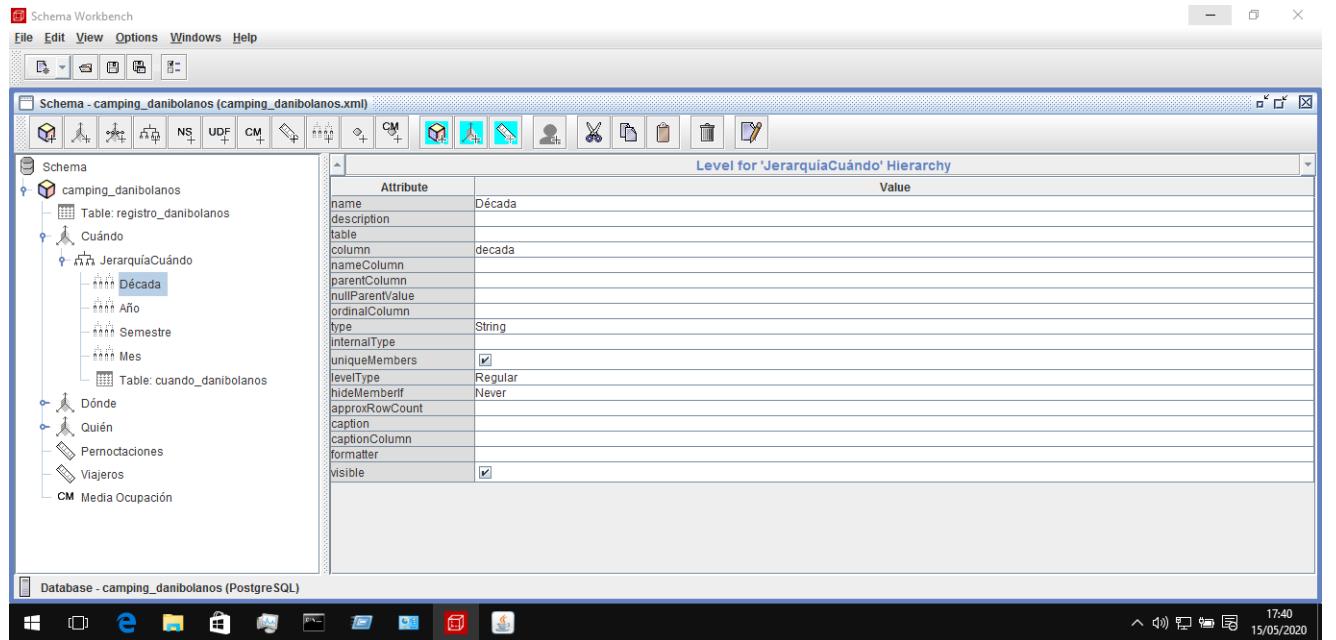


Figura 63: Definición de los niveles de **JerarquíaCuándo**.

Dentro de cada jerarquía, debemos definir una tabla de referencia a la dimensión correspondiente, en este caso a **cuando_danibolanos** de la BD.

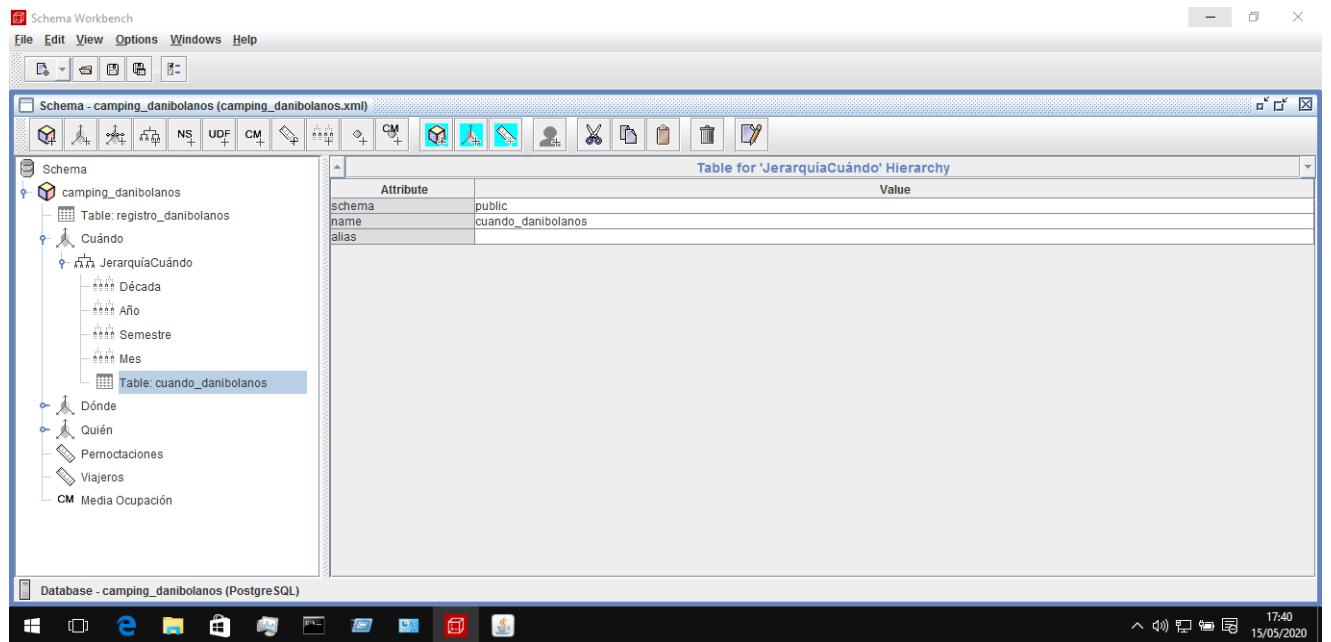


Figura 64: Definición de la tabla **cuando_danibolanos** para la jerarquía.

Estando situados en el cubo, definimos la nueva dimensión **Dónde**.

Al igual que la anterior su tipo será «StandardDimension».

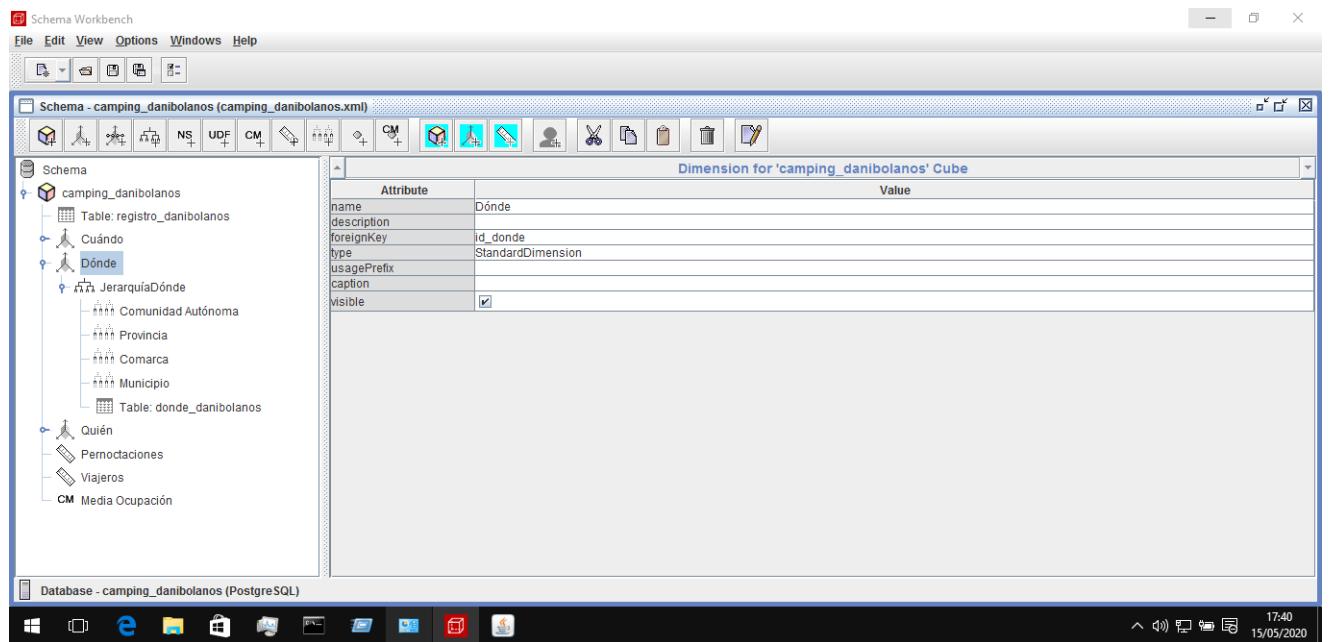


Figura 65: Definición de la dimensión **Dónde**.

La jerarquía utilizada para esta dimensión será:

- **JerarquíaDónde: Comunidad Autónoma, Provincia, Comarca, Municipio.**

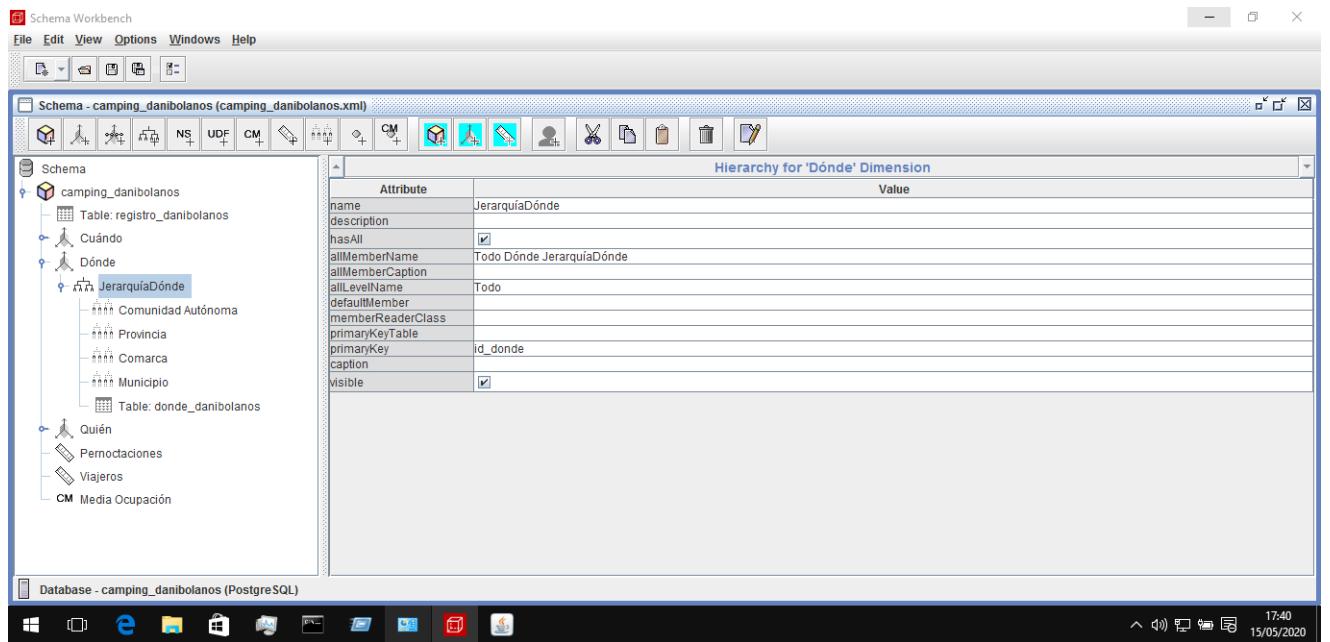


Figura 66: Definición de las jerarquías en la dimensión Dónde.

The screenshot displays two instances of the Schema Workbench application, both showing the configuration of a hierarchy named 'JerarquíaDónde'.

Top Window (Level for 'JerarquíaDónde' Hierarchy):

Attribute	Value
name	Comunidad Autónoma
description	
table	
column	ca
nameColumn	
parentColumn	
nullParentValue	
ordinalColumn	
type	String
internalType	
uniqueMembers	<input checked="" type="checkbox"/>
levelType	Regular
hideMemberIf	Never
approxRowCount	
caption	
captionColumn	
formatter	
visible	<input checked="" type="checkbox"/>

Bottom Window (Level for 'JerarquíaDónde' Hierarchy):

Attribute	Value
name	Municipio
description	
table	
column	municipio
nameColumn	
parentColumn	
nullParentValue	
ordinalColumn	
type	String
internalType	
uniqueMembers	<input type="checkbox"/>
levelType	Regular
hideMemberIf	Never
approxRowCount	
caption	
captionColumn	
formatter	
visible	<input checked="" type="checkbox"/>

The left pane of both windows shows a hierarchical tree structure of schema elements, including tables like 'registro_danibolanos', and categories like 'Cuándo', 'Dónde', 'Quién', 'Pernocaciones', and 'Viajeros'. The 'Dónde' category is expanded to show 'JerarquíaDónde' which further branches into 'Comunidad Autónoma', 'Provincia', 'Comarca', and 'Municipio'.

Figura 67: Definición de los niveles de **JerarquíaDónde**.

Definimos la tabla de referencia a la dimensión correspondiente, en este caso a **donde_danibolanos** de la BD.

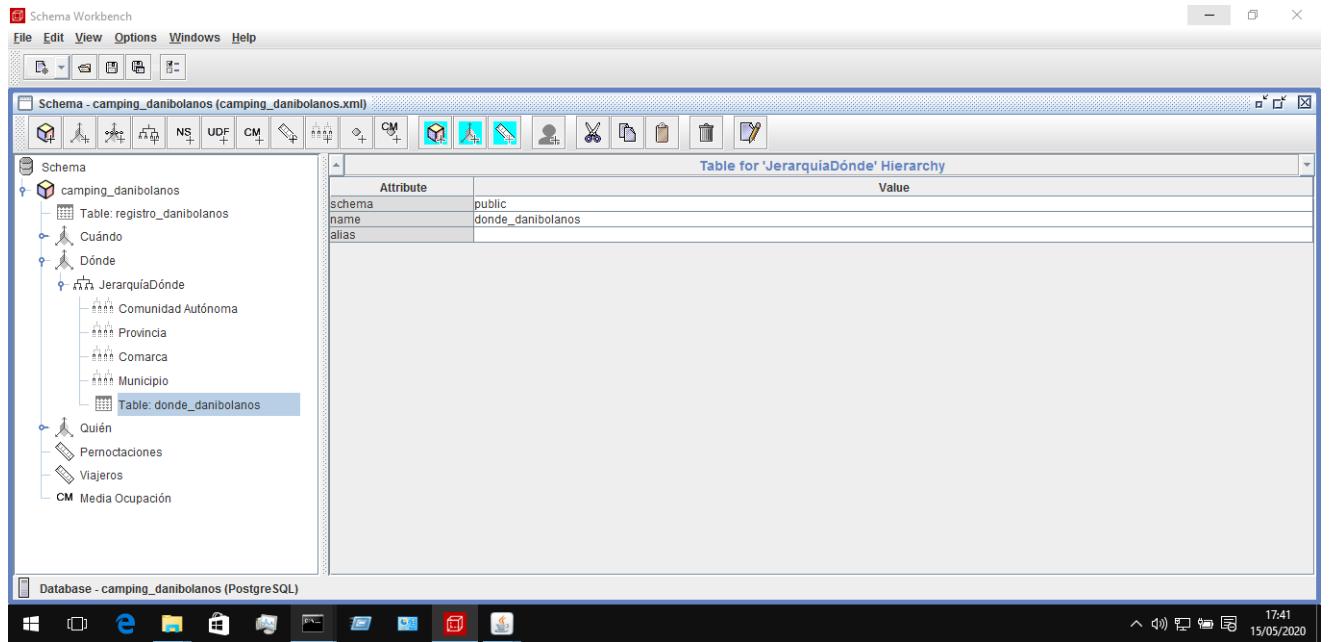


Figura 68: Definición de la tabla **donde_danibolanos** para la jerarquía.

Estando situados en el cubo, definimos la nueva dimensión **Quién**.

Al igual que la anterior su tipo será «StandardDimension».

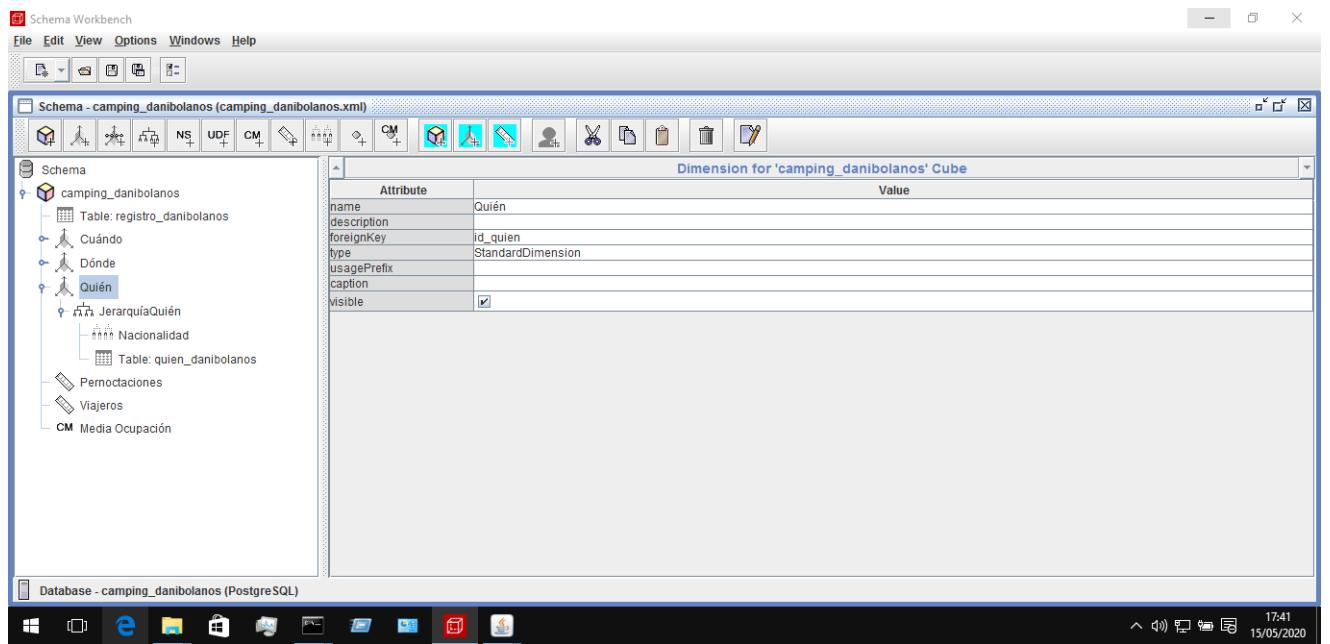


Figura 69: Definición de la dimensión **Quién**.

Para esta dimensión crearemos una jerarquía para un único campo para emular el funcionamiento de la herramienta SSDT, la cual implementa una jerarquía por campo por defecto.

La jerarquía utilizada para esta dimensión será:

- **JerarquíaQuién: Nacionalidad.**

Schema Workbench

File Edit View Options Windows Help

Schema - camping_danibolanos (camping_danibolanos.xml)

Hierarchy for 'Quién' Dimension

Attribute	Value
name	JerarquíaQuién
description	
hasAll	<input checked="" type="checkbox"/>
allMemberName	Todo Quién JearquiaQuién
allMemberCaption	Todo
allLevelName	Todo
defaultMember	
memberReaderClass	
primaryKeyTable	
primaryKey	id_quien
caption	
visible	<input checked="" type="checkbox"/>

Database - camping_danibolanos (PostgreSQL)

Schema Workbench

File Edit View Options Windows Help

Schema - camping_danibolanos (camping_danibolanos.xml)

Level for 'JerarquíaQuién' Hierarchy

Attribute	Value
name	Nacionalidad
description	
table	nacionalidad
column	
nameColumn	
parentColumn	
nullParentValue	
ordinalColumn	
type	String
internalType	
uniqueMembers	<input checked="" type="checkbox"/>
levelType	Regular
hideMemberIf	Never
approxRowCount	
caption	
captionColumn	
formatter	
visible	<input checked="" type="checkbox"/>

Database - camping_danibolanos (PostgreSQL)

Figura 70: Definición de la **JerarquíaQuién**.

Podemos definir mediciones mediante las operaciones «Add Measure» para el cubo.

Definiremos las mediciones **Viajeros** y **Pernoctaciones** y realizaremos alguna medición calculada a partir de ellas.

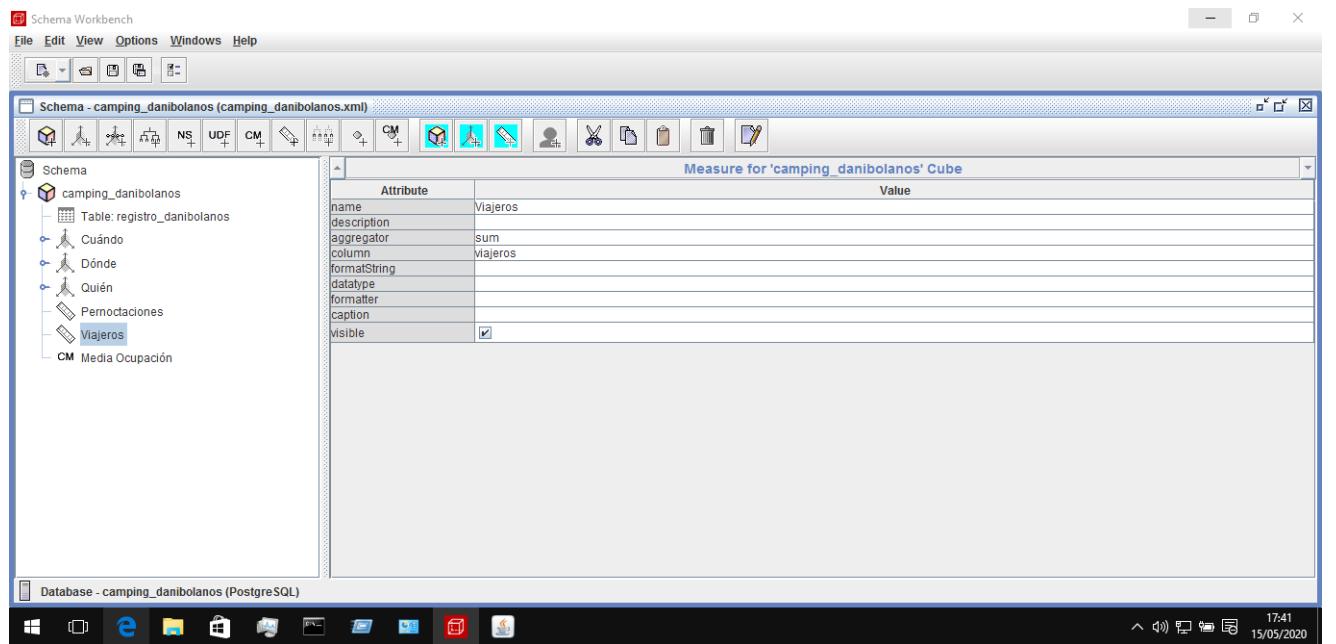


Figura 71: Definición de las mediciones.

* Mostraré únicamente la definición para **Viajeros** ya que la captura es similar para **Pernoctaciones**.

Podemos definir mediciones calculadas mediante las operaciones «Add Calculated Member» para el cubo.

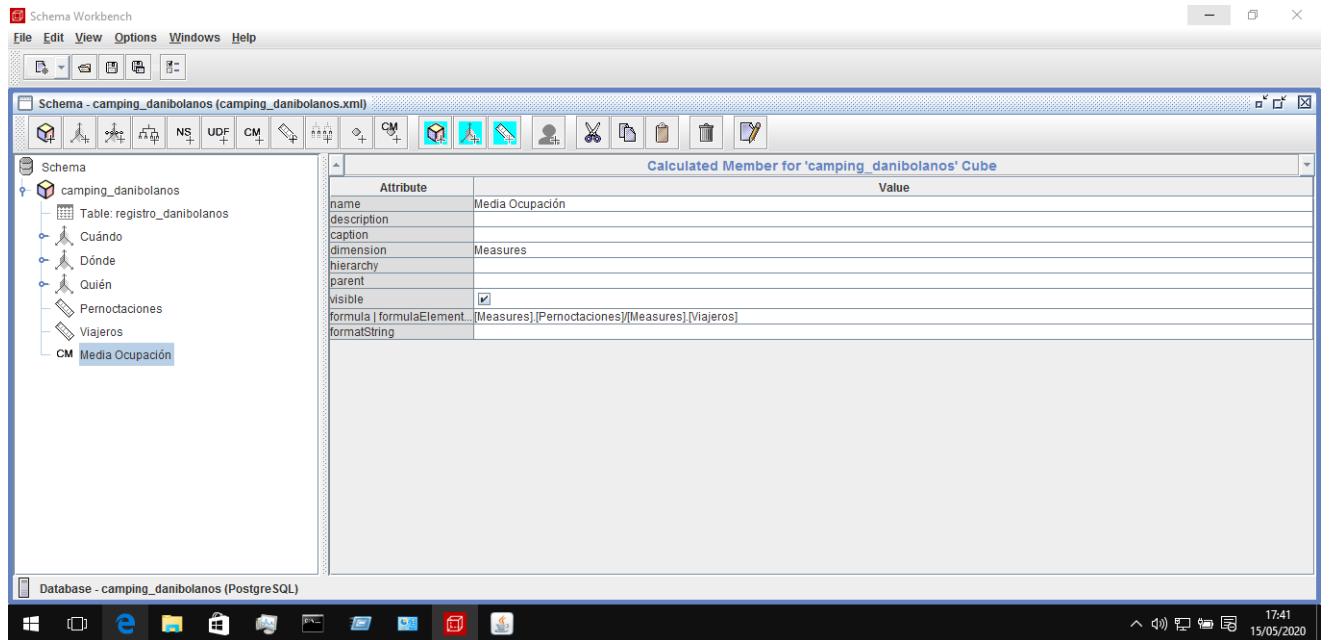


Figura 72: Definición de las mediciones calculadas.

Definiré la medición calculada **Media Ocupación** a partir de la división de las pernoctaciones entre los viajeros, de esta forma podremos estudiar la estancia media en días de los viajeros en un camping concreto en una época determinada.

Generamos el archivo XML y lo guardamos en la carpeta de catálogos en la ruta relativa `_java/jrubik/data/resources/catalogs` con el nombre de **camping_danibolanos.xml**.

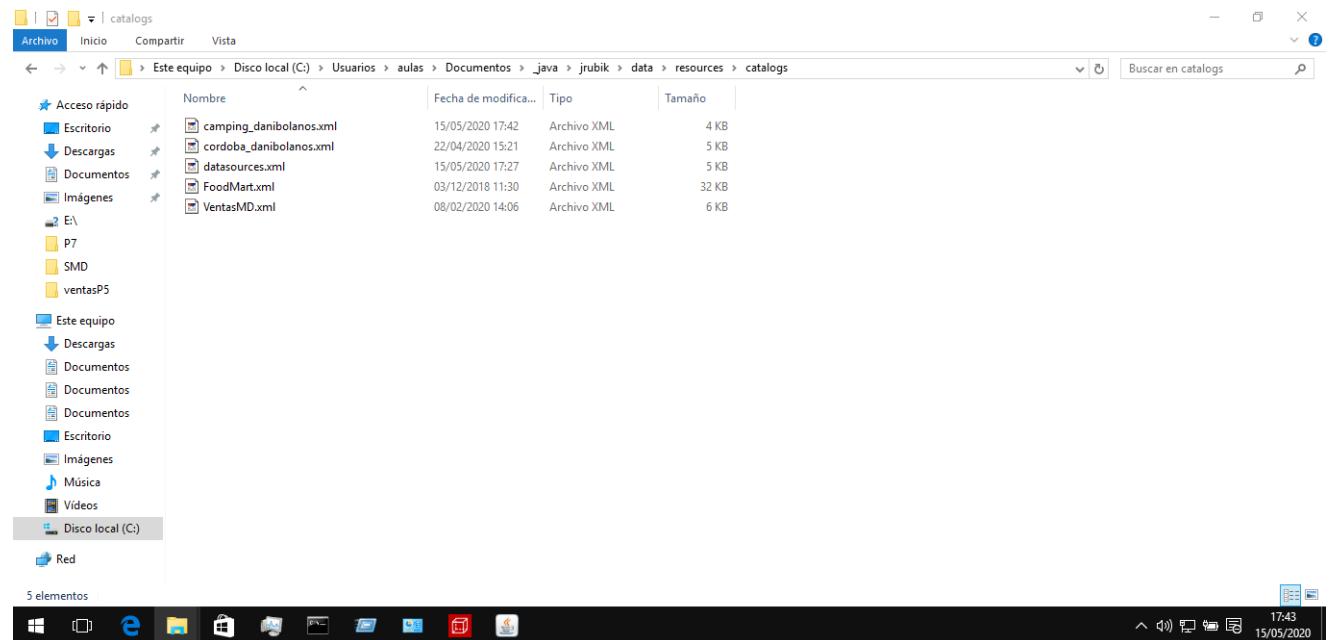


Figura 73: Guardamos el archivo XML generado.

Una vez iniciado *JRubik*, definimos la fuente de datos a partir de la BD de *PostgreSQL* y el archivo XML que hemos creado antes.

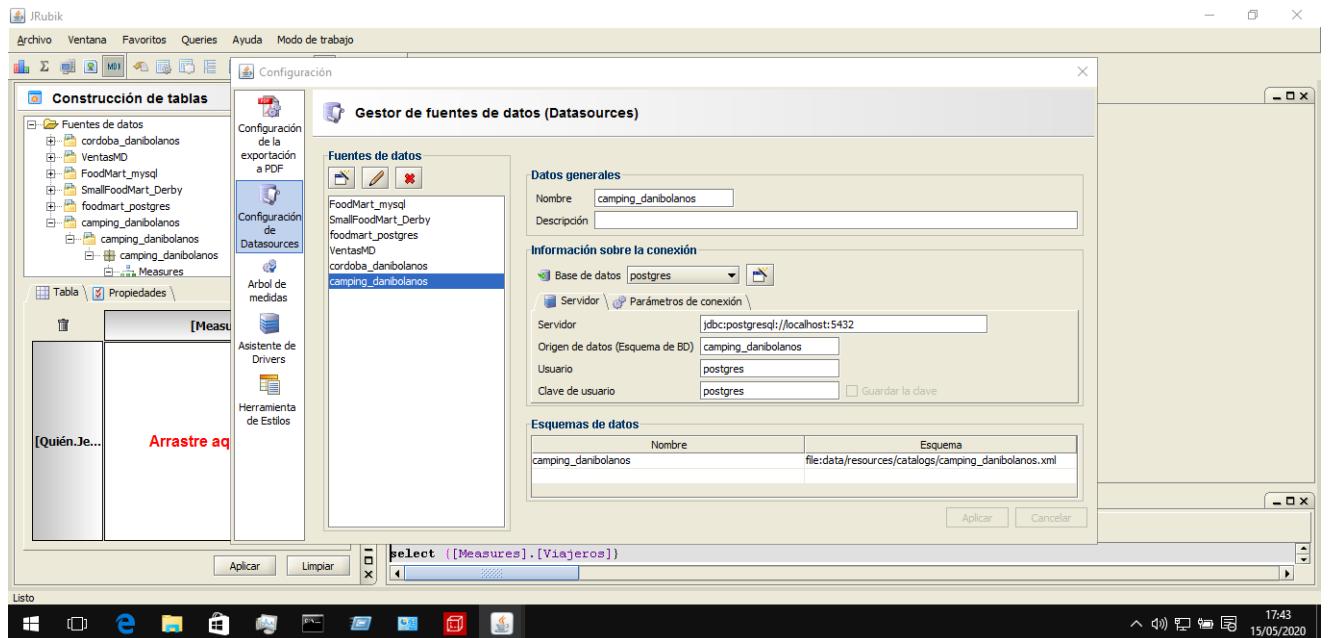


Figura 74: Definición de la fuente de datos.

En la ventana 'Navegador' podemos refinar la construcción de la tabla. Permitiendo seleccionar elementos adicionales o bien elementos distintos de los componentes incluidos en las filas, columnas y filtro.

Podemos ver las jerarquías y medidas definidas en el archivo XML.

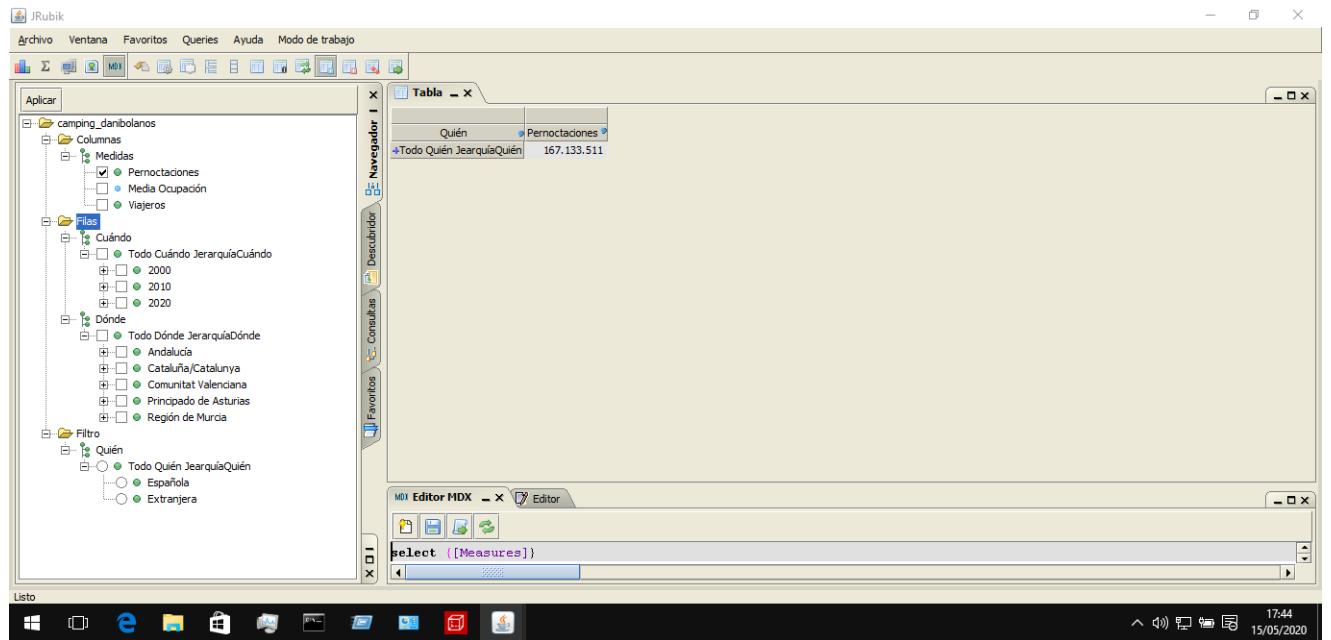


Figura 75: Ventana 'Navegador' en *JRubik*.

7. Obtén un informe inicial libre y explica su contenido.

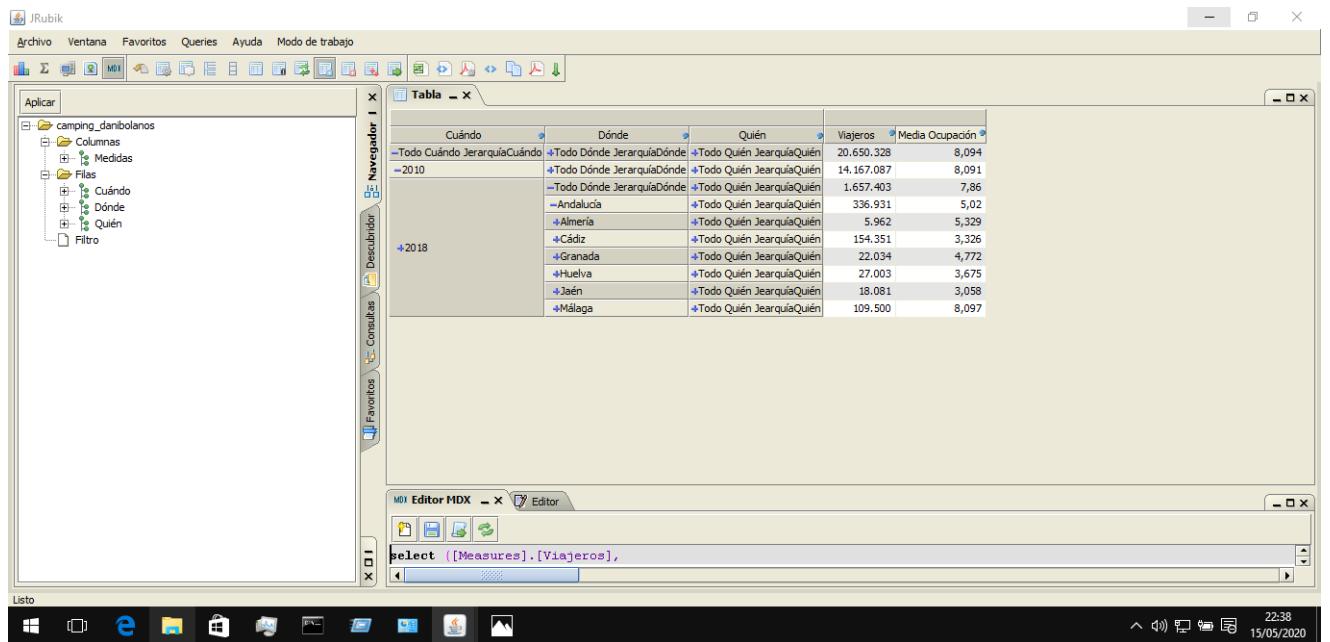


Figura 76: Informe inicial libre.

El valor de las mediciones que estamos utilizando son **Viajeros** y **Media Ocupación**.

Contenido del informe: Número de viajeros y ocupación media en días de los campings de las provincias de Andalucía en el año 2018.

Para cada dimensión nos encontramos en los siguientes niveles:

- Nivel **Año** en la dimensión **Cuándo** → Restringido al año 2018.
- Nivel **Provincia** en la dimensión **Dónde** → Restringido a las provincias de Andalucía registradas (Sevilla y Córdoba no cuentan con camping en el archivo original).
- Nivel **Todo** en la dimensión **Quién**.

7.1. Mediante Slice&Dice obtén un nuevo informe y explica su contenido.

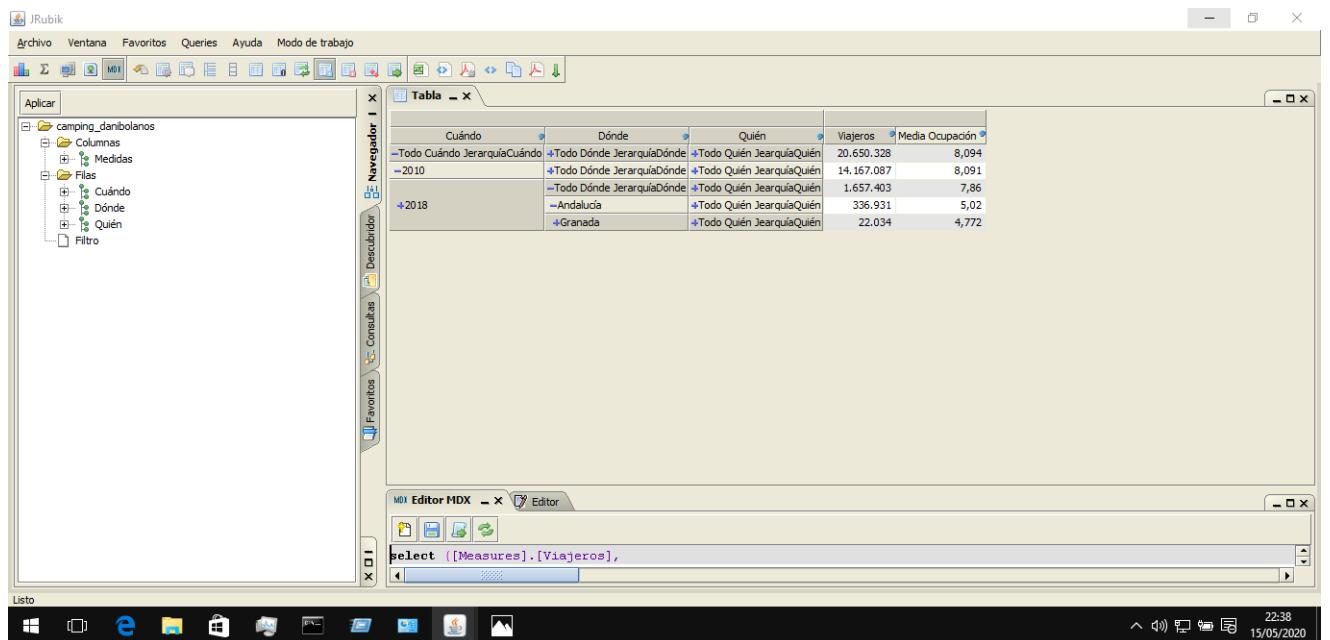


Figura 77: Operación de Slice&Dice.

Realizamos una operación de **Slice&Dice** para restringir el campo **Provincia** a **Granada**.

Contenido del informe: Número de viajeros y ocupación media en días de los campings de la provincia de Granada en el año 2018.

Para cada dimensión nos encontramos en los siguientes niveles:

- Nivel **Año** en la dimensión **Cuándo** → Restringido al año 2018.
- Nivel **Provincia** en la dimensión **Dónde** → Restringido a la provincia de Granada.
- Nivel **Todo** en la dimensión **Quién**.

7.2. Mediante Drill-down obtén un nuevo informe y explica su contenido.

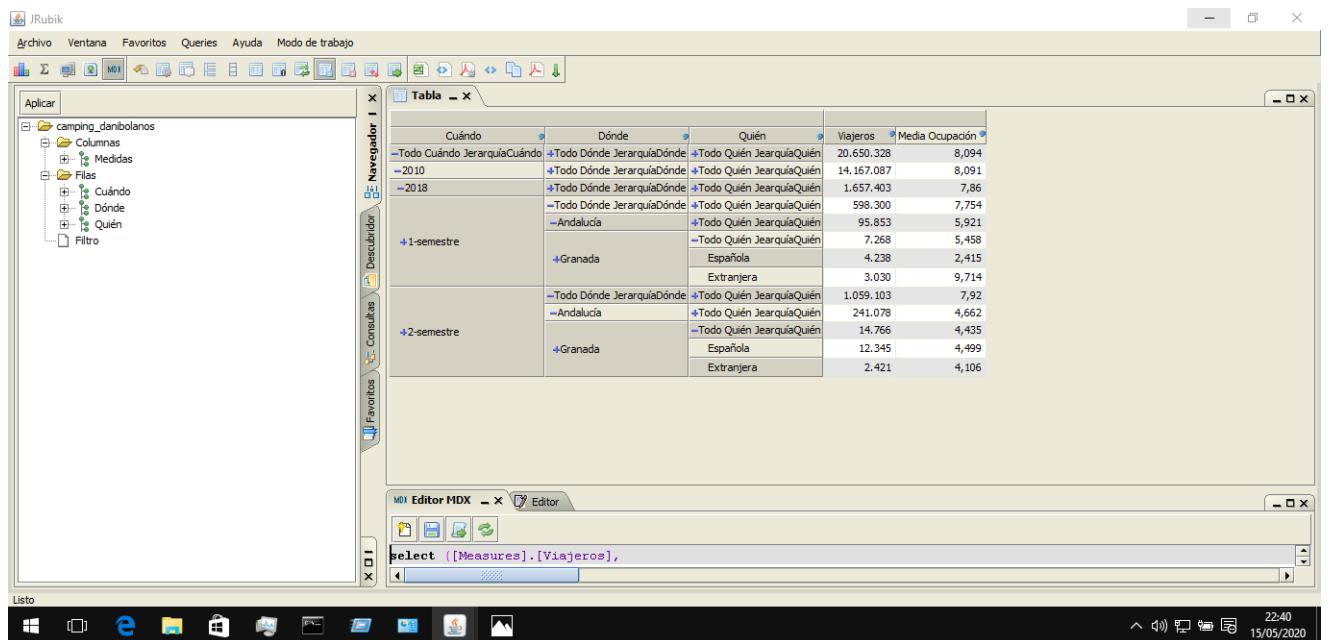


Figura 78: Operación de Drill-down.

Realizamos una operación de **Drill-down** por la dimensión **Cuándo** para pasar desde el nivel **Año** a **Semestre** y otra por la dimensión **Quién** para pasar desde el nivel **Todo** a **Nacionalidad**.

Contenido del informe: Número de viajeros y ocupación media en días por nacionalidad de los campings de la provincia de Granada para los semestres de 2018.

Para cada dimensión nos encontramos en los siguientes niveles:

- Nivel **Semestre** en la dimensión **Cuándo** → Restringido a los semestres de 2018.
- Nivel **Provincia** en la dimensión **Dónde** → Restringido a la provincia de Granada.
- Nivel **Nacionalidad** en la dimensión **Quién**.

7.3. Mediante Roll-up obtén un nuevo informe y explica su contenido.

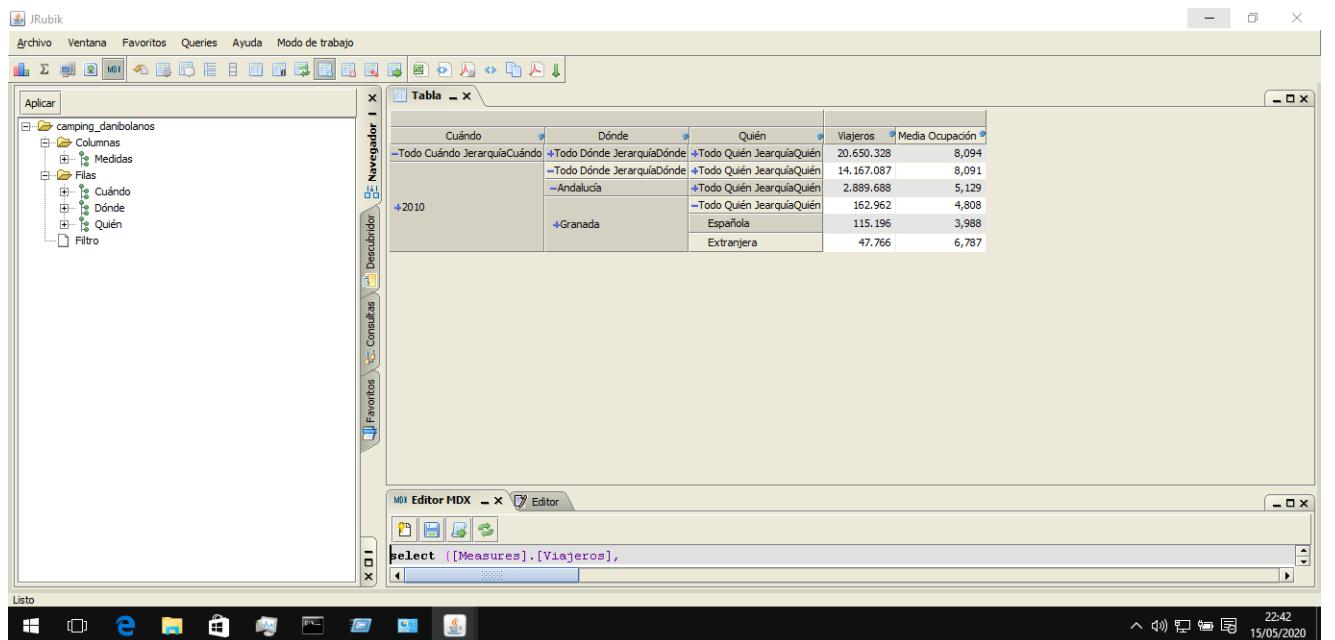


Figura 79: Operación de Roll-up.

Realizamos una operación de **Roll-up** por la dimensión **Cuándo** para pasar desde el nivel **Semestre** a **Década**.

Contenido del informe: Número de viajeros y ocupación media en días por nacionalidad de los campings de la provincia de Granada para la década de 2010.

Para cada dimensión nos encontramos en los siguientes niveles:

- Nivel **Década** en la dimensión **Cuándo** → Restringido a la década de 2010.
- Nivel **Provincia** en la dimensión **Dónde** → Restringido a la provincia de Granada.
- Nivel **Nacionalidad** en la dimensión **Quién**.

8. Bibliografía.

Referencias

- [1] Guiones de prácticas de la asignatura.
- [2] Web extracción de los datos: <https://datos.gob.es/es/catalogo/ea0010587-viajeros-y-pernoctaciones-por-puntos-turisticos-mensual-municipios-encuesta-de-ocupacion-en-campings-identificador-api-2084>