

# Documentación Técnica

## Análisis de necesidades

### Orientación del Proyecto

El proyecto está orientado a analizar los datos del sector primario de la economía, es decir, la agricultura. En concreto, se tienen datos de recogida de fruto.

### Objetivos Generales del Proyecto

Los objetivos planteados para este proyecto son los siguientes:

- Mediante las métricas analizadas de los datos pasados, ofrecer una solución de inteligencia de negocio que sea útil para la persona adecuada respecto a hacer una previsión de negocio. Es decir, predecir donde poder invertir a tiempo en el sector donde más rédito le pueda ofrecer.
- Ofrecer una perspectiva anual de evolución del negocio en un sector determinado, para así ofrecer unas estadísticas de progreso, parón o deterioro del negocio.

### Métricas que se pretenden analizar

La cantidad recogida de cada tipo de cultivo.

### Características de los sistemas de origen y modelos de datos subyacentes

Los sistemas de origen son ficheros excel obtenidos del gobierno de España.

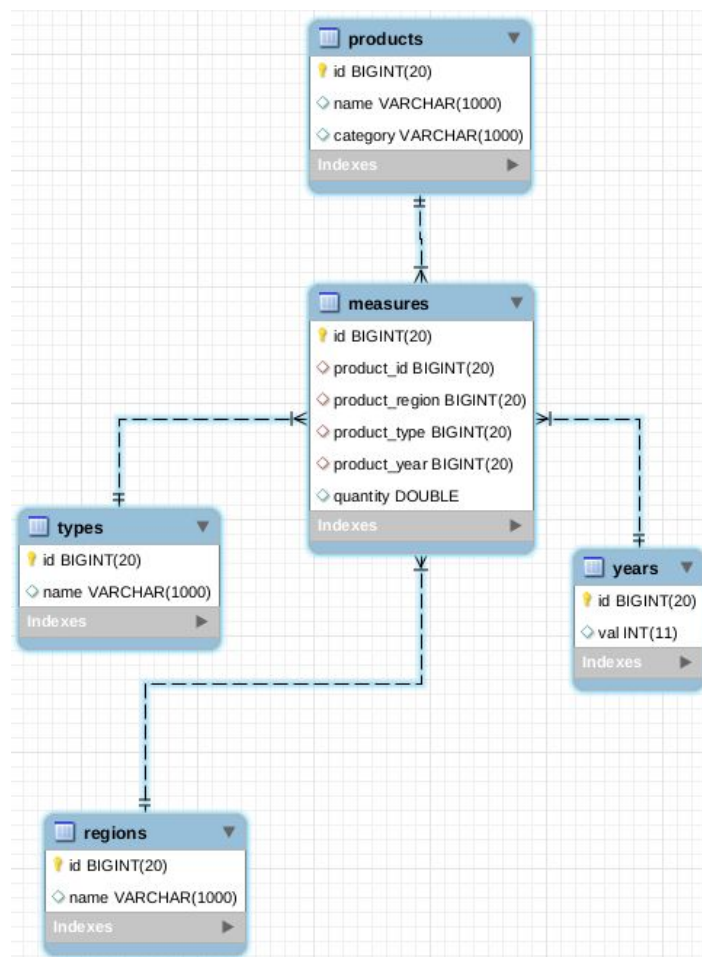
Los modelos de datos subyacentes son sistemas relacionales de base de datos, en concreto, el MySQL.

## Diseño e implementación

### Esquema ER para el Datawarehouse e implementación SQL

El esquema ER utilizado para crear el datawarehouse es como su propio nombre indica, un esquema de entidad relación propio de un sistema relacional. A continuación, presentamos el esquema en forma de diagrama y el código que ha permitido implementarlo en el servidor MySQL.

Así pues, el diagrama ER:



Como se puede apreciar, usamos un modelo en estrella, donde tenemos diferentes tablas de dimensiones y una tabla de hechos.

Por otra parte, presentamos el código SQL que hace posible la implementación del mismo en el motor de MySQL.

Así pues, el código:

```
SET FOREIGN_KEY_CHECKS = 0;

drop database if exists tin;
create database tin;

use tin;

drop table if exists products;
create table products(
    id bigint auto_increment,
```

```

    name varchar(1000),
    category varchar(1000),

    primary key (id)
);

drop table if exists regions;
create table regions(
    id bigint auto_increment,
    name varchar(1000),

    primary key (id)
);

drop table if exists types;
create table types(
    id bigint auto_increment,
    name varchar(1000),

    primary key (id)
);

drop table if exists years;
create table years(
    id bigint auto_increment,
    val integer,

    primary key (id)
);

drop table if exists measures;
create table measures(
    id bigint auto_increment,

    product_id bigint,
    product_region bigint default 1,
    product_type bigint default 1,
    product_year bigint default 1,
    quantity double default 0,

    primary key (id),

```

```

foreign key (product_id) references products(id) on delete
cascade,
foreign key (product_region) references regions(id) on delete
cascade,
foreign key (product_type) references types(id) on delete cascade,
foreign key (product_year) references years(id) on delete cascade
);

```

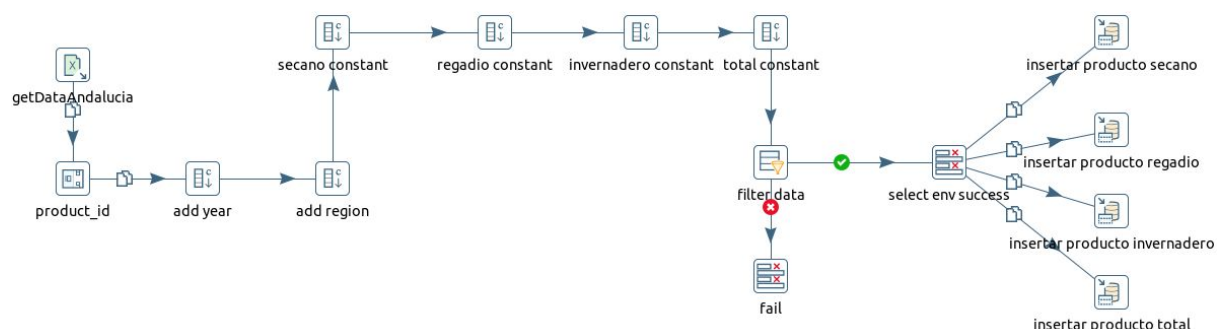
## Procesos ETL

Para hacer los procesos ETL hemos utilizado el Pentaho Data Integration (PDI), que nos permite hacer de forma sencilla y con interfaz gráfica, un proceso de Extracción, Transformación e Inserción en base de datos.

El proceso en todos a consistido en los siguientes pasos:

1. Seleccionar el fichero excel de donde queremos extraer la información.
2. Seleccionar las filas y columnas que nos interesan, con el tipo deseado.
3. Añadir alguna secuencia para poder compararlo con un id de base de datos.
4. Añadir según el fichero, una constante para identificar la región de donde se están extrayendo los datos.
5. Filtrar las filas que nos interesan para insertar en base de datos.
6. Insertar en base de datos, en concreto, en la tabla de hechos llamada “measures”.

Para mayor aclaración, adjuntamos una imagen de un proceso ETL. En concreto, de Andalucía. Así pues:



## Informes

Bien, respecto a este apartado, hemos usado el software Pentaho Report Designer, proporcionado por la misma fuente que el resto de los software de Pentaho usados.

Para ello, hemos usado la opción de “Report Design Wizard”, ya que, con esta opción nos permite crear un informe de forma sencilla y amena, pudiendo conectar de forma fácil la consulta SQL requerida con los campos a rellenar.

Por ende, ponemos las consultas SQL realizadas y pasamos a explicarlas. Así pues:

- **Consulta 1:**

```
select p.name as producto, t.name as tipo,
       round(m.quantity, 2) as "cantidad (kg)"
from
measures m join types t
on m.product_type = t.id
join products p
on m.product_id = p.id
where p.name != p.category and lower(p.name) like ${pProduct};
```

Como se puede apreciar en esta consulta, obtenemos los nombres de los productos, el tipo y la cantidad recogida en kilogramos. También podemos apreciar que tenemos opción de búsqueda por el nombre del producto, por si queremos acotar la longitud del informe.

- **Consulta 2:**

```
select r.name as "Comunidad Autonoma", ye.val as "Año",
       round(sum(m.quantity), 2) as "Cantidad"
from
measures m join regions r
on m.product_region = r.id
join years ye
on m.product_year = ye.id
group by ye.val, r.name
order by ye.val;
```

Como podemos observar, en esta consulta obtenemos las cantidades de fruto recogida por comunidad autónoma y por año.

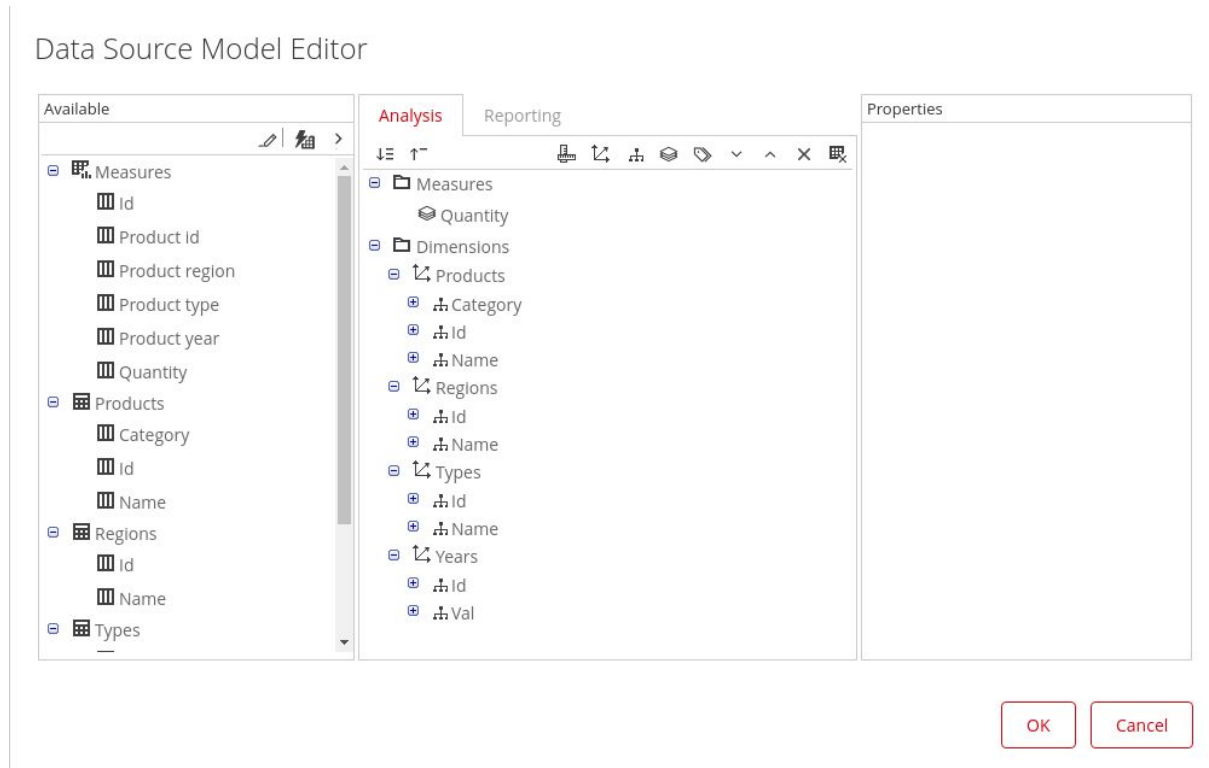
Por último, quedaría aclarar que esta parte es bastante sencilla de explicar dado que apenas tiene complejidad. La única que puede haber es a la hora de crear las consultas, pero si se tiene un buen dominio de MySQL, no es ningún problema.

## Cubos OLAP

Para crear el cubo OLAP, hemos utilizado el Manage Data Sources del Pentaho Server, que nos proporciona una interfaz sencilla y amena para poder crear el cubo OLAP sobre nuestro Datawarehouse y así poder realizar las consultas MDX en el posterior cuadro de mandos.

Por tanto, adjuntamos una imagen descriptiva de nuestro cubo OLAP terminado, indicando así las dimensiones y las métricas a analizar.

Así pues:



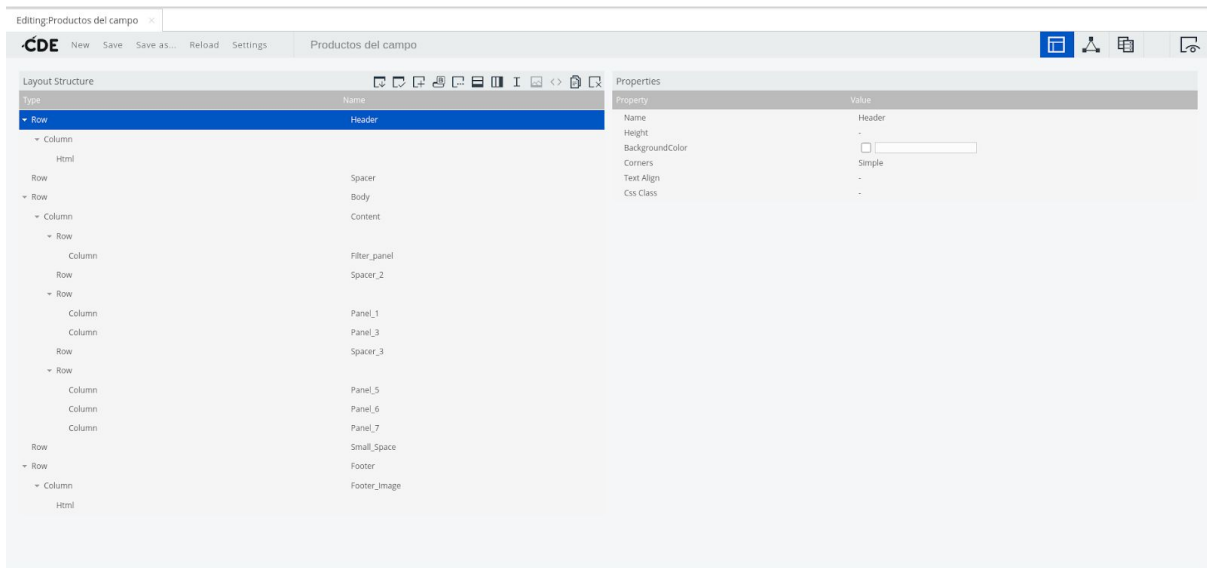
Como se puede apreciar, tenemos 4 dimensiones y solo observamos una sola métrica, que es la cantidad de fruto recolectado en nuestro contexto de productos agrícolas.

## Cuadro de mandos

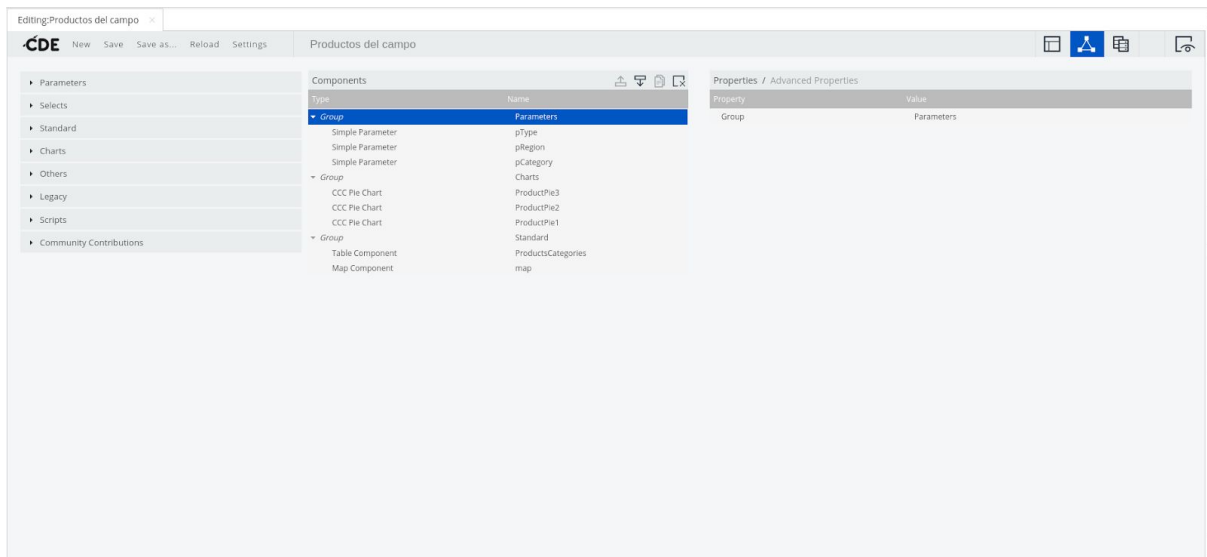
Bien, respecto a este apartado, en la creación de cuadro de mandos expondremos 4 imágenes, donde mostramos las 3 partes del cuadro de mandos en el editor del CDE, es decir, parte HTML, parte de elementos gráficos tales como tablas y gráficas y la parte de elementos de origen de datos (consultas SQL y MDX), y por último mostraremos la preview del cuadro de mandos, es decir, como queda finalmente.

Así pues, mostramos las partes del editor:

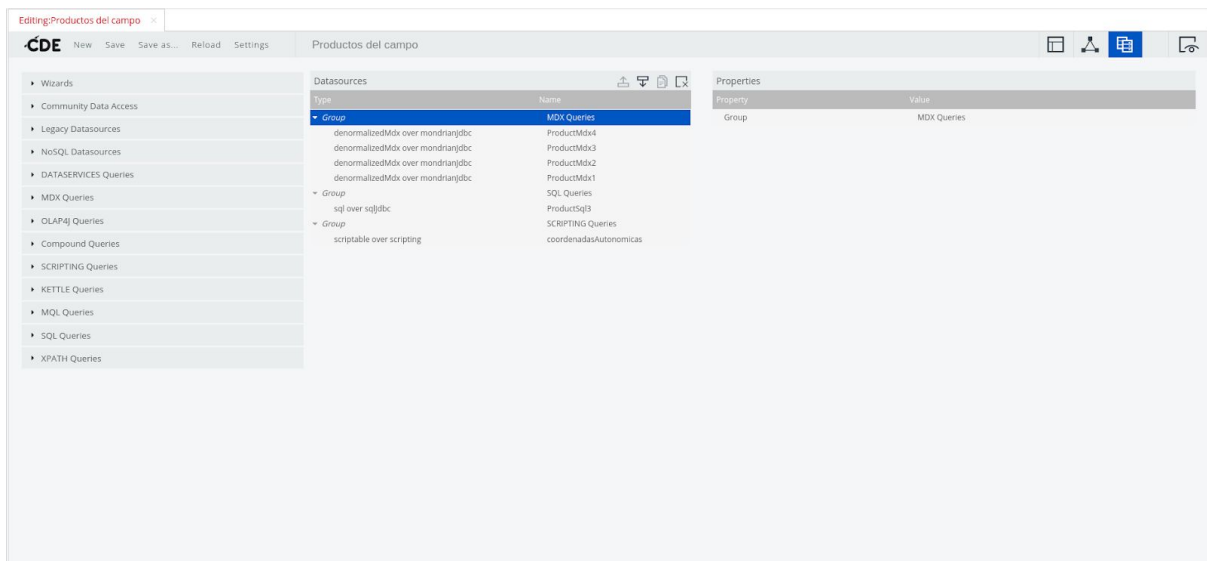
- Parte del DOM HTML:



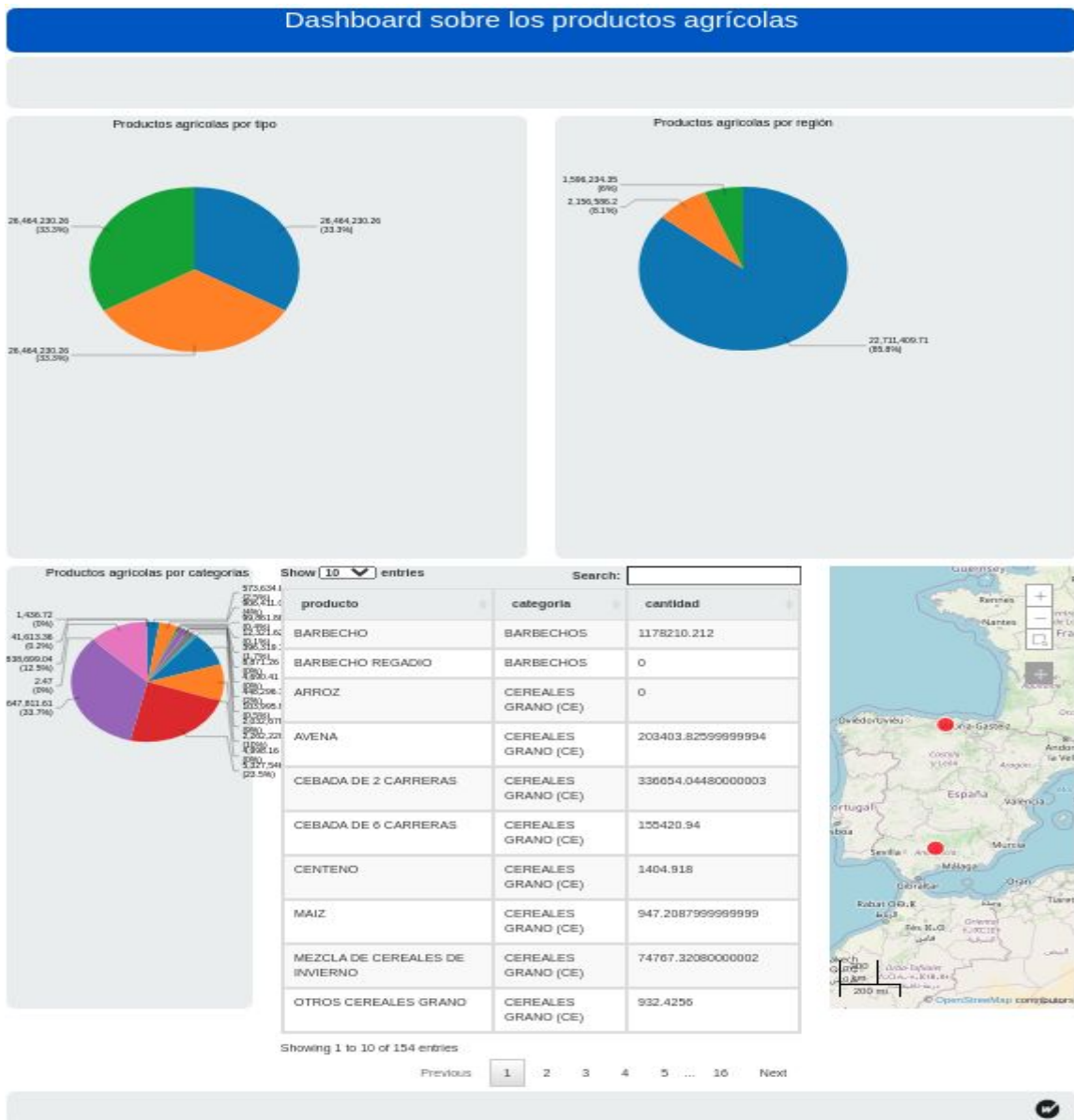
- Parte de los elementos gráficos:



- Parte de los datasource:



- Previsualización del Dashboard:





## Principales problemas durante el desarrollo

Ha habido dos principales problemas durante el desarrollo, que se han centrado en el data integration y en la realización del cuadro de mandos.

Respecto al data integration, a la hora de obtener los datos de forma adecuada, ya que, siempre no lo daba en formato "String", y no en tipo "Integer" o "Float". Esto fue solventado mirando la documentación de pentaho.

Respecto a la creación del cuadro de mandos de pentaho, los problemas fueron constantes uno detrás de otro, desde poner bien la información en los datasources para crear una adecuada consulta SQL o MDX hasta como crear los gráficos interactivos. Por suerte, gracias a la documentación de Pentaho y a la ayuda de nuestro profesor de prácticas, estas dudas fueron resueltas.

Otro problema durante la realización del cuadro de mandos es poder hacer los puntos del mapa interactivos, pero debido a diversos errores y falta de tiempo, se optó dejarlo como ayuda visual para saber de qué regiones se ha extraído información y nada más. Tal vez en una futura iteración, terminar de hacerlo bien.

## Pruebas

Las pruebas del software desarrollado son prácticamente las que se deben hacer en cualquier software mínimamente decente, es decir:

- Comprobar que después de lanzar los procesos ETL la información se escribía de forma correcta en la base de datos. Al principio cuando teníamos poco datos a mano, posteriormente con una consulta SQL donde se comprobaba que no hubiera ningún campo nulo o fuera de su rango permitido en el tipo de datos establecido.
- Si el proceso ETL daba algún error, comprobar el fichero de Log.
- Comprobar que después de hacer el cubo OLAP con el Manage Data Source de Pentaho Server, el cubo OLAP mostraba la información correcta. Esto lo hacíamos con la herramienta Pivot4J View que se descarga a través de la marketplace de Pentaho Server.
  - Para instalar la herramienta Pivot4J View se debe instalar el plugin llamado Pivot4J Analytics que se encuentra en la marketplace de Pentaho 8.2.
- Si el cubo OLAP daba error, investigar el error y en muchas ocasiones, rehacer el cubo OLAP.

- Respecto al Dashboard, las pruebas que se hacían eran manuales clickando en los elementos de las gráficas e interactuando con las tablas para que el resultado obtenido fuera el esperado.
- En resumen, todas las pruebas que se han hecho se han basado en prueba y error. De forma manual todas ellas, sin ningún tipo de automatización.

## Instrucciones de despliegue y uso

Bien, para este apartado voy a hacer una guía de instalación y configuración para el sistema GNU/Linux, en concreto para la distribución Ubuntu 18.04.

Así pues, empecemos con los pasos a seguir:

- **Java:**
  - Instalar [SDKMAN](#)
  - Ejecutar la instrucción **sdk install java 8.0.252-open**
  - Ejecutar la instrucción **sdk use java 8.0.252-open**
  - Escribir en la consola **java -version** para comprobar que la instalación se ha realizado de forma correcta.
- **MySQL:**
  - **sudo apt install mysql-client mysql-server mysql-workbench**
  - **sudo su -**
  - **mysql**
  - **create user 'guillermo'@'localhost' identified by 'guillermo';**
  - **grant all privileges on \* . \* to 'guillermo'@'localhost';**
  - **flush privileges;**
  - **mysql -u guillermo -p < sql.sql # sql.sql es el archivo con el datawarehouse**
  - **\*\* En el archivo database.sql hay un dump de la base de datos completa, con datos incluidos**
- **Pentaho Data Integration:**
  - Descargar de aquí: [PDI](#)
  - Iniciar con el archivo spoon.sh
  - Abrir los archivos con extensión **.ktr**
  - Modificar el path del “Excel input” al archivo excel. Los archivos fuentes están en la carpeta “**fuentes**” del repositorio expuesto más abajo.
  - Ejecutar la transformación
- **Pentaho Report Designer:**
  - Descargar de aquí: [PRD](#)
  - Iniciar con el archivo report-designer.sh
  - Usar el “**Report Design Wizard**”

- Desarrollar las consultas SQL con los correspondientes join
  - Generar el pdf con la herramienta
- **Pentaho Server:**
  - Descargar de aquí: [PS](#)
  - Iniciar con el archivo start-pentaho.sh
  - Abrir el navegador con la el dominio localhost y el puerto 8080, de la siguiente manera: <http://localhost:8080>
  - Logearse como admin
  - Importar los 3 archivos correspondientes al dashboard, con extensiones **.cda**, **.cdfde** y **.cde**
- **Repositorio con el código fuente:**
  - [GitHub Guillermo López García](#)