Integración de datos en UCI Sanitaria (ETL)

Álvaro González Plaza

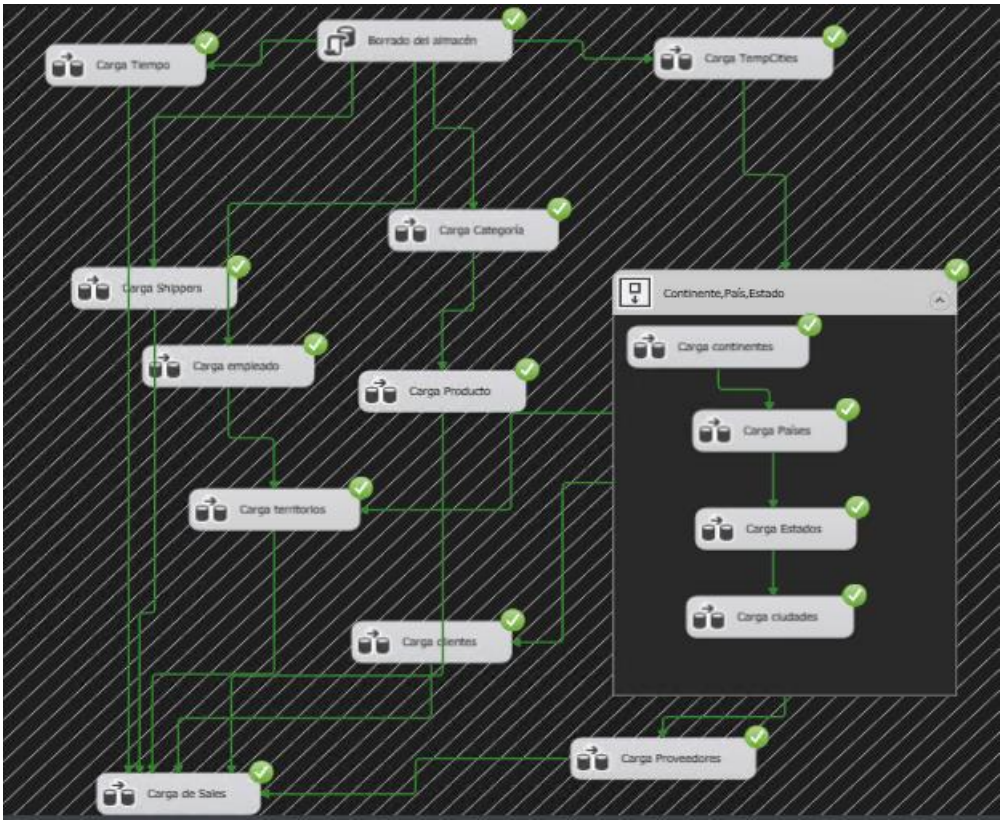
Contenido

Introducción

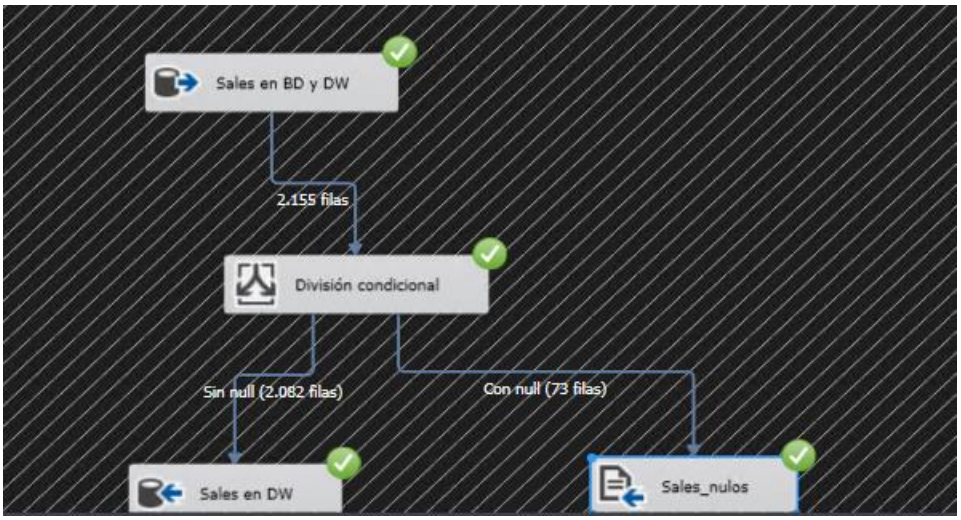
En este informe comentaremos principalmente la fase de integración de datos en nuestro almacén paso a paso, además de, comentar también de forma más breve como se llevo a cabo este proceso en la anterior base de datos de **NorthWind** y los cambios realizados en nuestro diseño desde el anterior informe tras encontrar problemas e implementar distintas soluciones.

NorthWind

Este proyecto se ha llevado a cabo de mediante el siguiente flujo de datos utilizando visual studio:



Iniciando con el borrado del almacén para que cada vez que iniciemos el proceso desde visual empecemos de 0 y no con datos escritos en ejecuciones anteriores, posteriormente cargamos tablas y por último el hecho.



Para verlo más en profundidad, podemos observar el proceso para la carga del hecho **Sales.**

Destacar que hemos utilizado un bloque de **división condicional** para distinguir en la columna **"ShippedDateKey"** entre campos nulos y no nulos, de esta manera enviamos todos aquellos sin problema a nuestro destino en el almacén y aquellos que no a un archivo de texto plano.

Observando el flujo de datos podemos confirmar que es correcto ya que, de esas 2155 filas totales, se dividen en:

- 2082 filas directas a nuestro almacén.
- 73 filas a nuestro archivo con campos nulos.

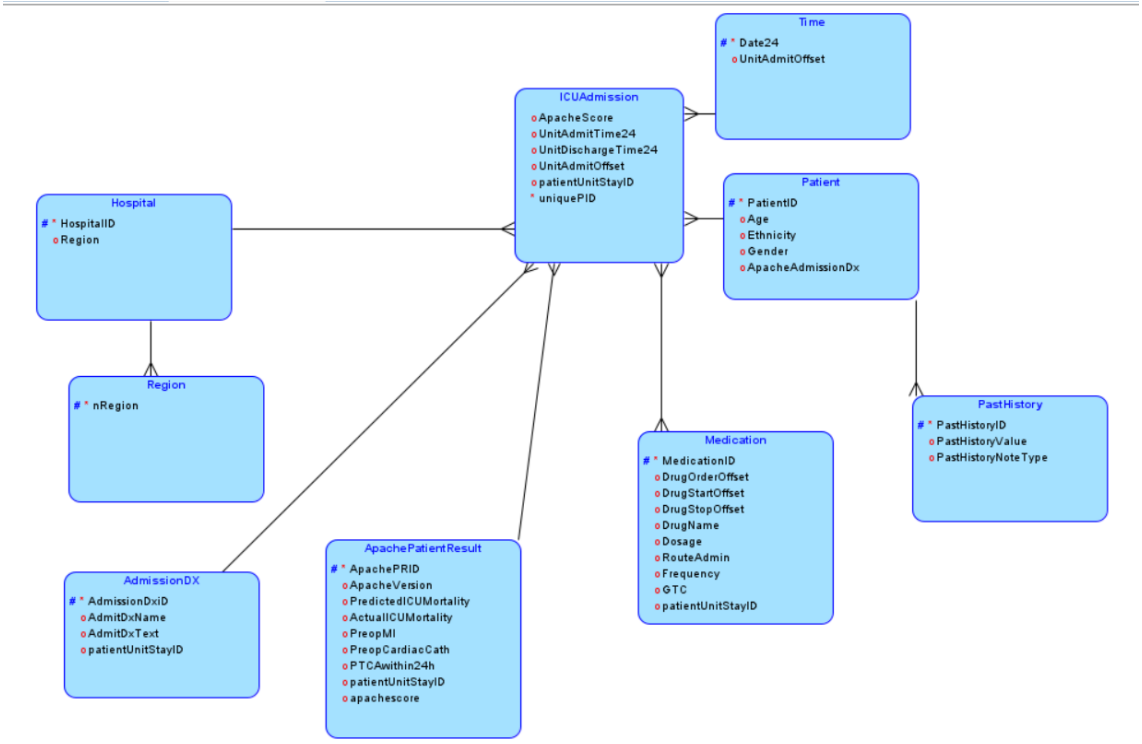
Problemas encontrados

Personalmente, los únicos obstáculos fueron al inicio mientras aprendía a usar los distintos bloques de operaciones (conversión de datos, búsqueda…) pero no he tenido ningún problema concreto a lo largo del proceso de carga de datos.

eICU Collaborative Research Database

Cambios en el diseño

Se han realizado algunos breves cambios y correcciones en algunos atributos, relaciones y dimensiones, por lo que, se adjunta a continuación el diseño lógico actual.



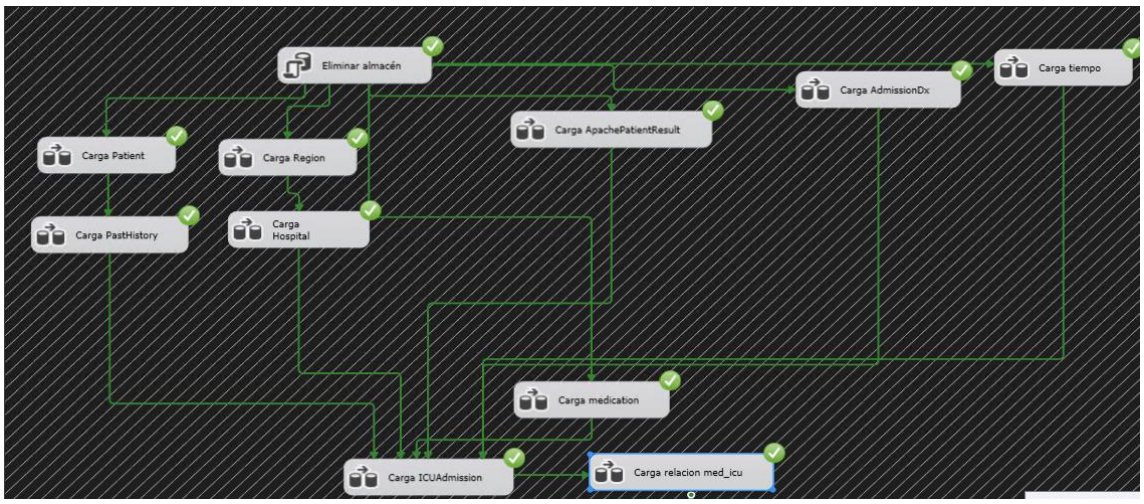
Entrando más en detalle, he modificado la tabla **time** ya que contenía anteriormente atributos que no existían en nuestra base de datos original, por lo que la he modificado usando "**date24"** como clave primaria, un atributo que surge de la concatenación de la hora de admisión y alta en urgencias de un paciente.

He corregido también la relación entre **icuadmission** y **medication** ya que esta debe ser tener una cardinalidad **M:N** (un paciente puede tener varios medicamentos y viceversa).

En cuanto a la tabla **hospital** he realizado una modificación muy similar a la de **time** mencionada previamente, y el resto de las modificaciones son breves cambios en atributos que hemos empezado a considerar o dejado de considerar importantes.

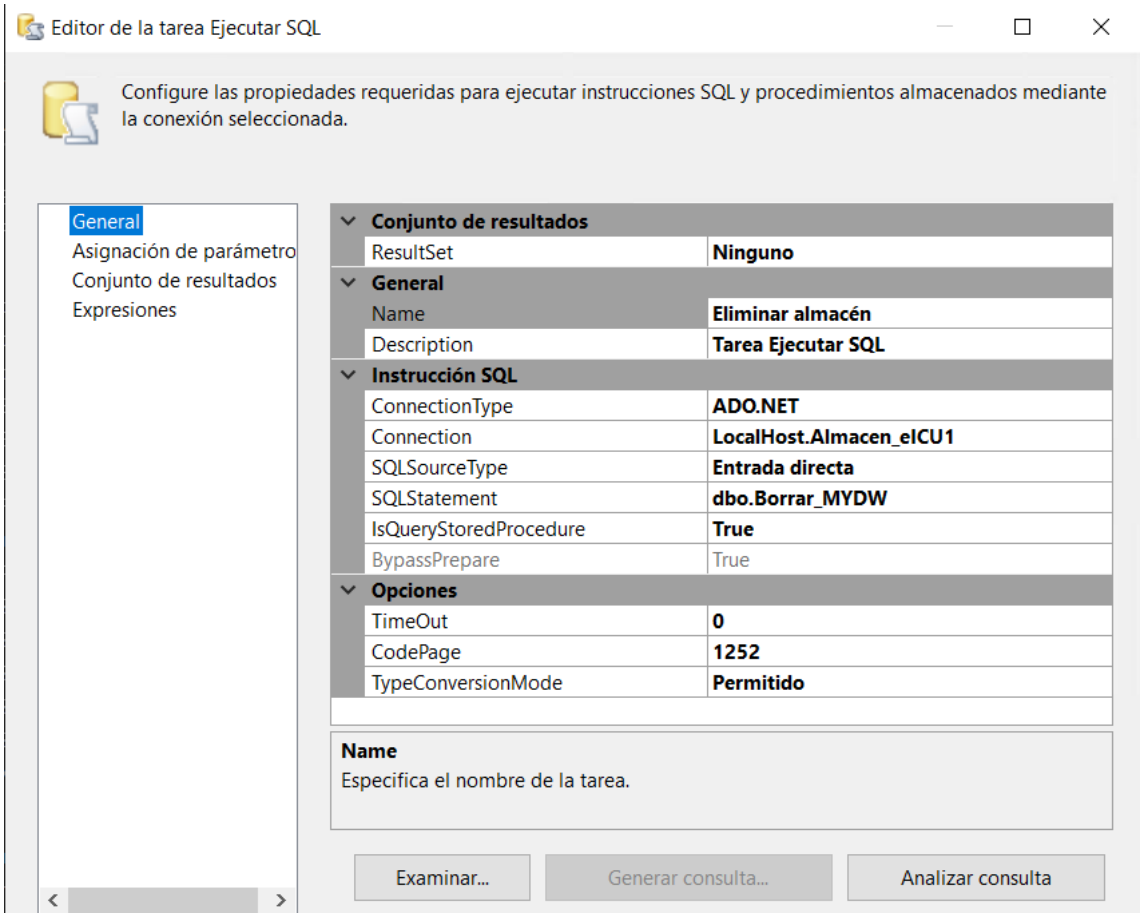
Carga en almacén

Comenzaremos a explicar todo el proceso detallando cada bloque y flujo de datos:



->Eliminar almacén

Realizamos esta tarea siempre al inicio del proceso de manera que comencemos con el almacén vacío y carguemos de nuevo los datos para comprobar que todo es correcto.



Si lo examinamos desde dentro vemos que nuestra tarea **“Eliminar almacén”** tiene activada la opción de ser un proceso almacenado y hace referencia a este, el cual es **“Borrar_MYDW”.**

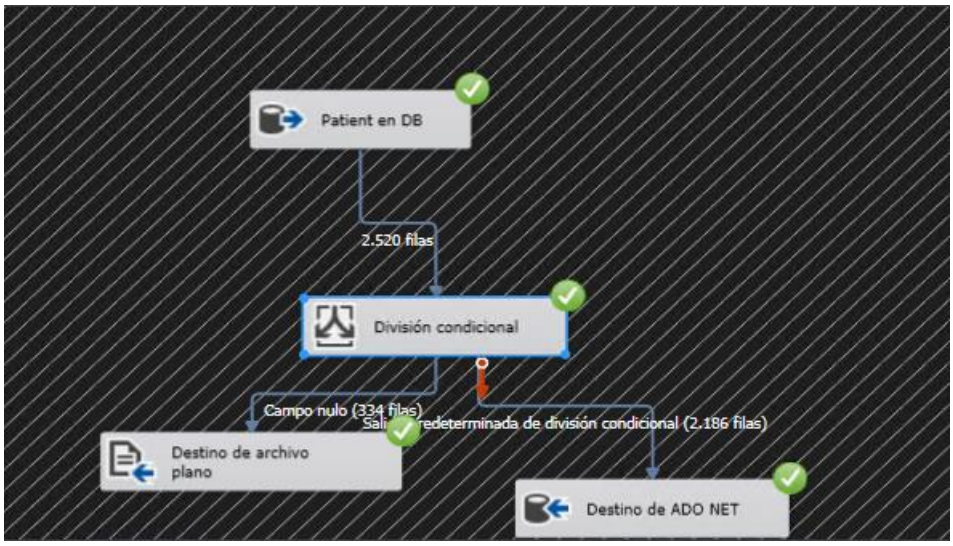
Si observamos nuestros procesos almacenados en sql server podremos ver mas en detalle que realiza concretamente este proceso:

```
USE [Almacen_eICU]
GO
/****** Object:  StoredProcedure [dbo].[Borrar_MYDW]    Script Date: 21/11/2024 2:01:46 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[Borrar_MYDW]

AS
BEGIN
delete dbo.relation_13
    delete dbo.icuadmission
    delete dbo.region
    delete dbo.hospital
    delete dbo.pasthistory
    delete dbo.patient
    delete dbo.admissiondx
    delete dbo.apachepatientresult
    delete dbo.time
    delete dbo.medication
END
```

Vemos como elimina cada tabla del almacén teniendo en cuenta que debemos eliminar primero aquellas que tienen dependencias de otras tablas, es por ello que por ejemplo, primero borramos **region** y luego **hospital**.

->Carga Patient

En esta tarea procedemos a hacer la carga de la tabla **patient** en nuestro almacén:



Por lo que podemos observar es un flujo correcto, sin problemas, pero destaco la necesidad de usar un bloque de división condicional ya que observamos en la base de datos original que en columnas como **Age**, **Ethnicity**, **Gender o ApacheAdmissionDx** contaban con valores nulos.

Gender	Age	Ethnicity	HospitalID	WardID	ApacheAdmissionDx
Male	42	NULL	123	175	Bleeding, upper GI
Female	72	Caucasian	122	212	Angina, unstable (angina interferes w/quality of li
NULL	NULL	NULL	123	175	NULL
Female	67	Caucasian	120	248	Chest pain, unknown origin
Male	50	Caucasian	110	251	Bleeding, GI-location unknown
Female	53	Caucasian	122	236	GI medical, other
Female	73	Caucasian	110	185	Pneumonia, bacterial
Female	20	Caucasian	123	175	Acid-base/electrolyte disturbance
Male	49	Caucasian	122	181	Cardiomyopathy
Male	49	Caucasian	122	181	Cardiomyopathy
Male	49	Caucasian	122	181	Cardiomyopathy
Male	21	Caucasian	120	248	Appendectomy
Female	58	African ...	131	227	Cancer, esophageal

Una vez filtrado distribuimos al flujo a dos salidas, una para aquellas sin campos nulos hacia la tabla en el almacén y otra para aquella con campos nulos a un fichero de texto plano.

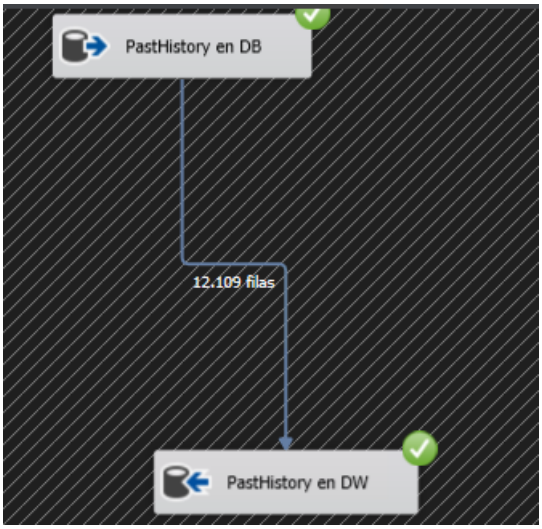
De esta manera observamos como el flujo inicial de 2520 filas se reparte en:

- 2186 filas hacia nuestro almacén.
- 334 filas hacia el fichero con campos nulos.

	patientid	age	ethnicity	gender	apacheadmissiondx
1	141765	87	Caucasian	Female	Rhythm disturbance (atrial, supraventricular)
2	143870	76	Caucasian	Male	Endarterectomy, carotid
3	144815	34	Caucasian	Female	Overdose, other toxin, poison or drug
4	145427	61	Caucasian	Male	GI perforation/rupture, surgery for
5	147307	55	Caucasian	Female	Endarterectomy, carotid
6	147784	60	Hispanic	Female	Coma/change in level of consciousness (for hepa...
7	148611	28	Caucasian	Male	Overdose, other toxin, poison or drug
8	149713	> 89	Caucasian	Female	Infarction, acute myocardial (MI)
9	151179	59	Caucasian	Female	Sepsis, cutaneous/soft tissue
10	151867	44	Caucasian	Male	GI perforation/rupture, surgery for
11	151900	66	Caucasian	Female	Sepsis, pulmonary
12	152954	41	Caucasian	Female	Respiratory - medical, other
13	153972	63	Caucasian	Male	Bleeding, lower GI
14	155961	57	Caucasian	Female	Knee replacement, total (non-traumatic)
15	156308	87	Caucasian	Male	Sepsis, pulmonary
16	156906	52	Caucasian	Female	Emphysema/bronchitis
17	157016	23	Caucasian	Female	GI medical, other
18	157427	73	Caucasian	Male	Rhythm disturbance (atrial, supraventricular)

->Carga PastHistory

En esta tarea procedemos a hacer la carga de la tabla **PastHistory** en nuestro almacén:



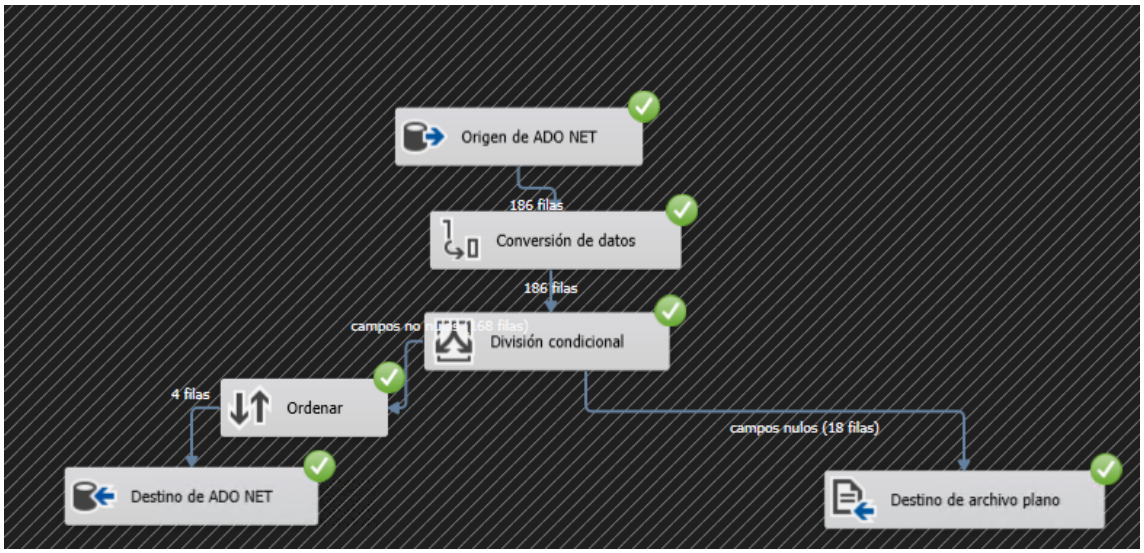
En este caso podemos hacer la carga de origen a destino de **manera directa**, ya que al realizar el diseño me asegure de que coincidieran el tamaño de los atributos para, de esta manera evitar en un futuro tener que estar usando bloques de conversión de datos y considerando que teniendo que emplear si o si ese tiempo más valía hacerlo cuanto antes.

También destacar que, una vez realizada la tarea, los datos se cargan **correctamente sin valores nulos** en sus columnas.

	pasthistoryid	pasthistoryvalue	pasthistorynotetype	patient_patientid
1	848035	Performed	Comprehensive Progress	157427
2	848036	insulin dependent diabetes	Comprehensive Progress	157427
3	848467	Performed	Comprehensive Progress	238463
4	848468	CHF	Comprehensive Progress	238463
5	851377	No Health Problems	Comprehensive Progress	174956
6	851418	No Health Problems	Comprehensive Progress	232447
7	852162	Performed	Comprehensive Progress	162842
8	852163	COPD - no limitations	Comprehensive Progress	162842
9	852164	hypertension requiring treatment	Comprehensive Progress	162842
10	852165	renal insufficiency - creatinine 1-2	Comprehensive Progress	162842
11	852166	atrial fibrillation - intermittent	Comprehensive Progress	162842
12	857319	Performed	Comprehensive Progress	205061
13	857320	MI - within 2 years	Comprehensive Progress	205061
14	857321	hypertension requiring treatment	Comprehensive Progress	205061
15	857322	peripheral vascular disease	Comprehensive Progress	205061

->Carga Region

En esta tarea procedemos a hacer la carga de la tabla **Region** en nuestro almacén:



Comencé realizando una **conversión de datos** ya que el atributo **Region** no coincidía en tamaño entre DB y DW.

Una vez hecho, observé que **Region** también contaba con campos nulos por lo que necesitaba utilizar un bloque de **división condicional**:

Region
South
South
South
North...
North...
NULL
North...
NULL
North...
North...
North...
South
South

Utilizando un bloque de **ordenar** y activando la opción de eliminar filas con valores repetidos se soluciona rápidamente también el problema de encontrar valores repetidos y cargamos en nuestro almacén finalmente.

Paralelamente, en el caso de la **división condicional** para los casos de campos nulos en **Region,** estos se dirigen hacia un archivo de texto plano.

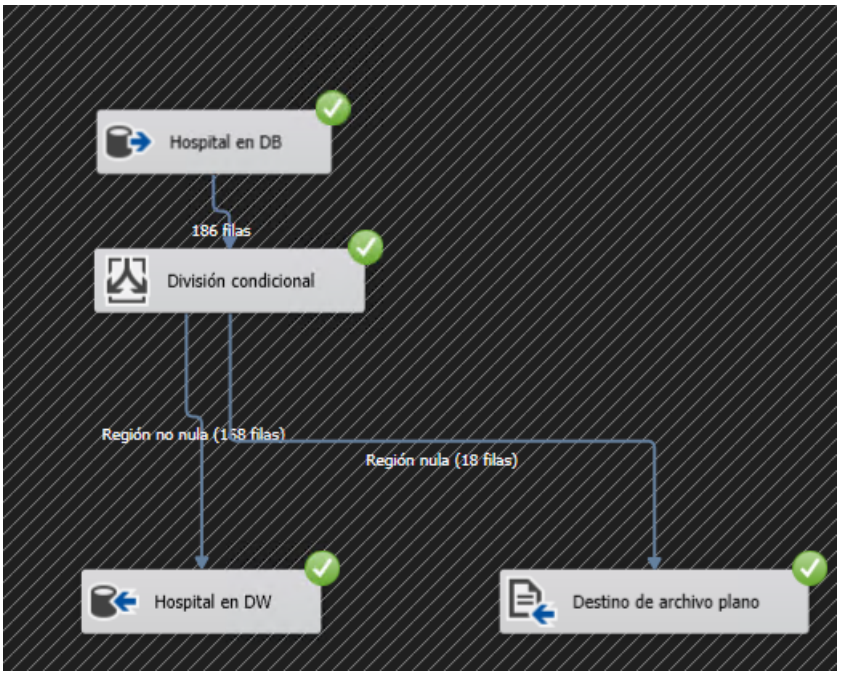
De esta manera observamos como el flujo inicial de 186 filas se reparte en:

- 4 filas hacia nuestro almacén de campos no nulos ni repetidos.
- 18 filas hacia el fichero con campos nulos.

	nregion	hospital_hospitalid
1	Midwest	138
2	Northeast	425
3	South	140
4	West	400

->Carga Hospital

En esta tarea procedemos a hacer la carga de la tabla **Hospital** en nuestro almacén:



Mirando la DB vemos que cuenta con varios campos null en **Region**, por lo que utilice un bloque de **división condicional** para dividirlo en dos flujos.

En el caso de campos **no nulos** se dirige directamente a nuestro almacén y en el caso de existencia de campos **nulos** se dirige a un archivo de texto plano.

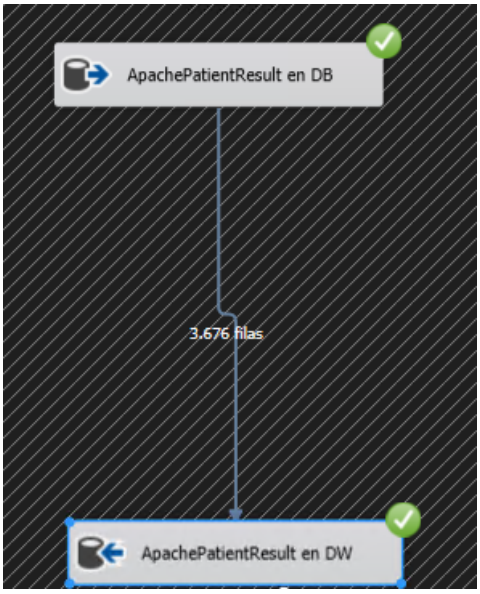
De esta manera observamos como el flujo inicial de 186 filas se reparte en:

- 168 filas hacia nuestro almacén de campos no nulos.
- 18 filas hacia el fichero con campos nulos.

	hospitalid	region
1	56	Midwest
2	58	Midwest
3	59	Midwest
4	60	Midwest
5	61	Midwest
6	63	Midwest
7	66	Midwest
8	67	Midwest
9	68	Midwest
10	69	Midwest
11	71	Midwest
12	73	Midwest

->Carga ApachePatientResult

En esta tarea procedemos a hacer la carga de la tabla **ApachePatientResult** en nuestro almacén:



En este caso no había incompatibilidad entre tipo de datos, valores nulos ni repetidos por lo que pude **cargar directamente** los datos en el almacén, ya que al realizar el diseño me asegure de que coincidieran el tamaño de los atributos para, de esta manera evitar en un futuro tener que estar usando bloques de conversión de datos.

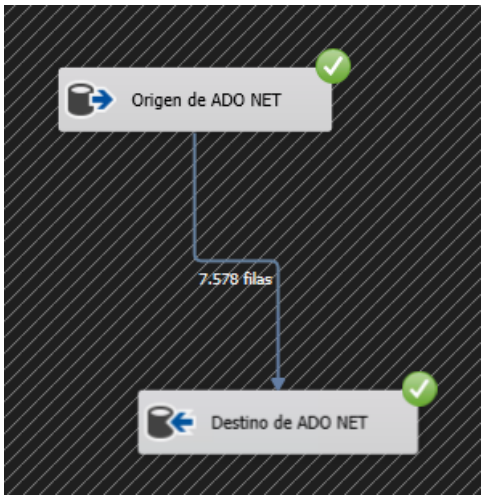
También destacar que, una vez realizada la tarea, los datos se cargan **correctamente sin valores nulos** en sus columnas.

De esta manera observamos un flujo de 3676 filas.

	apacheprid	apacheversion	predictedicumortality	actualicumortality	preopmi	preopcardiaccath	ptcawithin24h	patientunitstayid	apachescore
1	181	IV	1.8341987534681509E-3	ALIVE	0	0	0	144815	25
2	182	IVa	2.1330459623421791E-3	ALIVE	0	0	0	144815	25
3	473	IV	6.1409010299298142E-2	ALIVE	0	0	0	149713	57
4	474	IVa	3.2460274973360508E-2	ALIVE	0	0	0	149713	57
5	6058	IV	4.5817101950284354E-3	ALIVE	0	0	0	155961	31
6	6059	IVa	5.6437687730189808E-3	ALIVE	0	0	0	155961	31
7	8983	IV	9.0838207769845698E-3	ALIVE	0	0	0	211715	27
8	8984	IVa	4.659008763267484E-3	ALIVE	0	0	0	211715	27
9	9148	IVa	0.02796019241724889	ALIVE	0	0	0	214497	58
10	9149	IV	3.3463287912994272E-2	ALIVE	0	0	0	214497	58
11	10432	IV	0.77641510530608837	EXPIRED	0	0	0	238463	130
12	10433	IVa	0.73819216857275105	EXPIRED	0	0	0	238463	130

->Carga AdmissionDx

En esta tarea procedemos a hacer la carga de la tabla **AdmissionDx** en nuestro almacén:



De nuevo no había incompatibilidad entre tipo de datos, valores nulos ni repetidos por lo que pude **cargar directamente** los datos en el almacén, ya que al realizar el diseño me asegure de que coincidieran el tamaño de los atributos para, de esta manera evitar en un futuro tener que estar usando bloques de conversión de datos.

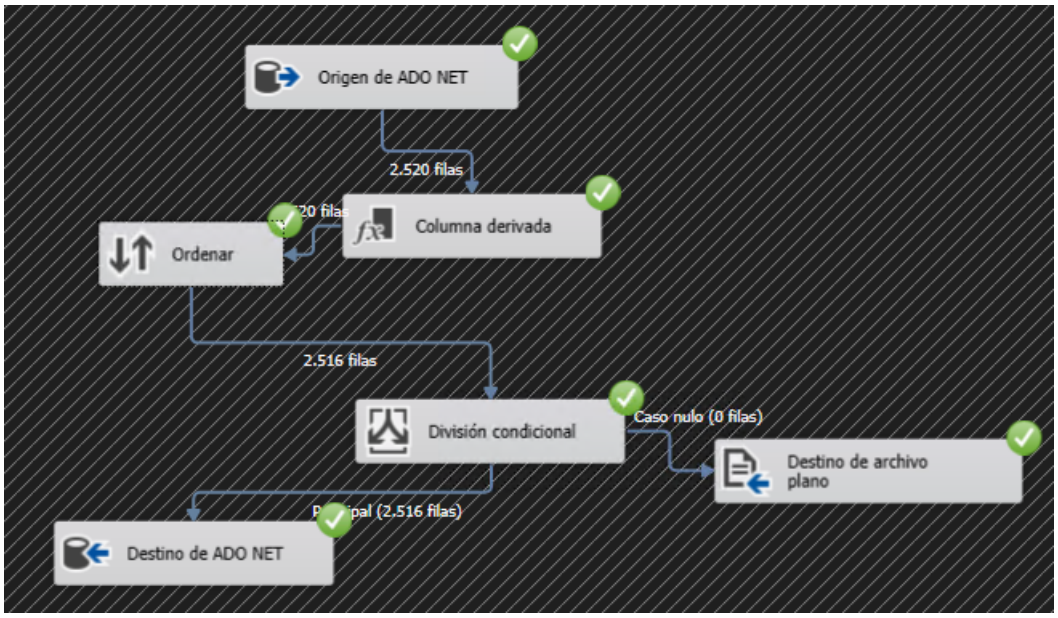
Finalmente observamos que los datos se cargan **correctamente sin valores nulos** en sus columnas.

De esta manera observamos un flujo de 7578 filas.

	admissiondxid	admitdxname	admitdxtext	patientunitstayid
1	471386	No	No	205928
2	471387	Cardiovascular	Cardiovascular	205928
3	471388	Sepsis, cutaneous/soft tissue	Sepsis, cutaneous/soft tissue	205928
4	472754	Yes	Yes	217838
5	472755	Yes	Yes	217838
6	472756	Musculoskeletal/Skin	Musculoskeletal/Skin	217838
7	472757	Hip replacement, total (non-traumatic)	Hip replacement, total (non-traumatic)	217838
8	472840	Yes	Yes	155961
9	472841	Yes	Yes	155961
10	472842	Musculoskeletal/Skin	Musculoskeletal/Skin	155961
11	472843	Knee replacement, total (non-traumatic)	Knee replacement, total (non-traumatic)	155961
12	473111	Yes	Yes	211715

->Carga Time

En esta tarea procedemos a hacer la carga de la tabla **Time** en nuestro almacén:



Esta tarea es algo mas elaborada debido a que decidí que la clave primaria **date24** fuera la concatenación de las columnas originales en la DB de **"UnitAdmit24"** y **"UnitDischarge24".**

Derived Column Name	Derived Column	Expression	Data Type
tiempo24	<add as new column>	UnitAdmitTime24 + " / " + UnitDischargeTime...	cadena Unicode [DT_...

Posteriormente, utilice un bloque **ordenar** activando la opción de eliminar las filas con valores repetidos y por último una **división condicional** para repartir mi flujo de datos entre campos **nulos** y **no nulos**.

De esta manera, los valores no nulos iban directos a nuestra tabla en el **almacén** y los nulos a un **archivo de texto plano**.

De esta manera observamos como el flujo inicial de 2520 filas pasa a ser de 2516 tras eliminar las filas con valores repetidos y luego de esto se reparte en:

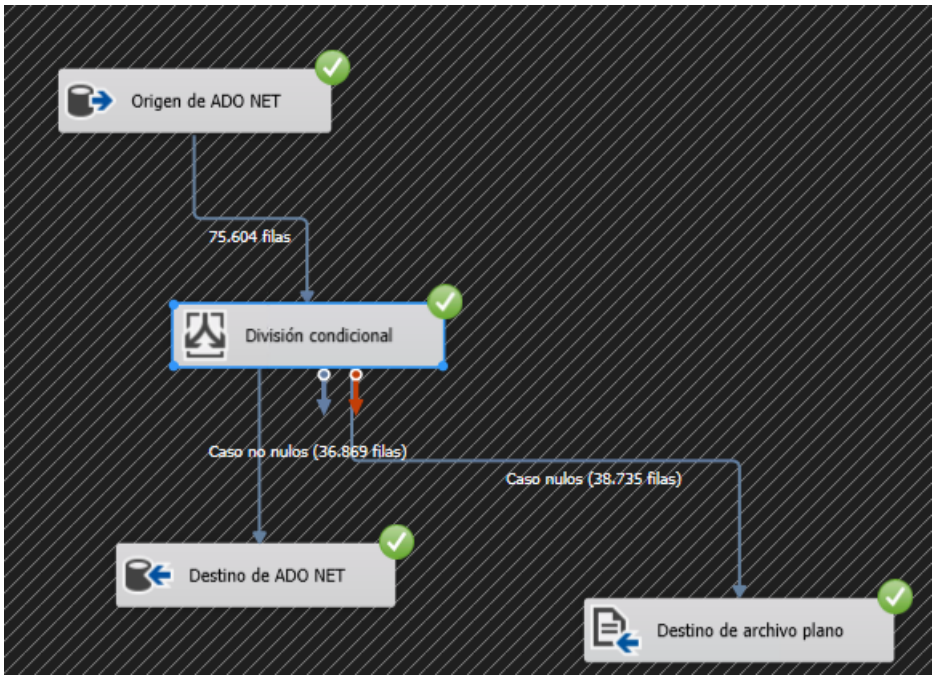
- 2516 filas hacia nuestro almacén de campos no nulos.
- 0 filas hacia el fichero con campos nulos.

En este caso, se hizo más bien de manera preventiva a pesar de dirigir 0 filas al archivo.

	date24	unitadmitoffset
1	00:00:00 / 11:34:00	3574
2	00:00:00 / 17:53:00	25553
3	00:00:00 / 18:00:00	8280
4	00:00:00 / 20:45:00	2685
5	00:01:00 / 22:06:00	2765
6	00:02:00 / 00:03:00	4321
7	00:02:00 / 20:22:00	2660
8	00:02:00 / 22:42:00	1360
9	00:03:00 / 02:20:00	1577
10	00:04:00 / 15:40:00	936
11	00:05:00 / 02:42:00	1597
12	00:05:00 / 23:07:00	5702

->Carga medication

En esta tarea procedemos a hacer la carga de la tabla **medication** en nuestro almacén:



Echando un vistazo a la DB vemos que cuenta con varios campos null en **Frequency, DrugName** y **Dosage**, por lo que utilice un bloque de **división condicional** para dividirlo en dos flujos.

En el caso de campos **no nulos** se dirige directamente a nuestro almacén y en el caso de existencia de campos **nulos** se dirige a un archivo de texto plano.

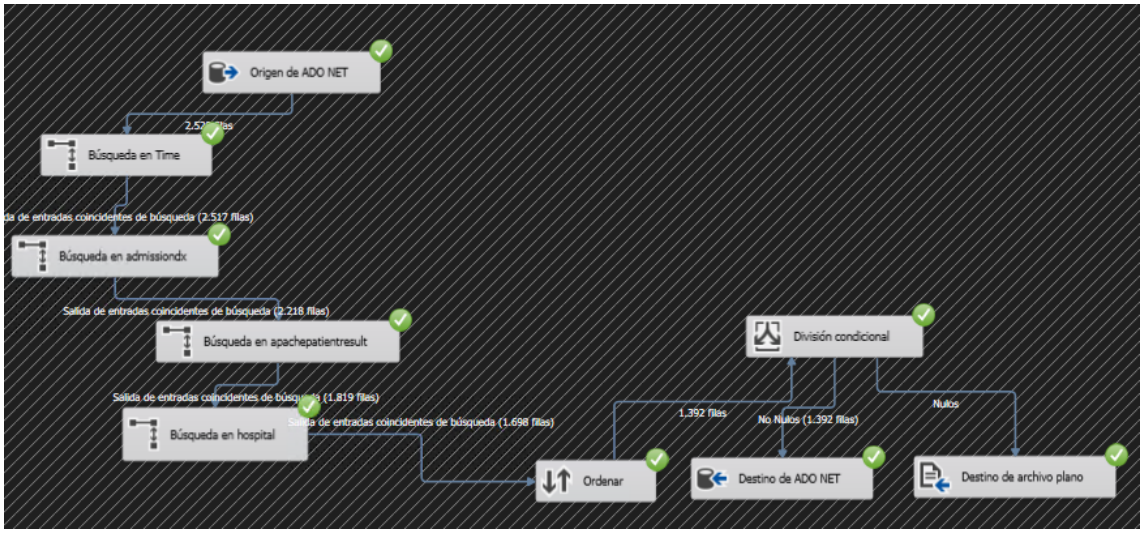
De esta manera observamos como el flujo inicial de 75604 filas se reparte en:

- 36869 filas hacia nuestro almacén de campos no nulos.
- 38735 filas hacia el fichero con campos nulos.

	medicationid	drugorderoffset	drugstartoffset	drugstopoffset	drugname	dosage	routeadmin	frequency	gtc
1	6666501	7515	8904	13225	VANCOMYCIN 1.25 GM IN NS 250 ML IVPB (REPACKAGE)	1,250 3	IV	Daily	0
2	6668057	5	-30	8304	2 ML VIAL : ONDANSETRON HCL 4 MG/2ML IJ SOLN	4 3	IV	BID PRN	65
3	6668520	12309	12266	15553	MAGNESIUM SULFATE 2 G IN NS PREMIX	2 4	IV	Daily PRN	59
4	6670295	22	401	4455	HYDROCHLOROTHIAZIDE 25 MG PO TABS	25 3	PO	Nightly	0
5	6670442	4	-3732	-44	POTASSIUM CHLORIDE 20 MEQ PO PACK	40 7	PER NG TUBE	Q4H PRN	0
6	6670560	2204	2178	8599	POTASSIUM CHLORIDE CRYS ER 20 MEQ PO TBCR	20 7	PO	Q4H PRN	0
7	6673110	2	-375	16	1000 ML - LACTATED RINGERS IV SOLN	10 mL/hr	IV	Continuous	59
8	6673263	2	-362	54	DEXTROSE 5 % IV SOLN	100 41	IV	Continuous PRN	59
9	6675638	1	-11	1206	473 ML - CHLORHEXIDINE GLUCONATE 0.12 % MT SOLN	15 1	SWISH	PRN	0
10	6675769	-432	-319	3173	LOSARTAN POTASSIUM 50 MG PO TABS	50 3	PO	Daily	0
11	6679099	5354	5348	5532	ONDANSETRON 4 MG PO TBDP	4 3	PO	On Call to Procedure	0
12	6681484	7	2	1011	ACETAMINOPHEN 650 MG RE SUPP	650 3	RE	Q4H PRN	0

->Carga ICUAdmission

En esta tarea procedemos a hacer la carga de la tabla de hecho en nuestro almacén:



Al ser la tabla de hecho, todo el flujo de datos de las cargas anteriores debe de haberse completado correctamente. Comenzamos cargando la tabla **Patient** de la DB, ya que es aquí de donde obtendremos la mayoría de los atributos de nuestro hecho.

Comenzaremos realizando una búsqueda en nuestra tabla **Time** del almacén para asociar su PK con nuestra FK en el hecho. A continuación, realice reiteradas veces el mismo procedimiento para conseguir el mismo fin en las tablas de **admissiondx, apachepatientresult y hospital.**

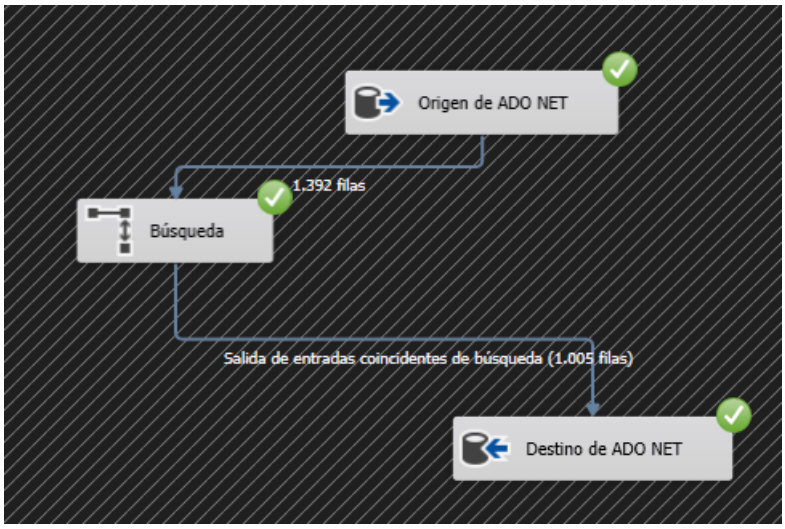
Luego, utilice un bloque **ordenar** para eliminar las filas con valores repetidos y finalmente un bloque de **división condicional**.

En este bloque distribuimos el flujo entre dos vertientes, una para los casos no nulos que irán directos a nuestro **almacén** para de esta manera completar la tabla, y otro hacia un archivo de texto plano por si encontrará algún caso de campo nulo.

	apachescore	unitadmittime24	unitdischargetime24	hospital_hospitalid	admissiondx_admissiondxid	patient_patientid	time_date24	apachepatientresult_apacheprid	unita
1	28	13:33:00	18:03:00	68	712402	147460	13:33:00 / 18:03:00	133821	Alive
2	65	09:36:00	02:39:00	56	510516	190098	09:36:00 / 02:39:00	131422	Alive
3	68	08:32:00	23:50:00	58	682125	176377	08:32:00 / 23:50:00	24488	Alive
4	36	21:25:00	01:37:00	60	666441	173545	21:25:00 / 01:37:00	135276	Alive
5	28	10:05:00	19:15:00	68	513490	163721	10:05:00 / 19:15:00	44896	Alive
6	34	20:41:00	21:30:00	56	696326	141489	02:35:00 / 03:24:00	11495	Alive
7	37	23:58:00	22:47:00	68	678827	132209	07:04:00 / 05:53:00	106474	Alive
8	29	22:36:00	23:42:00	68	650361	163508	16:54:00 / 18:00:00	87301	Alive
9	45	22:11:00	02:17:00	68	498817	160310	22:11:00 / 02:17:00	76550	Alive
10	66	23:28:00	01:35:00	60	477046	148079	23:28:00 / 01:35:00	96920	Alive
11	49	23:37:00	19:44:00	58	642028	148781	23:37:00 / 19:44:00	128703	Alive
12	18	02:20:00	15:48:00	61	670579	143004	02:20:00 / 15:48:00	148883	Alive

->Carga relation

Finalmente procedí con la carga de la relación muchos a muchos entre **medication** y **icuadmission**:



Cargamos de Origen la tabla **icuadmission** y posteriormente hacemos una **búsqueda** para obtener la **medicationid** en la tabla **medication**.

Finalmente enviamos ambos a la tabla **relation**.

De esta manera observamos un flujo de 1005 filas.

	medication_medicationid	icuadmission_uniquepid
1	6666501	002-11976
2	6668520	002-10018
3	6670295	002-10187
4	6670442	002-1115
5	6673110	002-10603
6	6673263	002-12289
7	6675638	002-1010
8	6675769	002-11623
9	6681484	002-10148
10	6682058	002-10079
11	6682337	002-11144
12	6684143	002-10063

Problemas encontrados

En general, ha sido una resolución fluida sin grandes problemas a destacar. Quizás donde mas he encontrado problema ha sido al intentar alimentar el hecho y la relación muchos a muchos:

-Alimentación de **ICUAdmission:** al llegar el momento de cargar el hecho me encontré con el problema de que no sabía la manera en la que relacionar las claves primarias de cada tabla externa con mis foráneas. Por lo que, la única solución que me ocurrió es utilizar el atributo **"PatientUnitStayID"**, en todas aquellas tablas donde fuera necesario para así poder relacionarlas y alimentar mi tabla sin problemas.

-Relación **muchos a muchos:** llegado este momento, era la primera vez que me encontraba con tener que alimentar una tabla de este tipo, por lo que tras darle varias vueltas finalmente encontré la manera cargando primero la tabla **medication,** luego **icuadmission** y finalmente la tabla de la **relación**.

Bibliografía

Como ayuda en el proceso, únicamente he consultado la página web donde se aloja la **DB** para consultar información de tablas, descripciones de atributos e información relevante de manera más rápida (Click aquí).