|  |
| --- |
| Sistemas Inteligentes II (2019/2020) |
| **Inteligencia de negocio**:  Proceso ETL construcción de un data warehouse para análisis evolución COVID-19 |
|  |
| Tender Vectores, Ilustraciones Y Gráficos - 123RF |
|  |
|  |

Alberto Ramos Sánchez [alberto.ramos104@alu.ulpgc.es](file:///E:\Apuntes\SI2\turism-ontology\docs\alberto.ramos104@alu.ulpgc.es)

Benearo Semidan Páez [benearo.paez101@alu.ulpgc.es](mailto:benearo.paez101@alu.ulpgc.es)

Índice

[Sistemas Inteligentes II (2019/2020) i](#_Toc43316103)

[Introducción 1](#_Toc43316104)

[Desarrollo del datawarehouse 1](#_Toc43316105)

[Tabla de hechos 3](#_Toc43316106)

[Tablas de dimensión temporales 6](#_Toc43316107)

[Tabla de dimensión de región 8](#_Toc43316108)

[Generación de informes 10](#_Toc43316109)

# Introducción

En esta memoria se detalla el diseño e implementación de estructura y contenido de un *datawarehouse* para el análisis de la incidencia del COVID-19 a nivel mundial. Para ello, hacemos uso de Pentaho Data Integration.

De manera adicional, mostraremos el uso del *datawarehouse* previo para realizar informes dinámicos con la herramienta Pentaho Report Designer.

# Desarrollo del datawarehouse

El trabajo principal de los procesos ETL realizados son de la manera que se muestra en la ilustración 1, justo debajo.

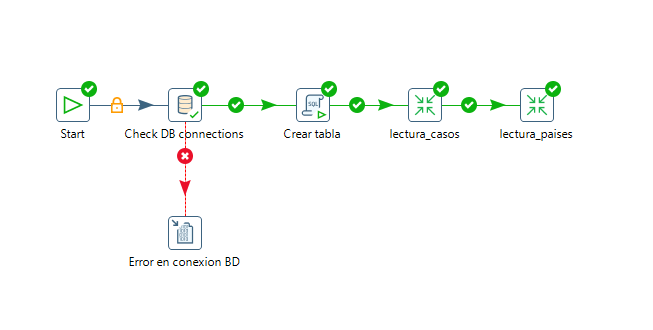


Ilustración - Trabajo principal

Al principio, realizamos la conexión con la base de datos SQLite3, en la cual creamos las tablas necesarias para seguir la estructura especificada para este trabajo. Las tablas se crean mediante las siguientes sentencias SQL:

DELETE FROM Quarter\_Time;

DELETE FROM Week\_Time;

DELETE FROM Time;

DELETE FROM Region;

DELETE FROM Cases;

DROP TABLE IF EXISTS Quarter\_Time;

DROP TABLE IF EXISTS Week\_Time;

DROP TABLE IF EXISTS Time;

DROP TABLE IF EXISTS Region;

DROP TABLE IF EXISTS Cases;

CREATE TABLE Quarter\_Time(

"date" DATE NOT NULL,

"year" int NOT NULL,

"quarter" int NOT NULL,

"month" int NOT NULL,

"day" int NOT NULL,

PRIMARY KEY("date")

);

CREATE TABLE Week\_Time(

"date" DATE NOT NULL,

"year" int NOT NULL,

"week" int NOT NULL,

"day\_week" int NOT NULL,

PRIMARY KEY("date")

);

CREATE TABLE Time(

"date" DATE NOT NULL,

"workday" BOOLEAN,

"fk\_qt" DATE,

"fk\_wt" DATE,

PRIMARY KEY("date"),

FOREIGN KEY ("fk\_qt") REFERENCES Quarter\_Time("date"),

FOREIGN KEY ("fk\_wt") REFERENCES Week\_Time("date")

);

CREATE TABLE Region(

"country\_code" varchar(3) NOT NULL,

"region" int,

"subregion" int,

"country" int,

"country\_name" varchar(100),

PRIMARY KEY("country\_code")

);

CREATE TABLE Cases(

"id" int NOT NULL,

"cases" int,

"deaths" int,

"cases100k" real,

"deaths100k" real,

"fk\_time" DATE,

"fk\_region" varchar(255),

PRIMARY KEY("id"),

FOREIGN KEY ("fk\_time") REFERENCES Time("date"),

FOREIGN KEY ("fk\_region") REFERENCES Region("country\_code")

);

Como aspecto destacable en esta estructura, podemos apreciar la necesidad de crear una tabla auxiliar *Time* para mantener la información de las dos tablas de dimensión temporal, cada una con las jerarquías necesarias. Esas referencias están en sus dos claves foráneas.

El siguiente paso es la lectura del CSV de casos de COVID-19 otorgado para la realización de este trabajo. También realizamos algunos procesos de limpieza del conjunto de datos para que sean válidos y usables, los cuáles describiremos a continuación.

En la ilustración 2 podemos ver todo el proceso realizado para dicha lectura de casos.

El CSV cuenta con cinco columnas que contienen: fecha (dd/MM/yyyy), casos, muertes, país (con el código textual, por ejemplo, Afganistán 🡪 AFG) y población en 2018 para dicho país.

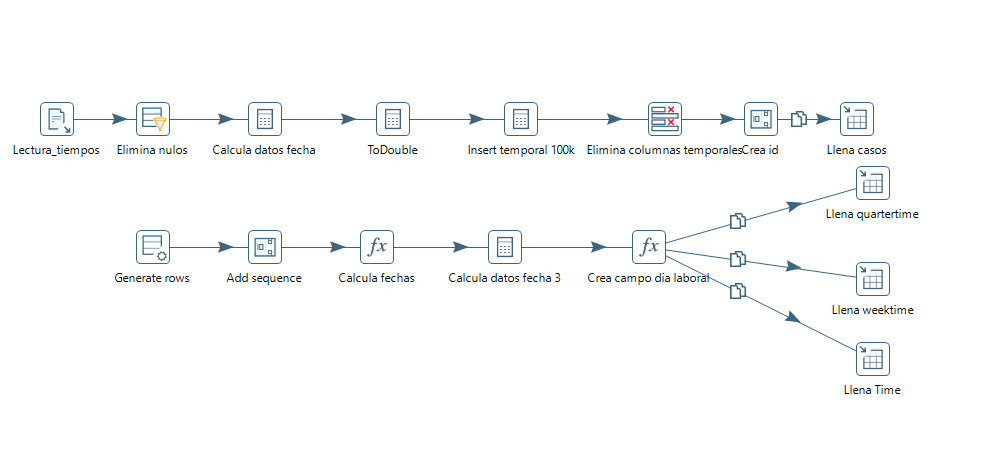


Ilustración - Lectura de casos

## Tabla de hechos

Empezando con la rama de arriba de la ilustración 2, el primer paso consiste en eliminar aquellas filas que contienen campos de interés nulos. Una primera revisión manual de los datos nos revela que los únicos nulos existentes se encuentran en el código de país y en la población de 2018. Por lo tanto, estos son los campos que revisaremos que no sean nulos.

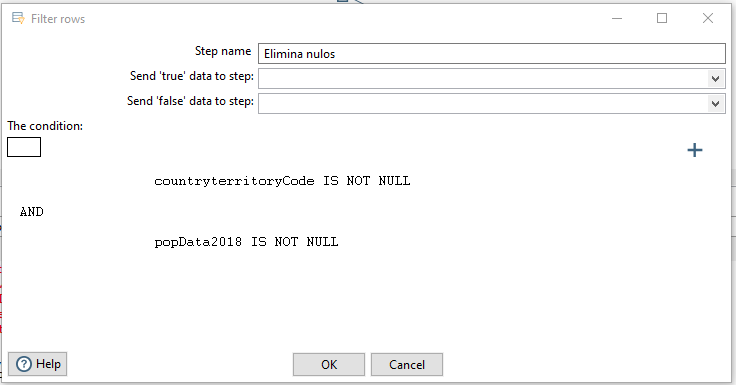


Ilustración - Eliminando nulos del fichero de casos

La ilustración 3 muestra que el proceso descrito se realiza con un *Filter rows,* seleccionando como condición las filas oportunas.

El siguiente paso es calcular las distintas partes de la fecha que usaremos en las tablas de dimensión temporal, realizado como se muestra en la ilustración 4. En ella podemos apreciar el uso de la herramienta *Calculator,* con la cuál extraemos de la columna *dateRep* todos los valores necesarios para la estructura de nuestro *datawarehouse.*

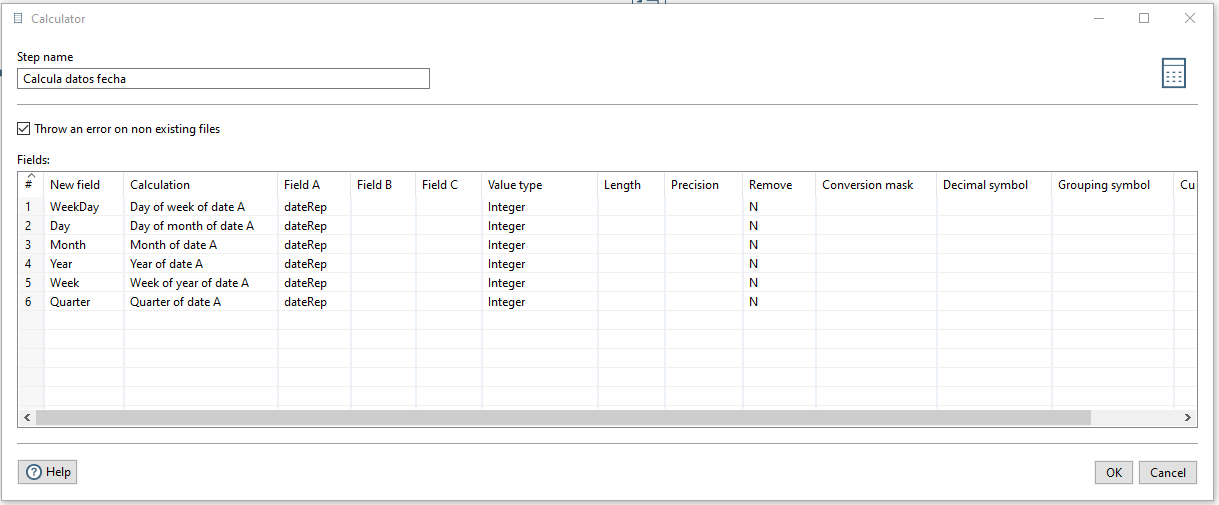


Ilustración - Calcula los datos de la fecha

El siguiente paso, *toDouble,* pasa algunos datos a tipo de datos numéricos con punto flotante para poder operar correctamente a continuación.

En el próximo proceso se crean cinco campos temporales para poder realizar cálculos con la herramienta *Calculator,* con lo que obtenemos el número de casos y fallecios por cada 100.000 habitantes. La ilustración 5 refleja dicho proceso.

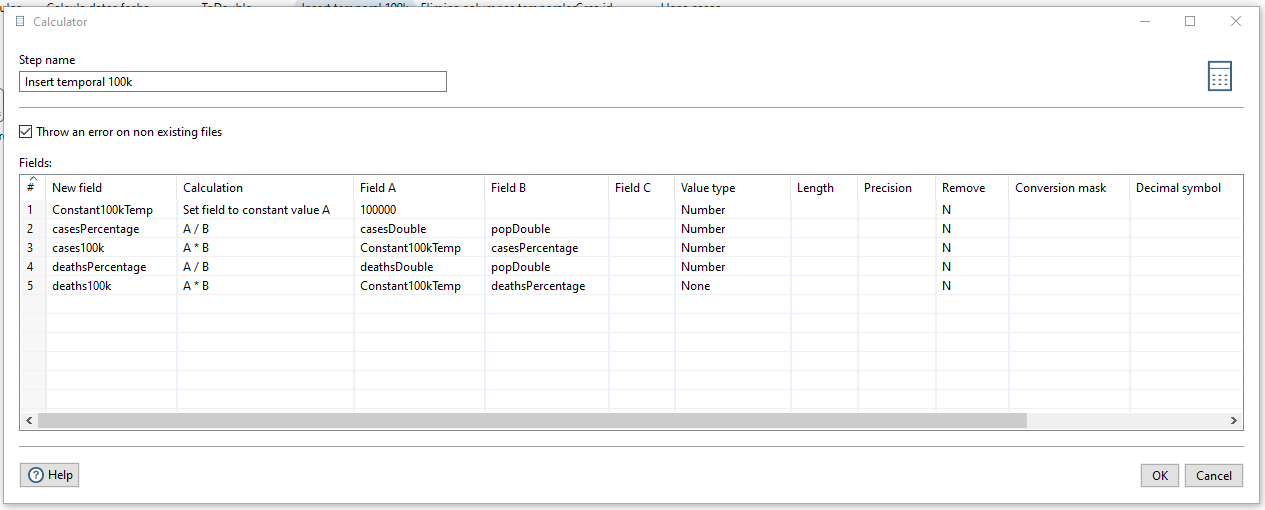


Ilustración – Creación de campos temporal para obtener casos y fallecidos por cada 100K habitantes

Una vez realizado el cálculo por cada 100.000 habitantes, procedemos a eliminar las columnas innecesarias, usando el *Select values* para este propósito, como se muestra en la ilustración 6, situada a continuación.

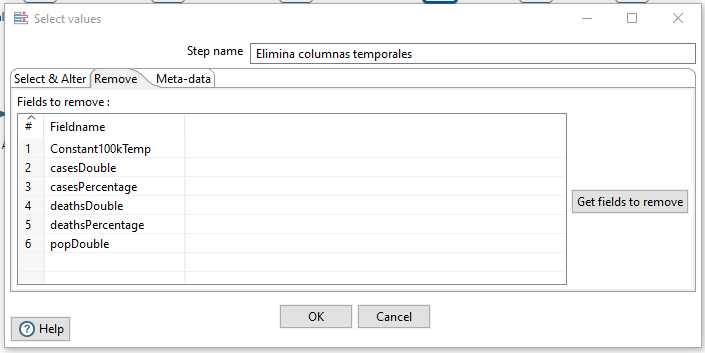


Ilustración - Eliminar campos temporales

El último paso antes de suministrar datos a la tabla de *Cases* consiste en asignarle a cada fila un identificador, usándose en este caso el Add sequence que tenemos disponible en Pentaho.

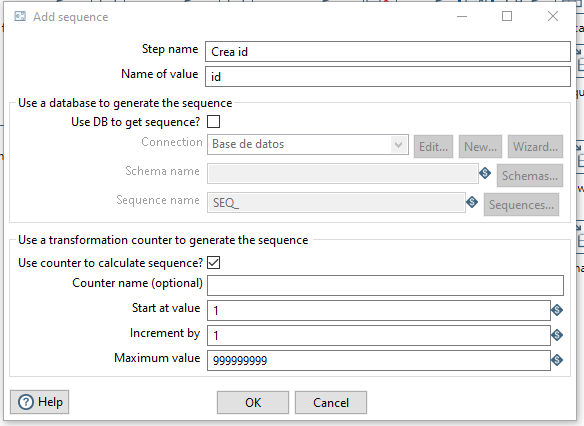


Ilustración - Crea identificadores para los casos

Al final, la tabla de hechos, en la que se almacenan los casos da como resultado la ilustración 8.

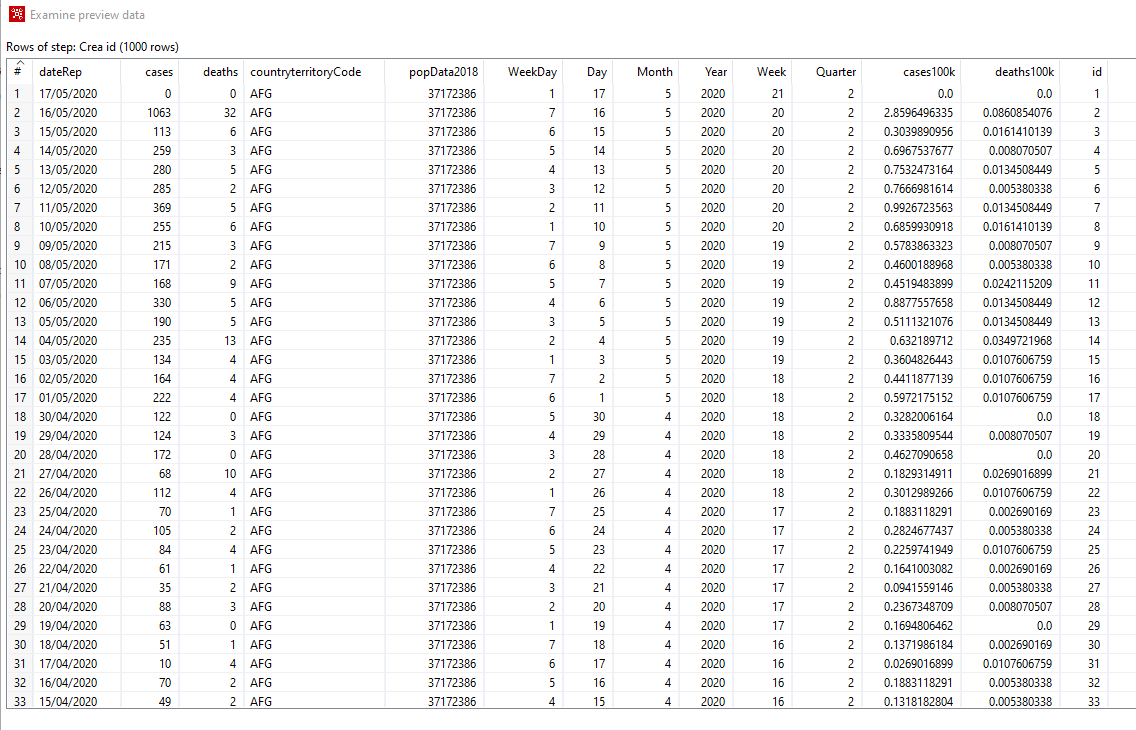


Ilustración - Resultado final tras el último paso para la tabla de casos

## Tablas de dimensión temporales

A continuación, abarcaremos la rama inferior de la ilustración 2. En el primer paso, generamos filas equivalentes a cada día de cada mes de los años 2019 y 2020. Debido a que 2020 es año bisiesto, el límite se establece a 731 fechas (365 + 366). Para ello, usamos el Generate rows, visible en la ilustración 9.

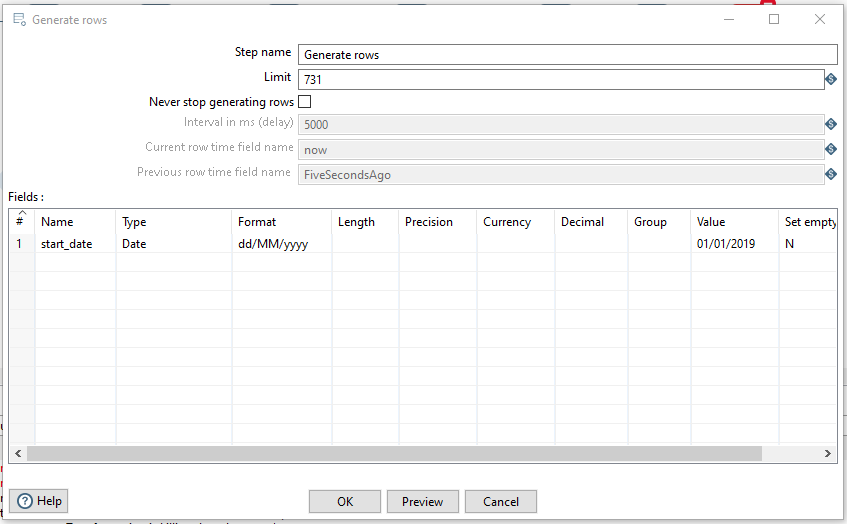


Ilustración - Generación de filas equivalentes a fechas desde el inicio de 2019 hasta finales de 2020

No obstante, el paso anterior solo nos da el primer día de 2019 731 veces. Esto lo solucionaremos al generar unos identificadores como lo hicimos en otro momento con *Add sequence.* Tras esto, usaremos el campo generado (al cuál llamamos *inc*) dentro de un proceso de fórmula, con el que obtendremos la fecha que le corresponde a cada fila. Dicha fórmula se puede observar en la ilustración 10.

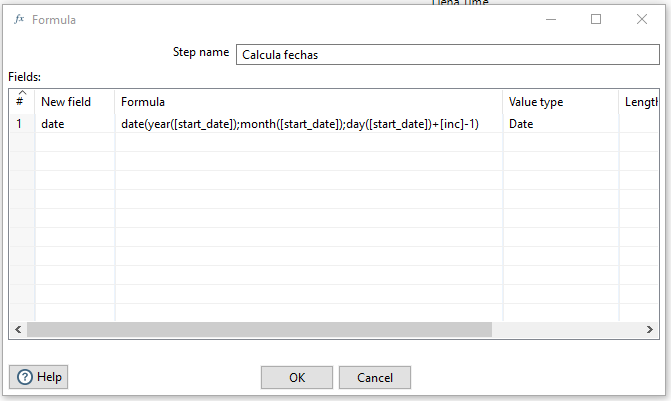


Ilustración - Creación de las fechas correctas

A continuación, en el paso ‘Calcula datos fecha 3’, extraemos de cada fila de fecha generada los campos que necesitamos para la estructura de nuestro *datawarehouse*, visible en la ilustración 11.

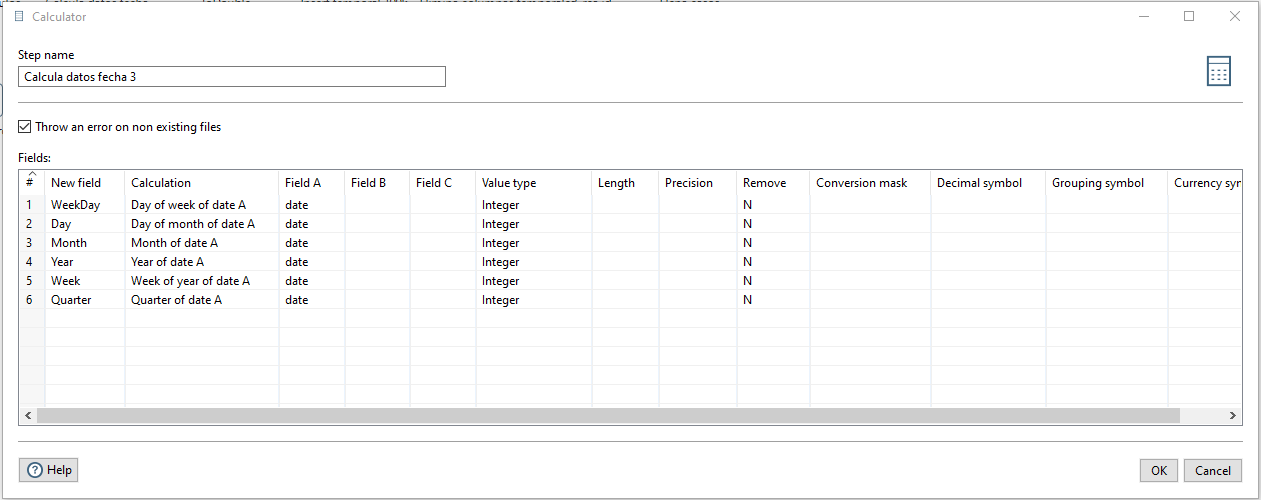


Ilustración - Generación de las jerarquías para las tablas de dimensión.

El último paso antes de llenar las tablas de dimensión es crear un campo que indicará en cada fila si esa fecha era día laboral o no, usando en la herramienta de fórmulas un rango muy básico, como se muestra en la ilustración 12. Cabe destacar que WeekDay = 0 es el domingo y WeekDay = 7 es sábado.

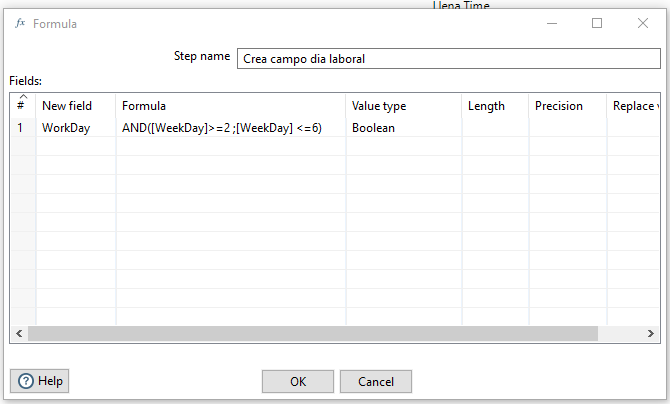


Ilustración - Creación del campo día laboral

Una vez hecho esto, asignamos cada valor a su respectiva tabla, ya sea *Time*, *Week*\_*Time* o *Quarter*\_*Time.* Con esto, terminamos con el proceso de ‘lecturas\_caso’ visto en la ilustración 1.

## Tabla de dimensión de región

Este proceso se realiza en la última etapa del trabajo principal y se denomina ‘lectura\_países’. Los pasos realizados en él son los mostrados abajo, en la ilustración 13.

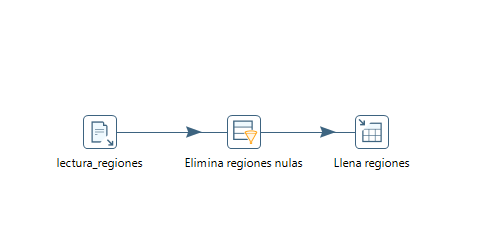


Ilustración - Lectura del CSV de países

Como se puede observar, solamente requerimos de una limpieza de nulos antes de rellenar la tabla de dimensión dedicada a las regiones. En la ilustración 14 se puede visualizar las columnas revisadas para el filtrado.

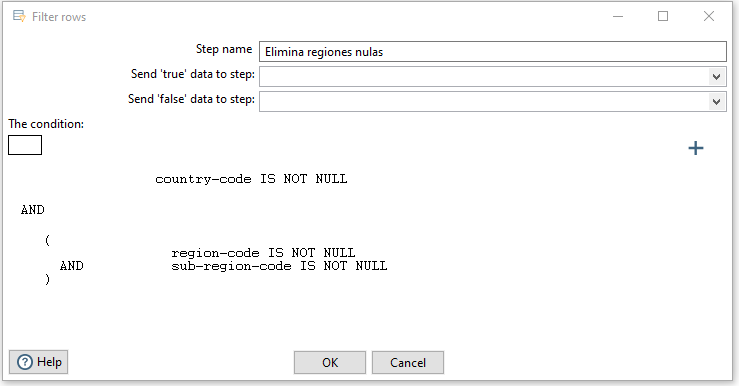


Ilustración - Eliminación de nulos para las regiones

Esto nos deja un resultado como el de la ilustración 15. Los campos que quedan a nulo no se usan en nuestro *datawarehouse* y por ello, no se realizó ningún proceso con ellos.

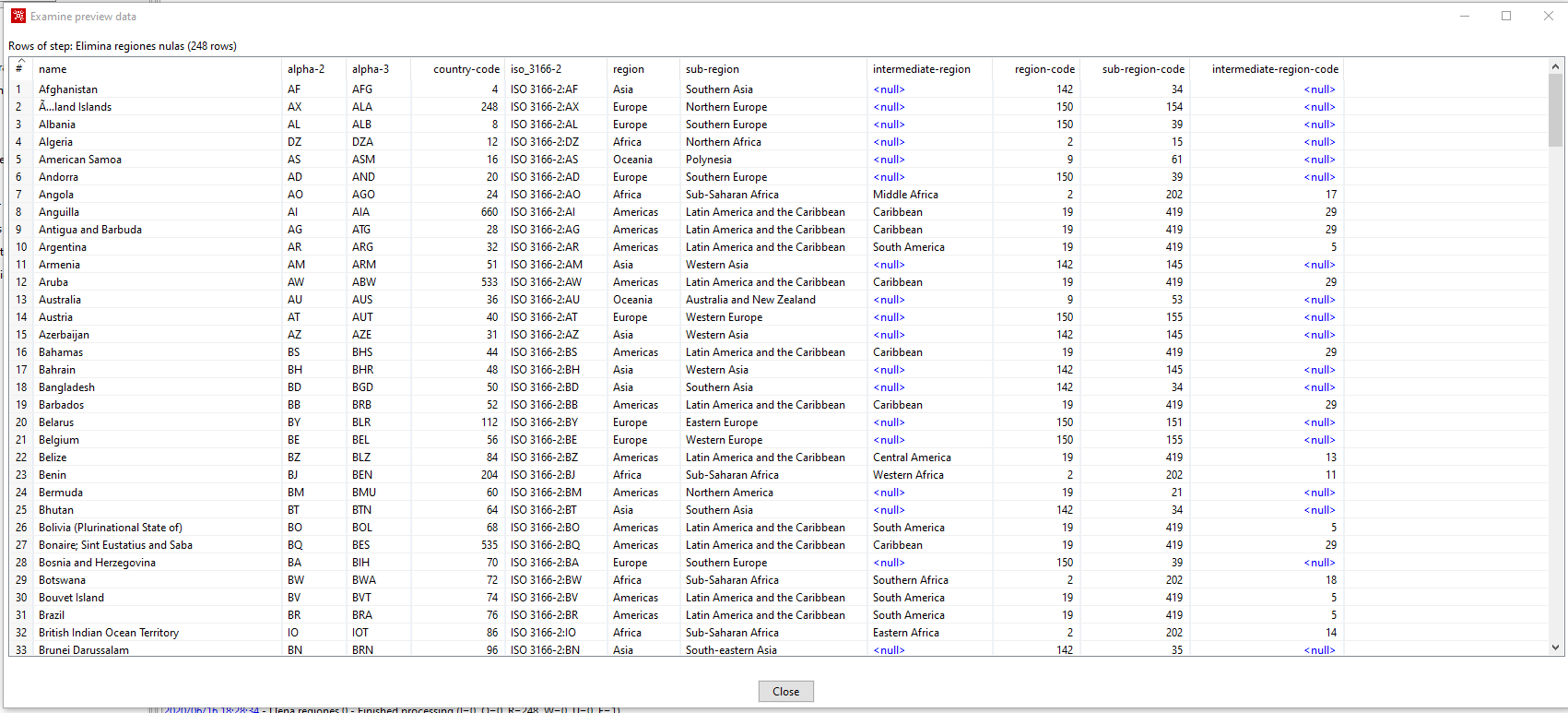


Ilustración - Resultado para rellenar la tabla de regiones

Con esto, concluye el desarrollo del *datawarehouse.* La lectura posterior le mostrará el uso de todo el trabajo ya explicado para la generación de informes dinámicos.

# Generación de informes

Para la generación de informes, realizamos una consulta a nuestras tablas en base a dos parámetros que el usuario puede insertar: el mes y el país. Dichos parámetros se muestran en la cabecera del informe, mientras que en la sección de detalles mostramos la fecha, el número de casos y muertes, tanto totales como por cada 100.000 habitantes. Pentaho Report Designer funciona con una interfaz visual de plantillas que nos permite acomodar estos valores en un lugar concreto del documento, como se muestra en la ilustración 16.

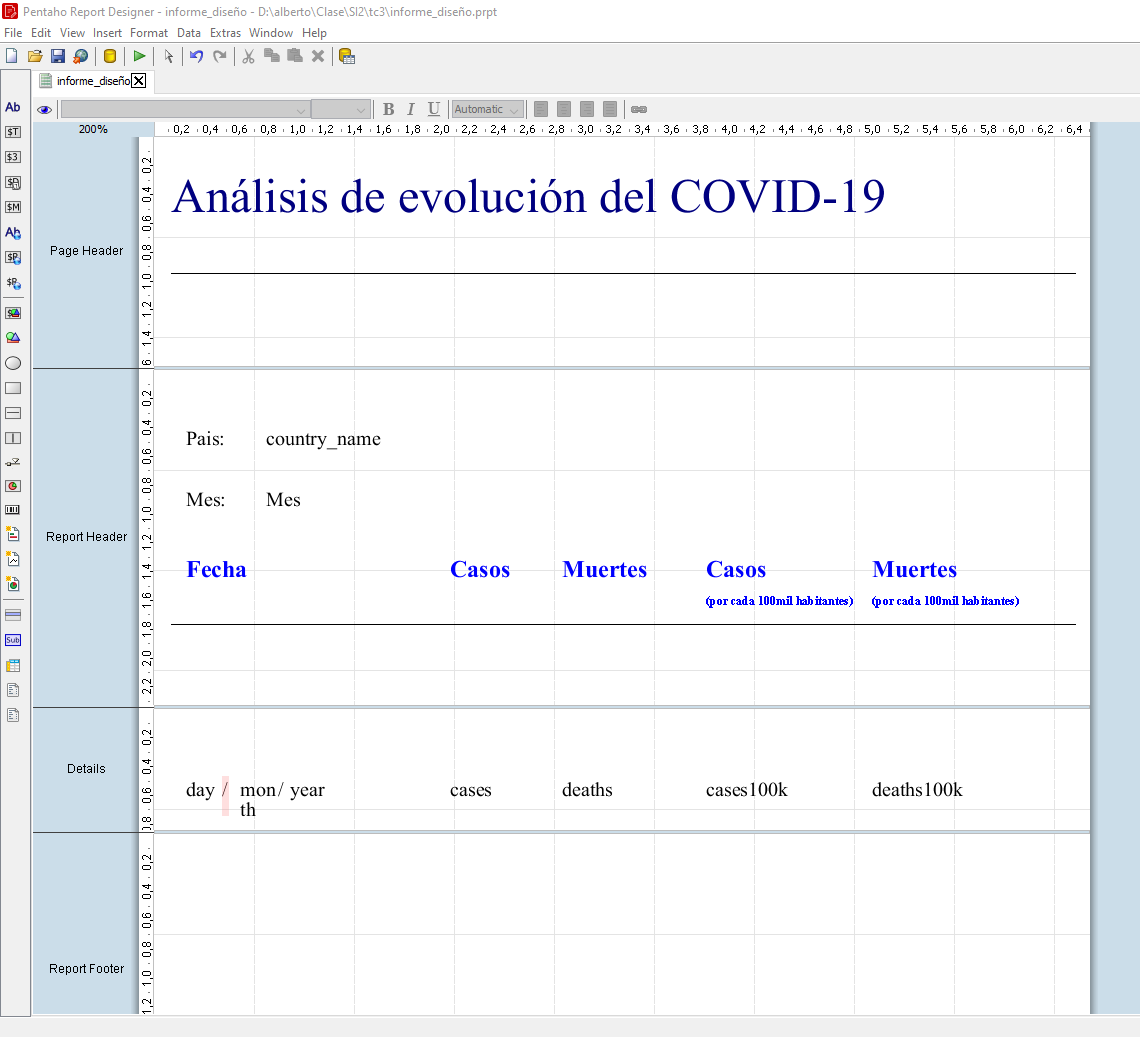


Ilustración - Plantilla informe

La consulta que nos permite obtener los campos se puede ver en la ilustración 17, así como un ejemplo de informe en la ilustración 18, las cuales están en la siguiente página.

Con esto concluye todo nuestro trabajo final de inteligencia de negocio para Sistemas Inteligentes II.

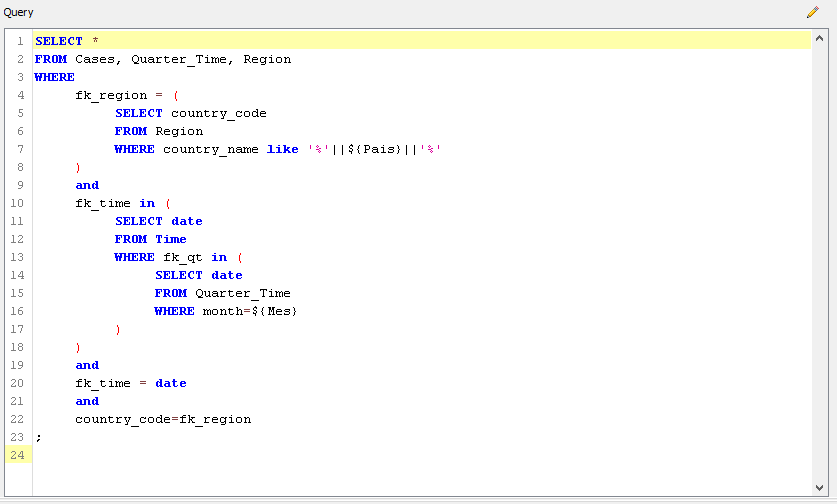


Ilustración - Consulta para generar el informe

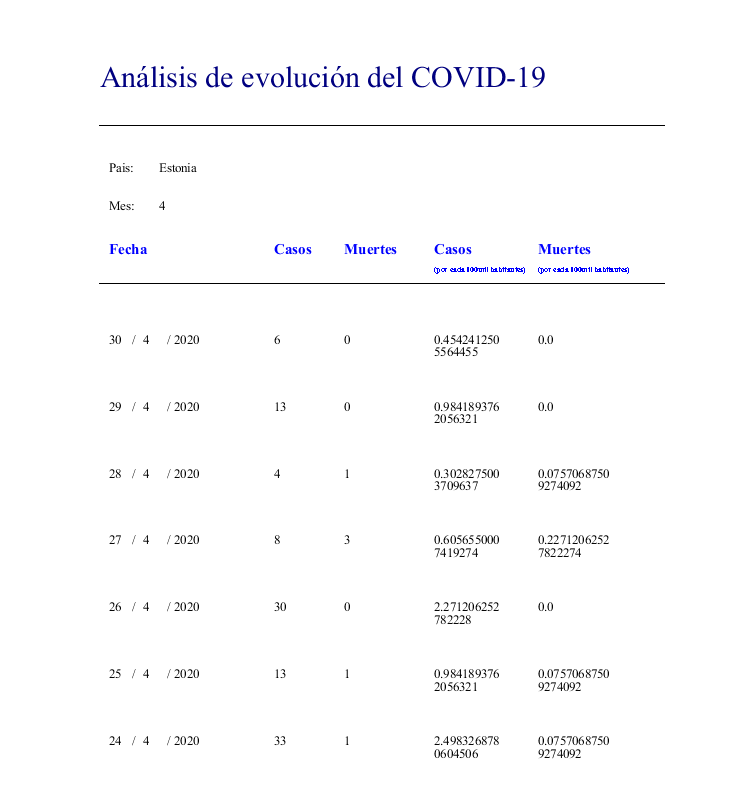


Ilustración - Resultado de una generación de informe