Universidad Zaragoza

Trabajo Fin de Grado

Sistema para la integración de productos fitosanitarios

Autor

Catalin Dumitrache

Director

Francisco Javier López Pellicer

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2017

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. _____,

con nº de DNI _____ en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) _____, (Título del Trabajo)

_____

_____

_____

_____

_____

_____,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, _____

Fdo: _____

AGRADECIMIENTOS

Agradezco a ...

RESUMEN EJECUTIVO

Este proyecto se centra en presentar y demostrar un modelo de integración de información relativa a productos fitosanitarios y su aplicación en diferentes productos agrícolas provenientes de fuentes heterogéneas en un esquema único y escalable.

Para lograr esto, se han seleccionado diferentes fuentes de datos sobre productos fitosanitarios, presentes en distintos formatos y se han analizado varias tecnologías de integración para elegir el stack tecnológico que más se adecue al problema en cuestión.

Así pues, en este proyecto se han usado herramientas de almacenamiento elegidas bajo un criterio de escalabilidad futura y herramientas de procesado de datos bajo el criterio de proporcionar soporte al mayor abanico de fuentes heterogéneas posibles. *Apache Hadoop* fue el indicado como sistema responsable del almacenamiento en conjunto con *Apache Hive* para facilitar el acceso a los datos mediante una aproximación relacional. Para el procesado de los datos se ha empleado *Talend Big Data*.

Para demostrar la viabilidad del sistema, se ha desarrollado un prototipo funcional que recoge datos de los productos fitosanitarios de España y Europa y complementa la información de ambas fuentes, constituyendo un primer paso hacia ese modelo compartido donde varias fuentes heterogéneas concuerdan en un mismo esquema congruente. Los datos se pueden ver mediante una aplicación web desarrollada con *JHipster*, un generador de proyectos ligero sobre Java capaz de desplegar rápidamente una aplicación con una cuidada GUI.

Palabras clave: Productos fitosanitarios, Hadoop, Hive, Integración, ETL, Base de datos, JHipster, MySQL, Modelo único, BigData, Escalabilidad, Sqoop, Fuentes heterogéneas.

Tabla de contenidos

Capítulo 1

Introducción

Actualmente, el proceso de consulta del uso y aplicación de los productos fitosanitarios sobre determinados productos resulta una tarea en ocasiones tediosa, sobre todo, cuando se trata de comprobar las especificaciones y regulaciones que imponen diferentes países en operaciones de importación o exportación de determinados productos agrícolas. Hoy en día, una persona que quiere comercializar estas sustancias debe tener en cuenta varios factores; en primer lugar, existen varios manuales extensos que recogen medidas de seguridad, buenas prácticas y pautas sobre la aplicación de los productos fitosanitarios. Dichos manuales se deben cumplir en todo momento [manual seguridad, aplicación fitosanitarios, buenas prácticas]. No solo eso, sino que por otra parte, las bases de datos o almacenes que recogen la información sobre sustancias autorizadas en muchas ocasiones no están bien gestionadas, son difíciles de encontrar, la información se presenta en formatos heterogéneos e incluso se puede encontrar desactualizada. Los avances conseguidos en este TFG, pretenden reducir la complejidad de esa tarea de búsqueda de información acerca de productos fitosanitarios, facilitando a una persona el acceso a un esquema común con toda la información centralizada y actualizada.

El objetivo principal de este proyecto es **proponer y validar un proceso de recogida, transformación y presentación de la información sobre productos fitosanitarios con el resultado en forma de esquema compartido estandarizable** que pueda beneficiar tanto a agricultores como a instituciones nacionales o internacionales, **y facilitar la consulta de dicha información de manera más rápida, simple y accesible que los métodos actuales**.

Entre los retos planteados figuran:

- Desarrollar un sistema capaz de visualizar los datos ya integrados nutriéndose únicamente de las fuentes originales sin intervención de una persona en el proceso
- La consistencia de los datos y su almacenamiento tanto en formato original como

en su formato procesado e integrado en la versión final

- El diseño de una solución escalable y actualizada en todo momento
- La posibilidad de añadir características de trazabilidad y mantenimiento a la aplicación.

1.1. Estructura del documento

Este documento se presenta dividido en varios bloques conceptuales; el primero abarca los primeros tres apartados (Resumen ejecutivo, Introducción y Definiciones) y se corresponde a una introducción al trabajo realizado. En él, se ofrece una visión completa y resumida del problema junto con su solución y se dan algunas definiciones técnicas de algunos de los términos empleados en esta memoria. El bloque de análisis abarca los apartados 4 y 5 (Phytoscheme y Análisis) y presentan el trabajo de análisis que se llevó a cabo, desde el estudio del entorno, hasta el análisis del stack tecnológico, pasando por el de riesgos y la captura de requisitos. El tercer bloque se corresponde a la solución desarrollada en sí; abarca los apartados 6 y 7 (Diseño e implementación) y se plasma el trabajo desde las fases tempranas del desarrollo de la solución hasta el momento de la finalización del software como solución tecnológica al problema. Los últimos bloques se corresponden a los detalles de gestión del proyecto, a las conclusiones tanto del proyecto como del alumno a nivel personal y a los diferentes anexos recogidos durante toda la duración del TFG.

Capítulo 2

Phytoscheme

2.1. Contexto y necesidades reales

Los productos fitosanitarios son un elemento imprescindible en la producción agrícola, tanto en los sistemas convencionales de agricultura como en los sistemas de agricultura integrada o ecológica. Sin su existencia, muchos cultivos de las zonas de producción de mayor interés económico y social serían inviables hoy en día debido a los estragos potenciales de las diferentes clases de plagas.

No obstante, el uso de dichos productos fitosanitarios debe estar regulado ya que una aplicación indebida de los mismos puede tener otros efectos no deseables. Dichos efectos de ninguna manera deben suponer un peligro para la salud humana y tampoco riesgos inaceptables para el medio ambiente.

Por ello el Estado sólo aprueba la comercialización de aquellos productos fitosanitarios que sean útiles para combatir las plagas pero no impliquen otros riesgos colaterales. Para que un producto fitosanitario pueda comercializarse debe estar inscrito necesariamente en el Registro Oficial de Productos Fitosanitarios - *Página web del ministerio de agricultura y pesca, alimentación y medio ambiente de España* [1]

Es, por tanto, necesario y casi obligatorio que la información del Registro de Productos Fitosanitarios llegue de manera precisa a todos los implicados en el área del uso de los productos fitosanitarios

La *Directiva 2009/128/CE* [2] establece el marco de la actuación comunitaria para conseguir un uso sostenible de los productos fitosanitarios. Esta Directiva implica, por ejemplo, la obligación del registro del uso de productos fitosanitarios. Un ejemplo del desarrollo de esta Directiva es el *Cuaderno de Explotación* [3]. Este cuaderno aglutina de manera ordenada y armonizada todos los elementos que deberán registrar los titulares de las explotaciones agrícolas, con el objetivo de facilitar el cumplimiento de la Directiva.

Actualmente, hay varias empresas que ofrecen aplicaciones (p. ej. *aGROSLab* [4],

Agricolum [5] o el *Cuaderno de Campo Agronev* [6]) que implementan el Cuaderno de Explotación. Un valor añadido que suelen incorporar estas aplicaciones es una base de datos con información acerca de los productos fitosanitarios autorizados. El problema al que se enfrentan estas empresas es que esta información no se publica de forma uniforme en toda la Unión Europea. Es decir, hay al menos un publicador por país miembro, la información publicada es heterogénea y los formatos normalmente son difícilmente accesibles. Además, a nivel Europeo, hay una *base de datos de referencia* [7] de las restricciones más o menos comunes en el uso de productos fitosanitarios. Dado que no hay un estándar de publicación establecido, una consecuencia adicional de esta situación es que es complicado verificar si un producto tratado con una serie de productos fitosanitarios en un país miembro puede ser exportado a otro, ya que la normativa fitosanitaria aplicable puede diferir.

En cuanto a la importación de productos desde un país miembro de la Unión Europea, el artículo 52 del *Reglamento (CE) 11/07/2009* [8] se refiere a este trámite como comercio paralelo y se especifica que para poder llevarlo a cabo, el Estado Miembro donde se desee introducir deberá determinar que el producto fitosanitario es idéntico en composición a otro ya autorizado en su territorio al que se denominará "producto de referencia". Para realizar el trámite, actualmente el proceso consta en rellenar la *solicitud* del certificado [9] (disponible en la página web del ministerio), entregarla a la Subdirección General de Higiene Vegetal y Forestal y esperar a que sea aprobada - *Preguntas frecuentes Mapama* [10]

Los principales problemas actuales de este proceso vienen por una parte en el lado del agricultor/empresa que solicita la importación puesto que no solo el tiempo de respuesta en ocasiones puede ser muy largo (de hasta de 45 días - Artículo 52, punto 2 del Reglamento (CE) 11/07/2009) sino que además, el agricultor o bien tiene poca información acerca de los productos permitidos en una importación/exportación o el acceso a dicha información es bastante tedioso y complicado. Por otra parte en el lado de la institución certificadora encargada de comprobar la solicitud el proceso no es del todo eficiente debido al hecho de tener que verificar manualmente si el producto cumple con los requisitos de composición del correspondiente producto en el país de importación/exportación. Estos problemas se pueden observar claramente en el siguiente diagrama, que muestra un flujo típico en un trámite de comercio paralelo entre un agricultor/empresa que quiere importar un producto a un determinado país, en este caso, España:

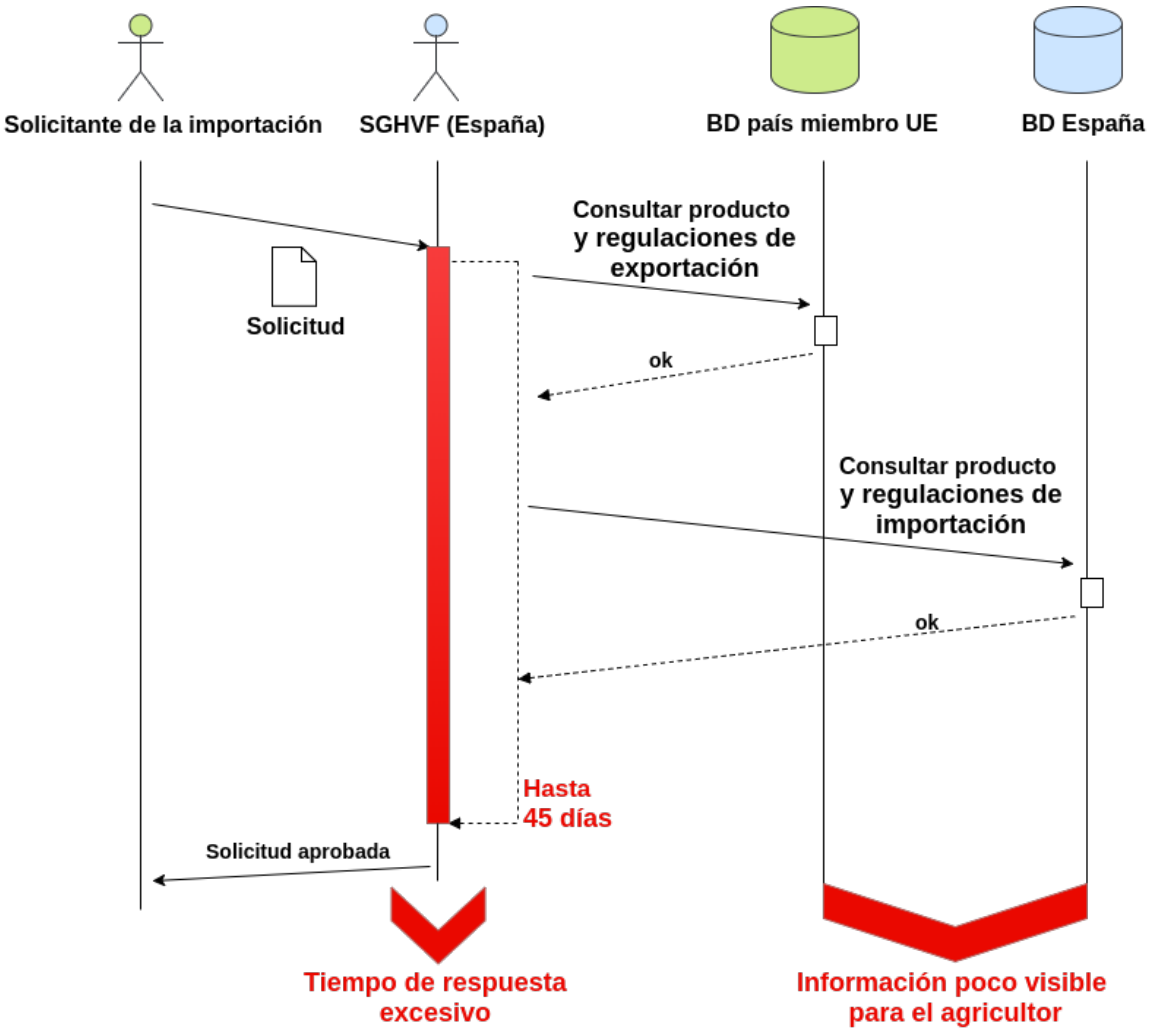

Figura 2.1: Diagrama de flujo del proceso actual de importación de un producto fitosanitario

*SGHVF = Subdirección General de Higiene Vegetal y Forestal

Es razonable desarrollar un sistema que provea un esquema único donde los datos estuvieran integrados y congruentes entre los diferentes países de la unión europea para facilitar un consumo posterior por las aplicaciones, e, incluso, directamente por los agricultores.

2.2. Motivación y objetivos

Habiendo visto el panorama descubierto en el apartado anterior, era evidente que se podían introducir mejoras al proceso actual que pueden beneficiar a las partes involucradas (tanto a agricultores y empresas como a las instituciones reguladoras). Con ese objetivo en mente, se propone desarrollar una aplicación prototipo que

valide el proceso de integración de la información de diferentes países en un esquema único que recogiese los datos que de otra manera tendrían que ser recopilados manualmente, con largos tiempos de espera y con una incertidumbre por parte de los agricultores/empresas en lo que a datos sobre productos fitosanitarios respecta.

El siguiente diagrama muestra el potencial proceso que tendría que seguir una persona interesada en realizar una importación y los problemas que sería capaz de solucionar el desarrollo de la aplicación planteada en este proyecto:

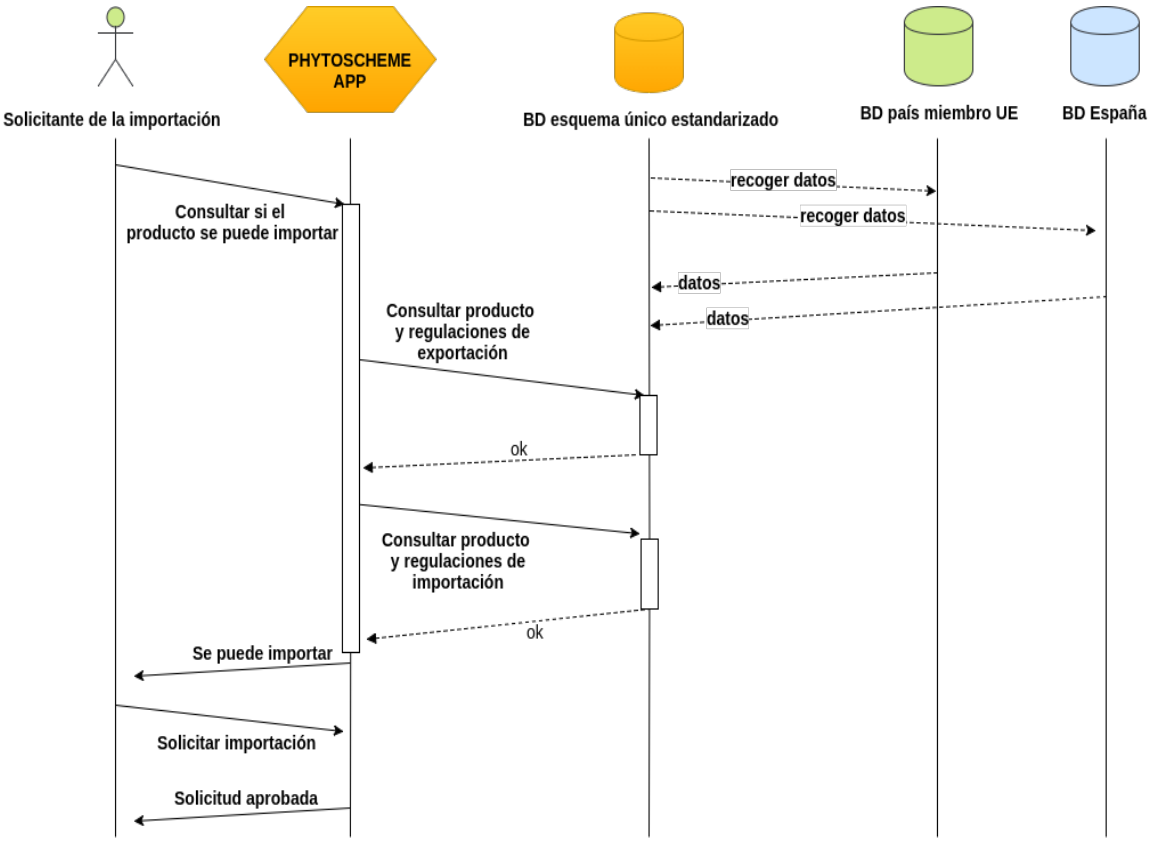

Figura 2.2: Diagrama de flujo del potencial proceso de importación de un producto fitosanitario

Se puede apreciar que con este desarrollo se conseguiría no solo reducir los plazos de respuesta al agricultor/empresa sino también una mayor transparencia en la consulta de los datos, puesto que al estar integrados en un modelo único, el agricultor/empresa o los interesados podrían tener acceso a toda la información y consultar cualquier aspecto que de otra manera habría sido prácticamente inaccesible. No solo eso, sino que además, al conseguir un modelo potencialmente estandarizado, no haría falta del trabajo manual y tedioso de comprobación de los requisitos por parte de la Subdirección General de Higiene Forestal y Vegetal sino que sería el propio sistema el encargado de comprobar si la petición del solicitante cumple con la legislación fitosanitaria vigentede cada país.

Uno de los aspectos a tener en cuenta al comienzo del proyecto fue el hecho de que

para que la solución se pudiera aplicar a un nivel real del problema, se necesitaba una gran capacidad de almacenamiento. Los datos deberían almacenarse periódicamente y además, potencialmente podrían provenir de multitud de fuentes, en diferentes formatos, con unos tamaños variables y constantes actualizaciones. Muchas fuentes equivalen a muchos datos, y muchos datos equivalen a la necesidad de un sistema de almacenamiento capaz de soportar toda esa carga.

Unido a esto, la elección de dicho sistema de almacenamiento suponía un punto crítico, puesto que no bastaría con cualquier base de datos sino que tendría que cumplir ciertas características como la posibilidad de guardar los datos en sus formatos originales, la necesidad de generar una jerarquía para su almacenamiento, o la posibilidad de integración con los trabajos de transformación y procesado de los datos.

Por ello, el proyecto tuvo como objetivo principal desarrollar una solución prototipo para demostrar la validez de los procesos y herramientas involucradas en la integración de varias fuentes heterogéneas de productos fitosanitarios en un modelo único.

2.3. Restricciones

Teniendo en cuenta que el proyecto actual es un TFG y no un proyecto comercial, se tienen que tomar en cálculo una serie de restricciones que vienen dadas tanto por la naturaleza del proyecto como por los partícipes del mismo. En primer lugar, siendo un TFG, debe realizarse una cantidad de esfuerzo equivalente a 12 ECTS.. Dado que el desarrollo realizado en este proyecto está enfocado a formar parte de un plan mucho mayor, donde el trabajo realizado se retomará y ampliará en futuros TFG, desde el principio se delimitaron aquellas características que se deseaban incluir en el proyecto y también aquellas que no corresponden a esta iteración. Siendo una fase temprana de dicho plan maestro, a esta fase le correspondía la parte de validación del modelo, de las tecnologías empleadas y de la viabilidad del producto, no siendo primordial tener un producto final robusto sino demostrar que las tecnologías en su totalidad se complementaban y funcionaban acorde a lo esperado.

Otro factor restrictivo dada la naturaleza del proyecto es el hecho de que existe un director, que puede marcar las pautas de desarrollo, sugerir e imponer metodologías de trabajo o incluso herramientas del stack tecnológico que pueden suponer una ventaja o una desventaja en el desarrollo del proyecto. El alumno debe ser capaz de discutir estas tendencias con el director y razonar las decisiones que se tomen, en conjunto, y exponiendo sus puntos de vista con el objetivo de llegar a un acuerdo común.

Capítulo 3

Análisis

3.1. Análisis del problema

Como ya se ha aclarado varias veces a lo largo de este documento, actualmente hay una necesidad real en el mundo agrícola; la de disponer de la información sobre productos fitosanitarios de una manera centralizada, actualizada y de fácil acceso. Hoy en día esta necesidad se ha intentado abordar de varias maneras, y por ello hay disponibles varias aplicaciones que intentan apoyar al consumidor en las tareas agrícolas que involucran productos fitosanitarios. Algunos ejemplos son los productos que ofrecen empresas como aGROSLab, Cuaderno de Campo Agronev, o Agricolum. No obstante, estas aplicaciones se enfrentan al mismo problema; la inexistencia de una base de datos estandarizada cuya información sea congruente a través de los diferentes países de la Unión Europea. Por eso mismo, estas aplicaciones tienen que implementar, desarrollar y mantener ellas mismas las bases de datos que permitan acceder a la información deseada. Nuestra solución ofrecería un sistema dotado de una base de datos estandarizada, congruente en su información y completa, de código abierto y altamente escalable, por lo tanto todas las aplicaciones mencionadas arriba podrían convertirse en potenciales clientes consumidores de nuestro sistema, sustituyendo sus bases de datos por nuestra solución, con costes de integración mínimos.

3.2. Análisis del marco conceptual

En este apartado se van a presentar los diferentes factores que van a intermediar en el proyecto: los proveedores de los datos, los recursos de los que se dispone, la propuesta de valor, los diferentes clientes del proyecto y los costes del mismo.

- **Proveedores:** Terceros de los que el proyecto se nutre para obtener los datos a emplear en la aplicación:

Cada fuente de datos proviene de algún portal web (nacional o internacional). En el proyecto se monta una infraestructura alrededor de dichos portales y por lo tanto la desaparición de alguno de estos sitios web significaría el cese de aprovisionamiento de los datos provenientes de dicha fuente. Actualmente, el proyecto se nutre de dos fuentes distintas: los productos autorizados recogidos del portal del *Mapama* [1] y los datos provenientes de la *base de datos de pesticidas a nivel Europeo* [7].

- **Recursos:** Elementos con los que el alumno ha contado para desarrollar el proyecto.

  - Equipo de trabajo (portátil propio)
  - Laboratorio de investigación - provisto por el director del proyecto, ubicado en el laboratorio L2.09 del edificio Ada Byron, en la Escuela de Ingeniería y Arquitectura de Zaragoza.
  - Herramientas open source - Todo el stack tecnológico, debido a un presupuesto nulo ha tenido que ser gratuito.
  - 300 horas contables de trabajo, las recomendadas para proyectos de este tipo.

- **Propuesta de valor:** Beneficios que aporta el proyecto respecto a la situación actual:

  - Diseño de una solución de integración que recoge datos sobre productos fitosanitarios de fuentes heterogéneas y los integra en un esquema único - que pretende solucionar algunos de los problemas actuales.
  - Desarrollo de una aplicación que muestre los resultados de manera gráfica - para demostrar los resultados tanto al tribunal como a los potenciales clientes.
  - Redacción de un documento explicativo del proceso anterior - la memoria, que servirá para que el tribunal comprenda la dedicación y el esfuerzo invertido en el proyecto y los avances y logros obtenidos.

- **Segmentos de clientes:** Los diferentes tipos de clientes objetivos a los que está dirigida la aplicación final:

  Por una parte están los agricultores que comercializan sus productos, las instituciones encargadas de validar y certificar la importación / exportación de productos agrícolas o las empresas importadoras / exportadoras de productos

agrícolas. Este segmento se beneficiaría diréctamente de los resultados del proyecto puesto que el proceso de comprobación de los requisitos fitosanitarios sobre los productos agrícolas / pesqueros / alimentarios se conseguiría de manera mucho más sencilla. Por otro lado también se podrían beneficiar de la infraestructura conseguida en este proyecto empresas que ya implementan sus propias aplicaciones fitosanitarias, pero carezcan de ese esquema y datos estandarizados que este proyecto propone.

– **Costes del proyecto:** Costes económicos o temporales que el alumno ha invertido en el proyecto:

  - Primera matrícula del proyecto: 259,92
  - Segunda matrícula del proyecto: 408,96
  - Transporte mediante bus: Bono bus 90 días x 3 = 104,90 x 3 = 314,70
  - 300 horas invertidas en el trabajo
    - TOTAL: 983,58 más las 300h de trabajo

3.3. Análisis de riesgos

En la fase inicial del proyecto se abordó el proceso de gestión de riesgos, para determinar los diferentes factores que podrían afectar a un proyecto de esta envergadura. Dicho proceso consta de varios pasos que en conjunto permiten tener una visión clara de aquello que puede entorpecer, frenar o incluso imposibilitar la finalización del proyecto. A continuación se listan dichos pasos y se presentan las conclusiones principales extraídas de cada una de ellas, dejando para los anexos la versión completa:

1. En primer lugar, **la identificación de riesgos** permite determinar la lista de riesgos capaces de romper la planificación del proyecto. Durante esta fase se estudió qué factores podrían influenciar, en mayor o menor medida el flujo de trabajo normal del proyecto. Se agruparon en diferentes categorías para delimitar las zonas a las que afectaría cada riesgo. Así pues, aparecen 31 riesgos divididos en 4 clases:

   a) 4 riesgos globales (referentes a todo el proyecto)

   b) 6 riesgos tecnológicos (referentes a los aspectos más técnicos y tecnológicos del proyecto)

   c) 7 riesgos de alcance (referentes al tamaño y alcance de la solución)

d) 14 riesgos de entorno de desarrollo (referentes tanto a la gestión como a las diferentes partes del entorno del desarrollador)

2. El **análisis del riesgo** ofrece una medición de la probabilidad y el impacto de cada riesgo. Maneja tres valores que determinan la gravedad de un riesgo: la probabilidad con la que se puede dar un riesgo, el impacto que tendría en el resultado final un riesgo y la aceptación del riesgo, una medida delimitadora que define aquellos riesgos que son considerados aceptables y aquellos ante los que se deben tomar medidas. En esta fase se detectaron un total de 6 riesgos reseñables, que se presentan en el siguiente punto.

3. La **priorización de riesgos**, fase donde se puede ver la lista de todos los riesgos anteriores ordenados por su gravedad. A continuación se mencionan aquellos que han tenido un factor de gravedad superior a 4, límite del criterio de aceptación. Todos los riesgos que aparecen aquí han obtenido una puntuación de 6/6:

   a) RG_1. Riesgo global "Plazos".

   b) RT_2. Riesgo de tecnologías "Software no probado".

   c) RT_6. Riesgo de tecnologías "Inalcanzable".

   d) RA_1. Riesgo de alcance "Tamaño estimado".

   e) RA_6. Riesgo de alcance "Número de cambios".

   f) RE_9. Riesgo de entorno de desarrollo "Formación".

4. Finalmente, la **planificación de la gestión de riesgos**, fase relativa al plan para tratar cada riesgo significativo. En este apartado la estrategia a seguir fue recoger los seis riesgos anteriores y proponer para cada uno una solución mitigadora. Los resultados de esta fase se pueden observar en el apartado *Análisis de riesgos* de los *Anexos*.

3.4. Análisis del contexto tecnológico

Dado que se trata de un escenario caracterizado por la heterogeneidad en contenidos y formatos de los datos, la necesidad de su procesamiento y de una escalabilidad futura, el proyecto está situado en un círculo de tecnologías Big Data, que presenta los siguientes retos:

– **Datos.** La información sobre los productos fitosanitarios no está habitualmente disponible en un formato estructurado. Es decir, la solución debe poder trabajar

con datos en formatos no estructurados y ser capaz de procesarlos, idealmente convirtiéndolos a un formato relacional.

- **Esquema.** No existe un esquema de referencia compartido para integrar la información de productos fitosanitarios de diferentes países, pudiendo darse el caso de la existencia de varios esquemas distintos en función del caso de uso. Es decir, la solución debe poder reconfigurar el esquema de integración con facilidad.

- **Procesado.** Derivado del reto anterior, los datos no se pueden procesar y almacenar directamente en el esquema de integración sino que deben guardarse en el formato original y ser procesados bajo demanda teniendo en cuenta las características específicas de cada fuente.

- **Almacenamiento.** No se dispone de un presupuesto para invertir en un gran sistema de almacenamiento que permita almacenar los datos en formato original. Por ello, toda solución deberá ser de código abierto y poder ejecutarse sobre hardware de bajo coste. Presentación de los datos. Es necesario desarrollar una interfaz visual para presentar los datos una vez integrados en un modelo único.

- **Agilidad.** La solución debe poder evolucionar con facilidad. Cualquier desarrollador que utilice la solución debe poder reconfigurar rápidamente el esquema común y los servicios que exponen dicho esquema para su uso.

- **Plazos.** Las limitaciones temporales y la priorización de tareas son influyentes en la elección del stack tecnológico, aunque en menor medida que los puntos anteriores.

Habiendo mencionado los factores anteriores, a continuación se presentan las tecnologías a las que se va a hacer referencia en el siguiente apartado con el objetivo de familiarizar al lector con las herramientas utilizadas:

- ***Hadoop.*** [11] La librería software Apache Hadoop es un framework que permite el procesamiento distribuido de múltiples conjuntos de datos a través de clusters de ordenadores mediante modelos de programación sencillos. Está diseñado para escalar desde servidores únicos hasta miles de máquinas, donde cada una ofrece computación y almacenamiento local. Más que depender del hardware para prestar una alta disponibilidad, la librería en sí está diseñada para detectar y gestionar fallos en la capa de aplicación y así permitir una alta disponibilidad a pesar de que los equipos individuales fallen.

- ***Hive.*** [12] El software de data warehouse Apache Hive facilita la lectura, escritura y gestión de grandes conjuntos de datos residentes en almacenes distribuidos de datos mediante SQL. La estructura de datos puede ser proyectada sobre los datos que ya existen en almacenamiento. Apache Hive provee una herramienta de línea de comandos y un driver JDBC para que los usuarios se puedan conectar a Hive.

- ***Sqoop.*** [13] Apache Sqoop es una herramienta diseñada para transferir bloques de datos entre Apache Hadoop y almacenes de datos estructurados como las bases de datos relacionales.

- ***JHipster.*** [14] JHipster es una plataforma de desarrollo para generar, desarrollar y lanzar aplicaciones con tecnología Spring Boot, AngularJS y Spring microservices.

- ***Spring Framework.*** [15] El framework Spring provee un modelo de programación y configuración sencillo para aplicaciones Java. Un punto clave de Spring es el soporte infraestructural a nivel de aplicación.

- ***Spring Boot.*** [16] Spring Boot es un framework ligero que tiene la intención de simplificar el proceso de configuración de una aplicación hecha con Spring.

- ***AngularJS.*** [17] AngularJS es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

- ***Talend.*** [18] Talend es un software open-source de integración, procesado y transformación de datos. Permite trabajar con paradigmas Big Data y ofrece una interfaz gráfica para diseñar y programar cómodamente procesos ETL.

3.5. Elección del Stack Tecnológico

Debido a los factores mencionados en el apartado anterior, el stack tecnológico se vio restringido a las herramientas mencionadas anteriormente; tal como se ha dicho, la solución necesitaba almacenar la información en crudo hasta el momento de su procesamiento y es por ello que una aproximación relacional habría sido inviable. Por lo tanto, viendo las diferentes opciones noSQL disponibles (MongoDB, Cassandra, DynamoDB, HBase, etc) y por recomendación del director del proyecto, la elección más clara fue elegir Hadoop como sistema de almacenamiento. Hadoop es una gran

herramienta para el escalado de aplicaciones y lo utilizan grandes empresas para Big Data. Se adapta perfectamente a las necesidades de este proyecto y el software es gratuito. Además, hay una gran comunidad de personas capaces de ayudar con cualquier proyecto y la documentación disponible es lo suficientemente extensa como para salir de cualquier situación problemática.

No obstante, Hadoop por sí mismo no era capaz de solucionar todos los problemas del proyecto; Hadoop es capaz de almacenar los datos y, si bien es cierto que las operaciones de mapreduce permiten transformar dichos datos, dada la necesidad del desarrollo ágil del proyecto, lo mejor fue disponer de algún mecanismo más sencillo para el procesado de los mismos. Por ello, se decidió emplear Hive como herramienta de traducción de los datos almacenados en Hadoop, tanto en crudo como procesados, a una estructura relacional, sobre la que se podían hacer preguntas SQL.

El otro reto planteado fue la presentación de los datos integrados mediante una interfaz web, lo que vendría siendo el lado del cliente de la solución. Indudablemente hay casi infinidad de posibilidades para el desarrollo de esta parte. No obstante, realmente la parte de visualización no fue una parte crítica del proyecto, ni el objetivo del mismo. Es por ello que se adoptó una postura de desarrollo rápido del lado del cliente. JHipster (generador de aplicaciones sobre Spring) resultó la herramienta indicada, puesto que facilmente, en unos cuantos pasos era capaz de generar una aplicación sobre un esquema de base de datos (en este caso MySQL) y con una interfaz gráfica cuidada que satisfacía las necesidades del proyecto. No obstante, para poder conectar la base de datos de Hadoop con la base de datos MySQL necesaria para JHipster, el paso intermedio fue conectar Hadoop con Hive y realizar operaciones ETL de Hive a JHipster mediante Sqoop, una herramienta de transformación y carga de datos entre distintas bases de datos.

Teniendo en cuenta que los anteriores puntos constituyen el core tecnológico del proyecto, conforme se avanzaba se veía que no eran suficientes para conseguir los resultados deseados. Se tuvieron que emplear otras herramientas adicionales para diferentes tareas como el procesado de los datos previo a su exposición en Hive pero posterior a su almacenamiento en Hadoop, o la traducción de los datos y estructuras de Hive a la base de datos que emplea JHipster. Las elegidas fueron Talend Big Data (herramienta de procesado de ficheros capaz de conectarse a Hadoop, extraer la información contenida en sus nodos, procesarla y volver a guardarla con los cambios efectuados, tal como se le indique) y Sqoop (herramienta diseñada para transferir bloques de datos entre Apache Hadoop o Hive y almacenes de datos estructurados como las bases de datos relacionales).

3.6. Captura de requisitos

En una fase temprana del proyecto se realizó un análisis amplio y genérico del problema en conjunto con el director del proyecto. Se observaron y entendieron los retos a los que se enfrenta y se definieron los objetivos que se pretendía conseguir y por lo tanto los requisitos que se debían cumplir. A continuación se recogen en una tabla los requisitos, tanto funcionales como no funcionales divididos en sistema y proyecto, siendo los de sistema los referentes al propio producto tecnológico en sí, y los de proyecto los referentes a la gestión del mismo:

Nombre	**Descripción**
RFS_1	El sistema deberá recolectar los datos oficiales tanto de productos fitosanitarios como de pesticidas.
RFS_2	El sistema deberá almacenar la última versión de los datos recolectados en el RF_1 en su formato original y además mantener todas las versiones del documento.
RFS_3	El sistema deberá monitorizar, almacenar y mostrar los procesos de recolección de los datos de entrada, así como las rutas de su procesado.
RFS_4	El sistema deberá ofrecer la infraestructura y herramientas de configuración necesarias para que futuros desarrolladores puedan integrar otras fuentes de datos de manera rápida y eficiente.
RFS_5	El sistema deberá implementar un modelo de aplicación consistente, ejemplificando un ciclo de vida típico de los datos, desde su recogida, su procesamiento, su posterior integración en un modelo más completo y su presentación en un Front-End de ejemplo.

Tabla 3.1: Requisitos Funcionales del Sistema

Nombre	Descripción
RNFS_1	Los información de los productos fitosanitarios deberá ser recogida de portales como mapama y los datos sobre los pesticidas de la base de datos europea.
RNFS_2	Se hará uso de algún tipo de crawler web para la descarga periódica de los datos sobre fitos y pesticidas.
RNFS_3	Como sistema de almacenamiento de los datos recogidos sobre fitos y pesticidas se usará Hadoop.
RNFS_4	Para monitorizar, almacenar y mostrar los procesos de recolección de los datos de entrada se usará Talend Big Data.
RNFS_5	Para conseguir unos desarrollos posteriores más ágiles se hará uso de la herramienta HIVE, que permite una aproximación relacional directamente sobre Hadoop.
RNFS_6	La presentación de los datos en su formato final se hará mediante una aplicación con GUI, desarrollada mediante la herramienta JHipster.

Tabla 3.2: Requisitos No Funcionales del Sistema

Nombre	Descripción
RFP_1	El proyecto deberá incluir una memoria en la que se documentan todos los pasos y procesos involucrados en su construcción.
RFP_2	Se deberá mantener constancia del esfuerzo dedicado durante el proyecto.
RFP_3	El proyecto deberá mantener un control de versiones actualizado en todo momento.

Tabla 3.3: Requisitos Funcionales del Proyecto

Capítulo 4

Diseño

4.1. Diseño conceptual

Explicar como se ha llegado a la solución y las partes que lo integran a un nivel abstracto, sin entrar en nombres de las tecnologías ni herramientas ni programas ("los datos se descargan periódicamente desde la web, se almacenan en formato original luego se transforman y se presentan en la aplicación..."

4.2. Arquitectura final del sistema

Diagramas arquitecturales, explicar los diferentes componentes y la manera en la que interaccionan y se comunican desde el momento de la recolección de los datos hasta su presentación en la aplicación final. Incluir aparte del diagrama de herramientas algun diagrama de componente y conector, e incluso algún trocito con un diagrama de flujo y a ser posible un diagrama de clases de JHipster y la manera en la que interaccionan con el sistema.

Capítulo 5

Implementación

5.1. Prueba de concepto

En la fase inicial del proyecto lo primero que se hizo fue demostrar si las herramientas elegidas en el stack tecnológico son viables, si funcionan en conjunto, determinar los problemas que presentan y los retos a los que se enfrenta desde una aproximación tecnológica. Para ello se instalaron independientemente todas las herramientas, empezando por *Hadoop*.

Hadoop:

Inicialmente la instalación de *Hadoop* supuso algunos problemas puesto que el alumno no había tenido contacto con la herramienta previamente, y la *información* [19] que el alumno seleccionó como base para la instalación estaba desafortunadamente incorrecta (la instalación disponible en dicha web había sido probada con Ubuntu Linux 10.04, pero no con la versión del alumno, la 16.04). Por lo tanto, se tuvo que empezar de cero, eliminando cualquier rastro de la primera instalación de *Hadoop* del sistema. Después de ello, en un segundo intento, y gracias al *tutorial de instalación de Hadoop de Digital Ocean* [20] el programa funcionó correctamente y se procedió a instalar el siguiente bloque software necesario para el funcionamiento del sistema: *Hive*.

Hive:

Para la instalación de *Hive* ocurrió un problema similar al de *Hadoop*. La fuente elegida para su instalación no fue la adecuada en un principio; el alumno eligió el tutorial expuesto en la web *Tutorial's Point* [21], que provee información no solo excesiva sino en ocasiones confusa. Como en el caso anterior, se tuvo que erradicar *Hive* del sistema para proceder con una instalación más limpia, esta vez desde la *página web oficial de Hive* [22], puesto que lo único requerido para su instalación fue su descarga y la declaración de las variables de entorno necesarias para su ejecución. De esta manera, se consiguió instalar la versión 2.2.1 de *Hive* sin dificultades.

JHipster: Instalación

Lo siguiente que se probó fue a instalar *JHipster*. Como *Java* y *Node.js* [23], dos de los componentes necesarios para su instalación ya estaban configurados en el equipo, lo único que se tuvo que configurar fue *Yarn*, que se hizo siguiendo los pasos recomendados para *Linux* en la *página oficial de Yarn* [24] para poder instalar *JHipster* con el comando *yarn global add generator-jhipster*, tal como indica la página oficial de *JHipster*. Esto en sí fue facil, y no supuso mayor problema; No obstante, el desconocimiento de *Yarn* junto con las actualizaciones periódicas que se introducían en *JHipster* hicieron que más de una vez se tuviese que borrar *JHipster* del equipo y volver a instalarlo en su última versión.

JHipster: Aplicación de prueba

Con *JHipster* instalado y configurado en el sistema, el siguiente paso obvio fue crear una aplicación y verla en funcionamiento. Para ello se siguió el tutorial del que se provee en la *página oficial* [14]. La creación de la aplicación con JHipster resultó bastante sencillo puesto que se trata de pasos secuenciales. No obstante, dado que en la web no existe mucha información acerca de la integración de un proyecto *JHipster* con *IntelliJ* [25] (entorno de desarrollo usado por el alumno), el arranque resultó bastante frustrante. El poco dominio que el alumno tenía tanto de *Gradle* como del propio entorno supuso un reto en las fases tempranas del proyecto, que se superó a base de leer documentación y realizar diversos intentos hasta que por fin se consiguió una versión de la aplicación corriendo en local, en el puerto 8080.

Integración Hadoop - Hive - JHipster

Si bien es cierto que la integración entre *Hive* y *Hadoop* resulta casi trivial, la integración entre *Hive* y *JHipster* es todo lo contrario. En primer lugar, se quería conectar *Hive* con *Hadoop* para disponer de una ayuda relacional para poder hacer consultas sobre datos en formatos no relacionales. Su configuración se realizó modificando los ficheros de configuración dentro del directorio de instalación de *Hive*, para permitir una conexión entre los servicios de *Hive* y los nodos de *Hadoop*. En segundo lugar, la conexión entre *Hive* y *JHipster* se quería realizar para tener un flujo directo entre la información que se muestra en pantalla desde *JHipster* y los datos que son importados en *Hadoop* desde las diferentes fuentes. Para ello hay que tener en cuenta que cuando se crea una aplicación con *JHipster*, este pregunta por la base de datos que se quiera usar tanto en producción como en desarrollo. Actualmente *JHipster* ofrece soporte para las siguientes bases de datos: *MongoDB*, *Apache Cassandra*, o una base de datos SQL (*H2 Database Engine*, *Microsoft SQL Server*, *MariaDB*, *PostgreSQL*, *Microsoft SQL Server*, *Oracle Database*). Como se puede observar,

JHipster no ofrece soporte oficial para una base de datos correspondiente ni a *Hive*, ni a *Hadoop*. Por eso, inicialmente la aplicación de *JHipster* se creó con una base de datos relacional *MySQL* puesto que su sintáxis es la más parecida a la de *Hive* (aunque no es igual). El objetivo en este caso fue conectar *JHipster* con *Hive* diréctamente, sustituyendo de alguna manera la base de datos *MySQL* y cambiando cualquier interacción que se tuviera con ella. No obstante, las tablas que se crean durante la creación de la aplicación se crean con una sintáxis propia de *MySQL*, en la base de datos *MySQL* y con un esquema a priori no visible. Tras muchas horas invertidas, muchos portales consultados, muchas preguntas en diversos foros de Internet, esta tarea se marcó como inalcanzable y se procedió a buscar otras soluciones con una viabilidad más alta.

Sqoop

Visto el resultado de la prueba anterior y por lo tanto el abandono de ese camino, el paso más lógico que se debía tomar a continuación era añadir un componente intermedio capaz de transferir datos de *Hadoop/Hive* a *MySQL*. Afortunadamente, ese componente existe y se trata de la herramienta *Sqoop*, que permite transferir datos de una tabla de *Hive* a otra tabla de una base de datos relacional, con un formato parecido o igual a la de *Hive*. Así pues, gracias a *Sqoop* conseguíamos disponer del mecanismo mediante el que los datos importados en *Hadoop* podían ser visualizados casi diréctamente (con su previa carga en *Hive*) en el *Front-End* provisto por *JHipster*.

Estado de la prueba de concepto

Con *Hive* conectado a *Hadoop*, *Sqoop* en marcha y una aplicación *JHipster* de prueba para ver un primer resultado de la integración de los datos, el sistema funcionaba acorde a las espectativas del *workflow* de la información: Almacen de los datos en *Hadoop* tanto en su versión "en crudo" como en una versión procesada, recuperación de los datos procesados desde *Hive*, transferencia de los mismos a la base de datos *MySQL* mediante *Sqoop* y visualización por pantalla mediante la aplicación de *JHipster*. No obstante, analizando el estado de la prueba de concepto, se podía observar que en el anterior proceso faltaban tres cosas:

- **Automatización** de las tareas involucradas: Hasta este punto, cualquier parte del proceso requería de la intervención manual de un usuario, esto es, descarga de los datos desde sus fuentes, procesado de los mismos, carga de la información en *Hadoop*, creación de una tabla en *Hive* correspondiente a los datos de dicha fuente particular, carga de los datos desde *Hadoop* a *Hive* y la transferencia de los mismos mediante *Sqoop* hacia la base de datos de *JHipster*, *MySQL*. Viene

siendo evidente la necesidad de un mecanismo que permita la automatización de todas estas tareas, con una intervención mínima por parte de un usuario. Esto permitiría aparte de un incremento considerable en el tiempo de resolución del *workflow*, un desarrollo futuro más ágil y sencillo.

- **Procesado de los datos** de manera eficaz: El *TFG* requería de un módulo de procesado de datos puesto que, como ya se ha explicado en ocasiones durante esta memoria, los datos pueden provenir de diferentes fuentes en formatos heterogéneos. Así pues, una carencia en este punto era esa herramienta o módulo que permitiera trabajar con datos en distintos formatos de una manera rapida y eficaz. Previamente los datos se habían "procesado" manualmente, con un sencillo editor de texto.

- **Actualización periódica** de la ejecución de todas las tareas: Teniendo en mente una visión futura y acabada del proyecto, otro aspecto que se echaba en falta en este punto era la posibilidad de que todo el *workflow* se ejecutase de manera periódica, obteniendo con esto una gran ventaja: la de proveer al consumidor de unos datos actualizados en todo momento.

Kettle

Dejándo de lado la *automatización de las tareas involucradas* y *la actualización periódica de la ejecución de todas las tareas*, lo siguiente que se abordó durante la prueba de concepto fue el problema del *procesado de los datos*. Para ello se requería del uso de alguna herramienta capaz de conectarse con *Hive* o con *Hadoop*, cuya especialidad sean las operaciones *ETL*. El director del proyecto impuso para esto la herramienta *Pentaho - Kettle* [26] dada su previa experiencia con este tipo de programas, sobretodo en el área *"GEO"*. Así pues, dentro del abanico de los diferentes productos de *Pentaho* [27] se encontraba *Pentaho Data Integration*, también conocida como *Kettle*, herramienta libre y gratuita con un diseñador gráfico para realizar operaciones *ETL* que, según mencionaba, permitía una integración sencilla con diferentes tecnologías como *Hive* y *Hadoop*, e incluso ofrecía soporte para una integración con proyectos *Java*. Según las especificaciones del producto, parecía que encajaba perfectamente con las necesidades del proyecto. No obstante, resultó en un dolor de cabeza constante desde el momento de su instalación. No solo la interfaz del programa presentaba fallos, mezclando módulos en español con módulos en inglés o duplicando algunas funcionalidades, sino que además, el intento de migrar los procesos construidos en *Pentaho Data Integration* a la aplicación de *JHipster* resultaron en muchas horas de frustración, errores y hasta largas tutorías con el director del proyecto para intentar portar el código. Tras muchos días o incluso

semanas de intentos y diversas formas de abordar el problema, lo que realmente acabó por apartar *Kettle* del *Stack Tecnológico* fue la adquisición de *Pentaho* por parte de *Hitachi* [28], privatizando el producto bajo el nombre de *Hitachi Vantara* [29] y ofreciéndo sólamente una versión de prueba del mismo.

Talend

Tras la privatización de *Kettle* y las tantas horas dedicadas a su integración dentro del proyecto de *JHipster*, se descartó *Pentaho* como parte del *Stack Tecnológico* y se empezó a buscar otras alternativas. La primera herramienta explorada fue *KNIME* [30] por recomendación de un compañero. Tras hacer algunas pruebas rápidas, se descubrió que realmente, aunque se ofertase como herramienta libre y gratuita, que lo era, algunas de sus funcionalidades eran de pago. La siguiente opción explorada fue *Talend* [18], que resultó ser la pieza clave para el funcionamiento del proyecto gracias a la sencillez de sus componentes, a la efectividad de su editor gráfico y gracias a una documentación extensa y bien organizada. A diferencia de las otras opciones para el procesado de los ficheros, con *Talend* se consiguió realizar una demostración de su funcionamiento mediante un sencillo proceso integrado en un proyecto *Java* nuevo, totalmente funcional. Ese proyecto posteriormente se empaquetó en un *.jar* y se exportó al proyecto de *JHipster* desde el que se pudo ejecutar con éxito, sin ningún problema de compatibilidad con el código ya existente.

Conclusiones

Una vez conseguidos los pilares fundamentales de la integración dentro del proyecto (*Hadoop*, *Hive*, *JHipster*, *Sqoop*, *Talend*) realmente las únicas preocupaciones que quedaban eran la automatización íntegra del proceso y la ejecución periódica del mismo para disponer de los datos en su versión actualizada. No obstante, para estas tareas no fue necesaria una prueba de concepto puesto que todo esto se podía conseguir desde el propio proyecto de *JHipster*, mediante *Spring* y código *Java*, cosas con las que el alumno ya estaba familiarizado. Dándo por finalizada la prueba de concepto, se empezó a diseñar y construir el prototipo real que quedaría como solución real del proyecto.

5.2. Prototipo real

Primera iteración para conseguir una integración y automatización completa - Productos autorizados de España

Lo primero que se hizo entrando en el desarrollo del prototipo real fue implementar un simple proceso mediante la interfaz gráfica de Talend. Este proceso realiza las siguientes operaciones:

1. Descarga desde la web del *Mapama* el fichero excel de los productos fitosanitarios autorizados.

2. Convierte dicho excel a un formato openoffice para poder ser procesado desde Talend con los componentes excel correspondientes.

3. Sube a Hadoop una versión sin procesar del fichero

4. Procesa el fichero añadiendole una columna llamada ID al principio y lo sube como versión procesada a Hadoop.

A continuación se exportó el proceso desde Talend: *Archivo* → *Export* → *Java* → *JAR file*. Esto exporta las clases y librerias que Talend necesita para lanzar el job en un archivo comprimido llamado ¡nombre_job¿.jar El siguiente paso fue descomprimir el JAR en cuestión, analizar su contenido y ver cómo se podría importar en un proyecto Java. El JAR contenía varias carpetas y ficheros pero lo que interesa es lo siguiente:

```
Nombre_del_jar
├── lib
│   └── librerias jar
├── Nombre_del_proyecto
│   └── Nombre_del_job
│       └── clase java principal del job
└── routines
    ├── system
    │   ├── api
    │   │   └── clases java
    │   ├── xml
    │   │   └── sax
    │   │       └── clases java
    │   └── clases java
    └── clases java
```

Así pues, a continuación se creó un nuevo proyecto Java con IntelliJ y Maven (TalendCrawler) y se copiaron todas las clases Java con su correspondiente estructura de carpetas. Dentro del fichero pom.xml del proyecto TalendCrawler donde se importaron todas las dependencias de Talend que figuraban como librerías locales en la carpeta lib. Para ello se tuvo que definir el *repositorio de Cloudera* [31], que es desde donde Maven buscaría la mayoría de librerías. Tras comprobar que la aplicación arrancaba y se comportaba correctamente, el próximo paso fue encapsular y exportar la aplicación como un Jar, en conjunto con sus librerías. Para ello se hizo uso del plugin *one-jar* de Maven que recoge las dependencias del proyecto y las empaqueta junto a las otras clases en un único jar.

En el proyecto de JHipster lo que se hizo fue crear una clase llamada Talend, desde la que periódicamente (mediante @Scheduled) se ejecutaba el Jar anterior a través del comando Runnable.

Teniendo ya el proceso de *Talend* integrado en la aplicación de *JHipster*, el siguiente problema a abordar fue el de la automatización de su ejecución. Se sabe que los productos fitosanitarios autorizados son actualizados periódicamente en la web de *Mapama*. Por eso mismo, nuestra aplicación requería también de una descarga periódica de dichos datos, para asegurarse de que en todo momento el programa tiene la versión actualizada de los fitosanitarios autorizados de España. Esto se consiguió gracias al *módulo de scheduling*[32] de *Spring* que permite programar la ejecución de un método de manera periódica. Como decisión estratégica se propuso lanzar el proceso de *Talend* cada media hora. Resuelto este problema también, el siguiente objetivo fue automatizar toda la ejecución del proceso, desde la descarga del fichero de los productos autorizados hasta la visualización de los datos mediante *JHipster*. Aprovechandose del mismo módulo anterior de scheduling, el desarrollo tendría que seguir el siguiente esquema:

- Primero, los datos deberían descargarse y procesarse y almacenarse en *Hadoop* mediante el módulo de *Talend*.

- A continuación, se debería implementar otro módulo encargado de la carga de dichos datos procesados a una tabla de *Hive*.

- Después de eso, se deberían transferir los datos de *Hive* a la base de datos *MySQL* que emplea *JHipster*.

Así pues, para cada uno de los módulos mencionados se creó un paquete con una clase que contenía los métodos necesarios para lograr sus tareas particulares. A continuación se adjunta un diagrama de clases para ilustrar de una mejor forma la infraestructura que se construyó para soportar el comportamiento mencionado en los puntos anteriores.

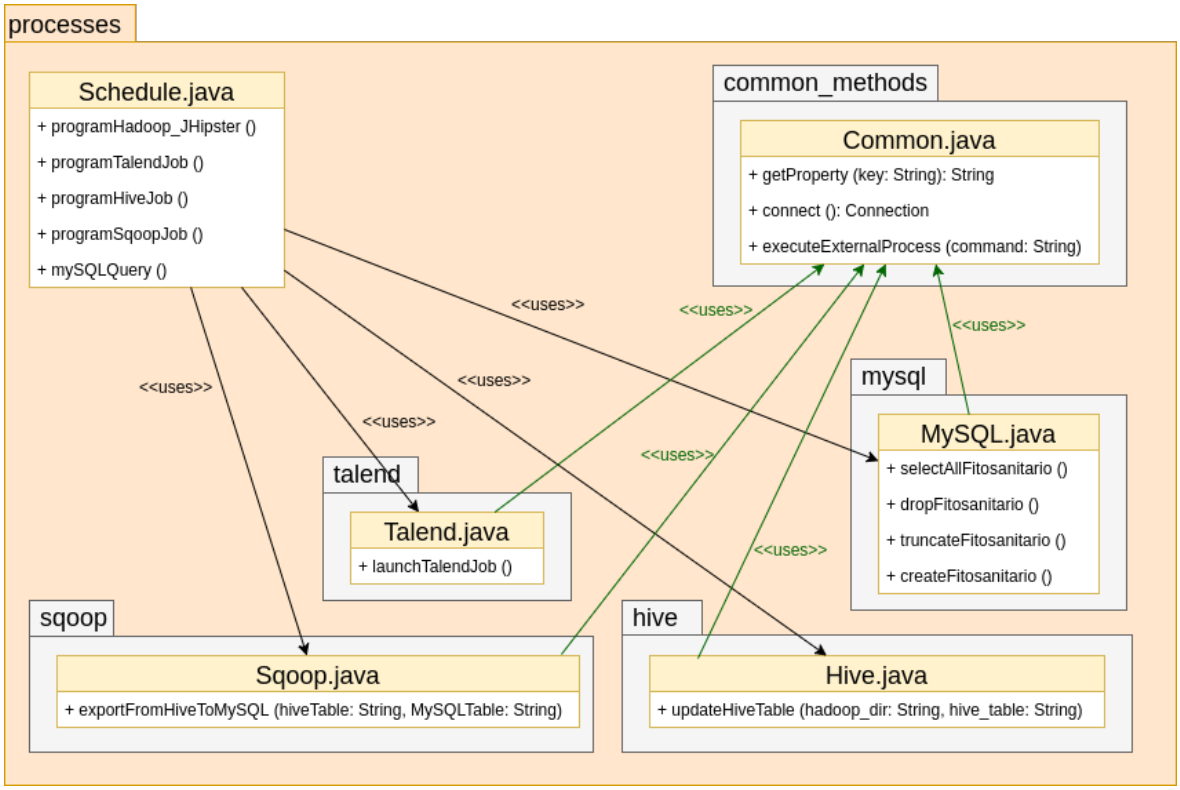

Figura 5.1: Diagrama de clases y paquetes para soportar la automatización del *workflow*

Una vez vista la estructura del diagrama anterior, a continuación se presenta un diagrama de secuencia para ilustrar la interacción de los diferentes componentes y el rol que juegan en el *workflow* desde que los datos se descargan hasta que pasan a visualizarse mediante *JHipster*.

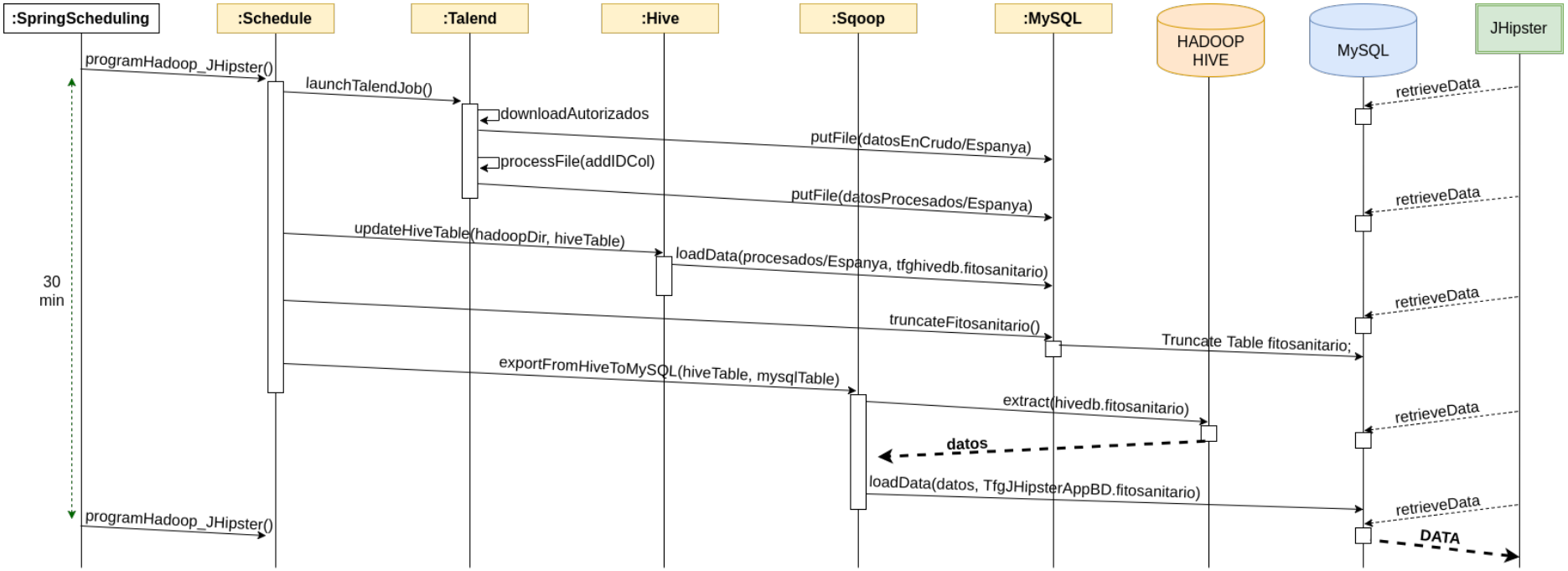

Figura 5.2: Diagrama de secuencia del *workflow* implementado

Segunda iteración para conseguir una integración y automatización completa - Sustancias activas de Europa

La primera iteración supuso los mayores problemas debido no solo al desconocimiento previo de las tecnologías sino también al hecho de no saber exactamente si dichas tecnologías iban a funcionar en conjunto. Una vez conocidas las tecnologías y tomado un primer contacto con ellas (el alumno no había trabajado con *Talend* previamente) la segunda parte de la integración se llevó a cabo de una manera mucho más fluída. Para esta iteración se conocía previamente el *modus operandi* para automatizar todo el proceso, desde la descarga de los datos hasta su visualización con *JHipster*. Por lo tanto, lo único diferente con respecto a la primera iteración fue desarrollar el trabajo de procesado específico de los datos de entrada.

Para la segunda iteración se eligieron los datos expuestos en la *Base de datos europea sobre pesticidas* [33] para seguir expandiendo la solución. Como ya se ha explicado, el objetivo de este proyecto es conseguir validar un modelo de integración para datos sobre productos fitosanitarios. En la primera iteración se obtuvieron los datos sobre los productos fitosanitarios autorizados en España. Estos contenían un campo llamado *Formulado*. Dicho campo se refiere a la *sustancia activa* de cada producto. Resulta que los datos descargados de la *base de datos europea* contienen una amplia estructura de datos e información relativa a los productos fitosanitarios. No obstante, dicha cantidad de información también resulta excesiva. Por ello, se ha optado por una aproximación minuciosa, cogiendo y procesando un solo elemento de todos los disponibles a la vez. En este caso dicho elemento corresponde a un fichero con la información relativa a las *sustancias activas*. Esta aproximación permitire ese objetivo de integración puesto que gracias a ello se puedo hacer un mapeo casi directo con los datos sobre productos autorizados de España.

De igual manera que en la primera iteración, se implementó en *Talend* el workflow necesario para procesar los datos de las sustancias activas. Esto es, por una parte, descargarlos de la página web, añadir la fecha y hora del momento de la descarga y guardarlos en *Hadoop* como datos en crudo de España sin alterar ni su formato ni su contenido. Por otra parte se formateó el contenido, para almacenar en *Hadoop* un fichero *.csv* con sólamente la información relevante del fichero original y con una columna extra para el identificador de las filas. El mismo proceso de Talend también se encarga de subir este *.csv* a Hadoop en la carpeta de datos procesados de Europa.

A continuación se preparó la infraestructura necesaria para soportar la carga de datos en *Hive* mediante una nueva tabla que se mantendrá actualizada con los datos más recientes sobre sustancias activas de Europa. Esto se consiguió gracias al desarrollo

implementado en el proyecto de *JHipster* desde el que periódicamente se lanza el workflow anterior de Talend, y posteriormente se realiza una importación de los datos a *Hive*. Además, en el lado del cliente, en *JHipster* se creó la tabla correspondiente a la d *Hive* en *MySQL* y, una vez más, periódicamente, los datos de *Hive* son transferidos a la base de datos *MySQL* a través de *Sqoop*. El resultado de esta iteración es que periódicamente, en *JHipster* se pueden visualizar los datos actualizados de las sustancias activas europeas sin necesidad de que el usuario tenga que intervenir o interactuar con el sistema en ningún momento.

Tercera iteración para conseguir una integración y automatización completa - unión de los datos anteriores en una nueva tabla - Fitosanitario_Sustancia_Activa_Europa

Mientras que las dos primeras iteraciones se centraron en recoger datos periódicamente de fuentes independientes, subirlas a *Hadoop* y luego importarlas en *Hive* y *MySQL* para ser consumidas por *JHipster*, la tercera iteración tuvo que ver con la integración de dichas fuentes independientes dentro del sistema. Como se ha mencionado anteriormente, los datos de las sustancias activas europeas se eligieron como fuente para este proyecto dado que encajaban en cierta medida con los datos de los productos fitosanitarios autorizados en España: Estos ultimos contienen un campo referente a las sustancias activas involucradas en el producto autorizado y gracias a eso se pudo hacer un *mapping* entre ellos. No obstante, el *mapping* no fue directo, puesto que los datos no venían en el mismo formato: en el caso de los productos autorizados, el campo en cuestión contenía además de los nombres de las sustancias activas en mayúscula la cantidad en la que podian estar presentes, mientras que en el caso de las sustancias activas europeas, los nombres venían en minúscula y sin la cantidad correspondiente. Así pues, en una primera aproximación lo que se hizo fue crear una tabla que contuviera los datos de los productos autorizados de España más una columna que fuera el identificador real de la **primera** sustancia activa involucrada en el producto.

Esta aproximación no es la solución perfecta, no obstante, es una primera iteración que soluciona una parte del problema. Se consiguió gracias a una consulta en *Hive* que partía los datos del campo "formulado"(referente a las sustancias activas que forman el producto) de los productos autorizados de España, se quedaba con la primera cadena de sólamente literales y hacía el *JOIN* con el nombre de la sustancia activa (pasado a mayúsculas) de la tabla de las sustancias activas europeas.

Como primera solución provisional, se consigue hacer un *matching* exitoso de unos cuatrocientos registros de un total de aproximadamente mil trescientas sustancias

activas. Los problemas que presenta son los siguientes:

- Hay productos autorizados que tienen mas de una sustancia activa como parte de su formulado y la consulta solo reconoce la primera de ellas.
- Hay sustancias activas que aparecen en los productos autorizados de España que vienen en español y la consulta no es capaz de reconocerlos puesto que las sustancias activas de europa tienen su nomenclatura en inglés.

Fichero de configuración

Para simplificar el acceso a los recursos se ha hecho uso de un fichero de configuración a los que acceden varios componentes: En primer lugar, el script bash que descarga los datos de los productos autorizados del *Mapama* [1]. Este Script usa una función bash para solicitar los valores del fichero de propiedades de la web del *Mapama*, y saber la ruta en el sistema donde guardar dicho fichero. Si en cualquier momento se quiere modificar dicha localización, gracias al fichero de configuración, el único sitio que se debería modificar sería en el propio fichero.

En segundo lugar, la aplicación Java del Job de Talend también accede a dicho fichero de configuración, puesto que en él se han establecido tanto rutas de almacenamiento dentro del HDFS de Hadoop, como el nombre del nodo o del usuario. No obstante, tal como se ha comentado en el apartado anterior, esta aplicación Java ha tenido que ser empaquetada en un Jar único y conjunto con todas sus librerías. Entonces ... ¿cómo accede a dicho fichero de configuración?. La solución ha sido hacer que el Jar reciba la ruta a dicho fichero mediante un argumento, de forma robusta, tal que si no recibe argumentos, o si el fichero que se le pasa no es un fichero de propiedades, el proceso alerta del error y se detiene.

5.3. Problemas técnicos detectados

Mencionar los problemas confrontados durante la realización del proyecto y la manera en la que se han solucionado.

Capítulo 6

Gestión

6.1. Metodología

Explicar la metodología que se ha seguido en la realización del proyecto. Explicar como se ha hecho el control de esfuerzos. Dónde están las hojas, etc. Hacer un resúmen de los esfuerzos y estadísticas contabilizadas del proyecto, mencionando aquellos puntos de más interés.

6.2. Control de versiones

Explicar como se ha hecho el control de versiones (GIT y Drive)

6.3. Pautas e imposiciones

- La memoria deberá seguir las pautas indicadas en las recomendaciones de redacción de memorias técnicas de la Universidad de Zaragoza - Algunas partes del proyecto han sido restringidas por el director ... - Las horas son las que se imponen por la universidad en relaciona los 12 ECTs

6.4. Estimación del coste

¿Cuanto cuesta en dinero tu TFG?

Capítulo 7

Conclusiones

7.1. Resultados y objetivos

Se han conseguido los objetivos propuestos ? Están todos los requisitos cubiertos? Está bien documentada la solución? Es escalable ? ...

7.2. Conocimientos adquiridos

Resumir los conocimientos tanto a nivel personal como a nivel de tecnologías adquiridos.

Capítulo 8

Bibliografía

[1] Página web del ministerio de agricultura y pesca, alimentación y medio ambiente de españa. http://www.mapama.gob.es/es/agricultura/temas/sanidad-vegetal/productos-fitosanitarios/registro/menu.asp.

[2] Directiva 2009/128/ce. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:309:0071:0086:es:PDF.

[3] Cuaderno de explotación. http://www.mapama.gob.es/es/prensa/noticias/el-ministerio-de-agricultura-alimentacion-y-medio-ambiente-aprueba-un-modelo-armonizado-de-cuaderno-de-explotacion-de-los-productos-fitosanitarios/tcm7-311275-16. Online; Último acceso: 22 - October - 2017.

[4] Cuaderno de explotación agrícola campo agroslab. http://www.cuadernoexplotacion.es.

[5] Cuaderno de campo agrícola agricolum. https://agricolum.com.

[6] Cuaderno de campo agronev. http://jnevado.com/CUADERNOCAMPO/.

[7] Base de datos de pesticidas a nivel europeo. http://ec.europa.eu/food/plant/pesticides/eu-pesticides-database/public/?event=homepage&language=EN.

[8] Reglamento (ce) nº 1107/2009 del parlamento europeo y del consejo, de 21 de octubre de 2009, relativo a la comercialización de productos fitosanitarios y por el que se derogan las directivas 79/117/cee y 91/414/cee del consejo. https://www.boe.es/buscar/doc.php?id=DOUE-L-2009-82202.

[9] Formulario para solicitudes relativas al registro oficial de productos y material fitosanitario. http://www.mapama.gob.es/agricultura/pags/fitos/registro/fichas/pdf/Modelo

[10] Preguntas y respuestas frecuentes sobre el registro de productos fitosanitarios del ministerio de agricultura, alimentación y medio ambiente. http://www.mapama.gob.es/agricultura/pags/fitos/registro/fichas/pdf/FAQ.pdf.

[11] Apache™ hadoop®. http://hadoop.apache.org. Version XXXXXX.

[12] Apache hive™. https://hive.apache.org. Version XXXXXX.

[13] Apache sqoop™. http://sqoop.apache.org.

[14] Jhipster. http://www.jhipster.tech. Version XXXXXX.

[15] Spring framework. https://projects.spring.io/spring-framework/.

[16] Spring boot. https://projects.spring.io/spring-boot/.

[17] Angularjs. https://angularjs.org.

[18] Talend. https://www.talend.com.

[19] Tutorial de instalación de hadoop incorrecto para la versión ubuntu (16.04) del alumno. http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/.

[20] Tutorial de instalación de hadoop de digital ocean para ubuntu 16-04. https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-in-stand-alone-mode-on-ubuntu-16-04.

[21] Tutorial de tutorial's point para la instalación de hive. https://www.tutorialspoint.com/hive/index.htm.

[22] Página de hive - sección de instalación. https://cwiki.apache.org/confluence/display/Hive/GettingStarted.

[23] Node.js ®. https://nodejs.org/en/.

[24] Página oficial de yarn - instalación en linux ubuntu. https://yarnpkg.com/en/docs/install.

[25] Intellij idea. https://www.jetbrains.com/idea/.

[26] Pentaho data integration. http://www.pentaho.com/product/data-integration.

[27] Pentaho. http://www.pentaho.com.

[28] Hitachi. http://www.hitachi.com.

[29] Hitachi vantara. hitachivantara.

[30] Knime. https://www.knime.com.

[31] Repositorio de cloudera. https://repository.cloudera.com/content/repositories/releases/.

[32] Spring scheduling module. https://spring.io/guides/gs/scheduling-tasks/. Online; Ultimo acceso: asiqwjf.

[33] Base de datos europea sobre pesticidas. https://data.europa.eu/euodp/es/data/dataset/s8QJJ4blyMdeI2AM1TtmXA/resource/ce5c6843-eb27-4168-9c28-b0f13b4ccbb8.

[34] Centro científico tecnológico (cct) mendoza. http://www.cricyt.edu.ar/enciclopedia/terminos/ProducFito.htm.

[35] Wikipedia - fármaco. https://es.wikipedia.org/wiki/Fármaco.

[36] Glosario de términos fitosanitarios. https://www.ippc.int/static/media/files/publication/es/2016/06/ISPM_05_2016_Es_2016-06-23_FullReviewLRG-CPAM.pdf.

[37] Glosario de términos del mapama. http://www.mapama.gob.es/es/agricultura/temas/sanidad-vegetal/glosario-de-terminos-f-a-o/.

[38] Gnu general public license. https://www.gnu.org/licenses/gpl.html.

[39] Java ™. https://java.com/es/.

[40] Mysql. https://www.mysql.com. Version XXXXXX.

[41] Oracle corporation. https://www.oracle.com/index.html.

[42] Wikipedia - apache cassandra. https://en.wikipedia.org/wiki/Apache_Cassandra.

[43] Wikipedia - h2 (dbms). https://en.wikipedia.org/wiki/H2_(DBMS).

[44] Wikipedia - mariadb. https://en.wikipedia.org/wiki/MariaDB.

[45] Wikipedia - mongodb. https://en.wikipedia.org/wiki/MongoDB.

[46] Wikipedia - microsoft sql server. https://en.wikipedia.org/wiki/Microsoft_SQL_Server.

[47] Wikipedia - mysql. https://en.wikipedia.org/wiki/MySQL.

[48] Oracle database. https://en.wikipedia.org/wiki/Oracle_Database.

[49] Wikipedia - postgresql. https://es.wikipedia.org/wiki/PostgreSQL.

Glosario

Cassandra Sistema de gestión de bases de datos distribuidas NoSQL gratis y libre diseñada para gestionar grandes cantidades de datos a través de diferentes servidores. *Wikipedia - Apache Cassandra* [42].

Certificado fitosanitario Documento oficial que atestigua el estatus fitosanitario de cualquier envío sujeto a reglamentaciones fitosanitarias [FAO, 1990]. *Glosario de términos Mapama* [37].

H2 Sistema de gestión de bases de datos relacionales escrito en Java. Puede ser embebido en aplicaciones *Java* o lanzarse en modo cliente-servidor. *Wikipedia - H2 (DBMS)* [43].

Legislación fitosanitaria Leyes básicas que conceden la autoridad legal a la organización nacional de protección fitosanitaria a partir de las cuales podrán elaborarse las reglamentaciones fitosanitarias [FAO, 1990; revisado FAO, 1995]. *Glosario de términos fitosanitarios* [36].

MariaDB Fork del sistema de gestión de bases de datos relacionales *MySQL* con el objetivo de mantener una versión libre de *MySQL* dada la adquisición del mismo por *Oracle*. *Wikipedia - MariaDB* [44].

MongoDB Base de datos NoSQL gratis, libre y multiplataforma orientado a documentos (formato JSON) con un esquema. *Wikipedia - MongoDB* [45].

MSSQL Sistema de gestión de bases de datos relacional desarrollado por *Microsoft*. *Wikipedia - Microsoft SQL Server* [46].

MySQL Sistema de gestión de bases de datos relacional gratuito, libre y publicado bajo una licencia *GNU GPL* [38]. *Wikipedia - MySQL* [47].

Oracle Sistema de gestión de bases de datos relacional orientado a objetos producido y desarrollado por *Oracle Corporation* [41]. *Wikipedia - Oracle Database* [48].

PostgreSQL Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL. *Wikipedia - PostgreSQL* [49].

Producto fitosanitario De acuerdo con la Organización Mundial de la Salud (OMS), se define al producto fitosanitario como la sustancia o mezcla de sustancias destinadas a prevenir la acción de, o destruir directamente, insectos, ácaros, moluscos, roedores, hongos, malas hierbas, bacterias y otras formas de vida animal o vegetal perjudiciales para la salud pública y también para la agricultura. Inclúyase en este ítem los plaguicidas, defoliantes, desecantes y las sustancias reguladoras del crecimiento vegetal o fitorreguladores. *CCT Mendoza* [34].

Sustancia, Sustancia activa, Fármaco Un fármaco (o sustancia activa) es toda sustancia química purificada utilizada en la prevención, diagnóstico, tratamiento, mitigación y cura de una enfermedad, para evitar la aparición de un proceso fisiológico no deseado o bien para modificar condiciones fisiológicas con fines específicos. En el dominio de aplicación actual, nos referiremos en concreto a aquellos fármacos utilizados en la prevención, diagnóstico, tratamiento, mitigación y cura de enfermedades relacionadas con los productos agrícolas, marinos o alimenticios. *Wikipedia - Fármaco* [35].

Tratamiento fitosanitario Procedimiento oficial para matar, inactivar o eliminar plagas o para esterilizarlas o desvitalizarlas [FAO 1990; revisado FAO, 1995; NIMF 15, 2002; NIMF 18, 2003; CIMF, 2005]. *Glosario de términos fitosanitarios* [36].

Lista de Figuras

Lista de Tablas

Anexos

Anexos A

Datos

A.1. Modelo de datos

A.2. Flujo de datos

Anexos B

Clientes

B.1. Clientes potenciales

– ***aGROSLab*** [4]

- Registro de Explotaciones - Las parcelas se presentan para su selección en un innovador formato de celdas, facilitando la visualización de todas sus características (identificación, cultivo, superficie, . . .), el acceso al visor GIS, la aplicación de filtros y su selección individual o en bloque.
- Registro de Parcelas Agrícolas - permite cargar las parcelas que componen su explotación a partir de la información generada por el aplicativo de gestión de la Solicitud de Ayudas PAC, para la mayoría de las Comunidades Autónomas.
- Compras de Productos Fitosanitarios - incorpora un registro de compras de productos fitosanitarios, en el que podrá archivar todas sus facturas de compra en formato PDF y a partir del cual podrá registrar los tratamientos realizados.
- Registro de Tratamientos Fitosanitarios- filtra los productos autorizados para cada uno de sus cultivos, presenta las plagas para las que puede ser aplicado según la nomenclatura del MAGRAMA y le informa del tipo y rango de dosis que puede ser aplicada.
- Registro de Comercialización de Cosecha - facilita el registro de la comercialización de su cosecha, presentado en formato de celdas el conjunto de parcelas de su explotación con un determinado cultivo e informando gráficamente de aquellas en las que puede existir un problema con los plazos de seguridad de un tratamiento fitosanitario.
- Receta Fitosanitaria

- Visor GIS con Capas - permite al agricultor visualizar gráficamente las parcelas que componen su explotación y la información de cultivos y los tratamientos realizados. Una herramienta especialmente útil a la hora de identificar sus diferentes parcelas y tener una visión de conjunto de toda su explotación.
- Unidades Homogéneas de Cultivo
- Control de Consumos de Fitosanitarios - permite llevar el control de los productos (fitosanitarios y fertilizantes) adquiridos y los aplicados
- Importaciones y Exportaciones
- Importaciones y Exportaciones

– ***Agricolum*** [5]

- Web + APP móvil y tableta
- Validación dosis cuaderno de campo
- Gestión de personal y maquinaria
- Informes personalizados y oficiales
- Control por GPS
- Control stock
- Gestión económica
- Rendimientos por campos y cultivos
- Soporte telefónico y por internet
- Gestión del cuaderno de campo
- Aplicación conectada con los datos del Sigpac y fitosanitarios MAGRAMA
- Sincronización de la información desde cualquier dispositivo
- Vista de tiempo actual y previsión semanal
- Ver histórico de todas las tareas realizadas
- Saber en tiempo real el precio del mercado
- Exportación de la información en otros formatos
- Importación de los datos de la PAC
- Generación del cuaderno de explotación oficial

– ***Cuaderno de campo Agronev*** [6]

- Labores - Asignación de labores a parcelas, siembra, semilla tratada, aperos, imputación de costes
- Abonado - Registro de Fertilización y Abonado. Composición de los Abonos, Forma de Abonado
- Tratamientos - Tratamientos fitosanitarios en parcelas, eficacia, asesor, equipo de aplicación
- Análisis de plaguicidas - Análisis de productos fitosanitarios, boletín de análisis, residuos detectados
- Recolección - Registro de recolección y loteado. Asignación de venta directa, imputación de costes
- Otros tratamientos - Aplicación de otros tratamientos fitosanitarios (Post-cosecha, Locales, Vehículos)
- Costes - Imputación de gastos / costes a parcelas. Directos / Selectivos.
- Gestión comercial - Compras, ventas, gastos, facturación, domiciliación bancaria SEPA, libro de fitos

Anexos C

Análisis

C.1. Análisis de riesgos completos

A continuación se exponen las diferentes fases del análisis de riesgos de manera detallada:

1. **Identificación de los riesgos**

   Se han intentado considerar el máximo número de riesgos y se han clasificado en diferentes categorías:

   - **Riesgos globales**

ID	Nombre	Explicación
RG_1	Plazos	El proyecto no se finaliza para la convocatoria de junio, septiembre o diciembre.
RG_2	Fallo del equipo	El equipo principal de desarrollo falla, se pierde o estropea.
RG_3	Incorporación mercado	El alumno se incorpora al mercado laboral durante el desarrollo del proyecto, a falta de varios meses de su finalización.
RG_4	Experiencia del alumno	El alumno no dispone de los conocimientos y preparación suficiente para el desarrollo del proyecto.

Tabla C.1: Riesgos globales del proyecto

   - **Riesgos tecnológicos**

ID	Nombre	Explicación
RT_1	Tecnología nueva	Se trata de una tecnología nueva.
RT_2	Software no probado	Se debe interactuar con software que no ha sido probado.

ID	Nombre	Explicación
RT_3	Interfaz especializada	Es requerida una interfaz de usuario especializada.
RT_4	Componentes diferentes	Se necesitan componentes de programa diferentes a los hasta ahora desarrollados.
RT_5	Rendimiento	Se necesitan componentes de programa diferentes a los hasta ahora desarrollados.
RT_6	Inalcanzable	Se necesitan componentes de programa diferentes a los hasta ahora desarrollados.

Tabla C.2: Riesgos tecnológicos del proyecto

– **Riesgos de alcance**

ID	Nombre	Explicación
RA_1	Tamaño estimado	Tamaño estimado del proyecto
RA_2	Confianza en la estimación	Confianza en la estimación
RA_3	Número de elementos	Número de programas, archivos y transacciones
RA_4	Tamaño almacenamiento	Tamaño de las bases de datos involucradas
RA_5	Número de usuarios	Número de usuarios
RA_6	Número de cambios	Número de cambios en los requisitos
RA_7	Software reutilizado	Cantidad de software utilizado

Tabla C.3: Riesgos de alcance del proyecto

– **Riesgos de entorno de desarrollo**

ID	Nombre	Explicación
RE_1	Gestión proyectos	Hay herramientas de gestor de proyectos
RE_2	Gestión proceso desarrollo	Hay herramientas de gestión del proceso de desarrollo
RE_3	Análisis y diseño	Se usan métodos y herramientas específicas para el análisis y diseño
RE_4	Generadores de código	Hay generadores de código apropiados para la aplicación
RE_5	Pruebas	Hay herramientas de pruebas apropiadas
RE_6	Gestión de configuración	Hay herramientas de gestión de configuración apropiadas
RE_7	Base de datos	Se hace uso de una base de datos o repositorio centralizado
RE_8	Integración	Están todas las herramientas de desarrollo integradas

ID	Nombre	Explicación
RE_9	Formación	Se ha proporcionado formación a todos los miembros del equipo de desarrollo
RE_10	Expertos	Hay expertos a los cuales solicitar ayuda acerca de las herramientas
RE_11	Ayuda online	Hay ayuda en línea y documentación disponible
RE_12	Diseño arquitectónico	Se utiliza un método específico para el diseño arquitectónico y de datos
RE_13	Métricas de calidad	Se disponen métricas de calidad para todos los proyectos de software
RE_14	Métricas de productividad	Se disponen de métricas de productividad

Tabla C.4: Riesgos de entorno de desarrollo del proyecto

2. **Análisis del riesgo**

   Para esta fase se han empleado los tres medidores del riesgo: la probabilidad, el impacto y la aceptación:

   - **Tabla para estimar la probabilidad de un riesgo:**

Valor	Descripción
Bajo (1)	La amenaza se materializa a lo sumo una vez cada año.
Medio (2)	La amenaza se materializa a lo sumo una vez cada mes.
Alto (3)	La amenaza se materializa a lo sumo una vez cada semana.

Tabla C.5: Probabilidad de un riesgo

   - **Tabla para estimar el impacto de un riesgo:**

Valor	Descripción
Bajo (1)	El daño derivado de la materialización de la amenaza no tiene consecuencias relevantes para la consecución de los objetivos.
Medio (2)	El daño derivado de la materialización de la amenaza tiene consecuencias reseñables para la consecución de los objetivos.
Alto (3)	El daño derivado de la materialización de la amenaza tiene consecuencias graves reseñables para la consecución de los objetivos.

Tabla C.6: Impacto de un riesgo

   - **Tabla para estimar la aceptación de un riesgo:**

Valor	Descripción
Riesgo $\leq$	La organización considera el riesgo poco reseñable.
Riesgo ≥ 4	La organización considera el riesgo reseñable y debe proceder a su tratamiento.

Tabla C.7: Aceptación de un riesgo

La aceptación es una medida delimitadora que define aquellos riesgos que son considerados aceptables y aquellos ante los que se deben tomar medidas. Para esta medida se ha establecido un criterio de aceptación de 4. Cualquier riesgo cuyo valor sea menor que 4 se considera aceptable y por tanto un riesgo poco reseñable, mientras que aquellos que se encuentran por encima de 4 se consideran reseñables y se debe proceder a su tratamiento.

El cálculo de la gravedad del riesgo y su aceptación se realiza de la siguiente manera: se multiplica la probabilidad por el impacto, y si dicho valor excede el límite del criterio de aceptación, el riesgo se considera reseñable. A continuación, en base a las métricas anteriores, se especifican los riesgos de la fase 1 en las mismas categorías iniciales. Se resaltan en rojo aquellos riesgos cuya aceptación supera el 4.

- **Riesgos globales**

ID	Nombre	Probabilidad	Impacto	Riesgo
RG_1	**Plazos**	**2**	**3**	**6**
RG_2	Fallo del equipo	1	3	3
RG_3	Incorporación mercado	1	2	2
RG_4	Experiencia del alumno	2	2	4

Tabla C.8: Valoración riesgos globales del proyecto

- **Riesgos tecnológicos**

ID	Nombre	Probabilidad	Impacto	Riesgo
RT_1	Tecnología nueva	3	1	3
RT_2	**Software no probado**	**2**	**3**	**6**
RT_3	Interfaz especializada	1	1	1
RT_4	Componentes diferentes	3	1	3
RT_5	Rendimiento	2	2	4
RT_6	**Inalcanzable**	**2**	**3**	**6**

ID	Nombre	Probabilidad	Impacto	Riesgo

Tabla C.9: Valoración riesgos tecnológicos del proyecto

– **Riesgos de alcance**

ID	Nombre	Probabilidad	Impacto	Riesgo
RA_1	**Tamaño estimado**	**2**	**3**	**6**
RA_2	Confianza en la estimación	2	2	4
RA_3	**Número de elementos**	**2**	**3**	**6**
RA_4	Tamaño almacenamiento	1	3	3
RA_5	Número de usuarios	1	3	3
RA_6	**Número de cambios**	**2**	**3**	**6**
RA_7	Software reutilizado	1	1	1

Tabla C.10: Riesgos de alcance del proyecto

– **Riesgos de entorno de desarrollo**

ID	Nombre	Probabilidad	Impacto	Riesgo
RE_1	Gestión proyectos	1	1	1
RE_2	Gestión proceso desarrollo	1	1	1
RE_3	Análisis y diseño	1	2	2
RE_4	Generadores de código	1	1	1
RE_5	Pruebas	2	2	4
RE_6	Gestión de configuración	2	2	4
RE_7	Base de datos	1	3	3
RE_8	Integración	1	1	1
RE_9	**Formación**	**2**	**3**	**6**
RE_10	Expertos	2	1	2
RE_11	Ayuda online	2	2	4
RE_12	Diseño arquitectónico	2	1	2
RE_13	Métricas de calidad	3	1	3
RE_14	Métricas de productividad	3	1	3

Tabla C.11: Valoración de los riesgos de entorno de desarrollo del proyecto

3. **Priorización de riesgos**

Esta fase incluye todos los riesgos, ordenados de mayor a menor severidad. Se

resaltan en rojo los riesgos que habrá que considerar en un plan de defensa estratégico posterior:

ID	Nombre	Riesgo
RG_1	**Plazos**	**6**
RT_2	**Software no probado**	**6**
RT_6	**Inalcanzable**	**6**
RA_1	**Tamaño estimado**	**6**
RA_6	**Número de cambios**	**6**
RE_9	**Formación**	**6**
RT_5	Rendimiento	4
RA_2	Confianza en la estimación	4
RE_5	Pruebas	4
RE_11	Ayuda online	4
RG_2	Fallo del equipo	3
RT_1	Tecnología nueva	3
RT_4	Componentes diferentes	3
RA_4	Tamaño almacenamiento	3
RA_5	Número de usuarios	3
RE_7	Base de datos	3
RE_13	Métricas de calidad	3
RE_14	Métricas de productividad	3
RG_3	Incorporación mercado	2
RE_3	Análisis y diseño	2
RE_10	Expertos	2
RE_12	Diseño arquitectónico	2
RT_3	Interfaz especializada	1
RT_7	Software reutilizado	1
RE_1	Gestión proyectos	1
RE_2	Gestión proceso desarrollo	1
RE_4	Generadores de código	1
RE_8	Integración	1

Tabla C.12: Priorización de riesgos del proyecto

Como se puede apreciar, hay 6 riesgos cuyo factor de gravedad es preocupante y deben ser tratados acordemente:

a) RG_1. Riesgo global "Plazos". Tiene que ver con el hecho de no acabar el proyecto dentro de los plazos establecidos para su defensa. Hay 2 fechas recomendables para su defensa, la primera en Junio de 2017 y la segunda en Septiembre de 2017. No obstante, se dispone de otra oportunidad en Diciembre de 2017, aunque sería la menos recomendable dado que supondría el retraso de la defensa y con ello la dificultad del estudiante de realizar otras actividades mientras tanto. A partir de Diciembre, la consecuencia sería volver a matricularse en el proyecto y aportar las tasas de la matrícula por segunda vez.

b) RT_2. Riesgo de tecnologías "Software no probado". Tiene que ver con la probabilidad de usar en el proyecto software que previamente no ha sido probado y pueda fallar. Obtuvo una valoración de gravedad de 6/6 puesto que si bien es cierto que todas las tecnologías han sido probadas individualmente y se sabe que funcionan bien, el proceso en su conjunto no ha sido probado. No se sabe si es viable o no.

c) RT_6. Riesgo de tecnologías "Inalcanzable". Relacionado con el hecho de que el proyecto puede tener como conclusión un resultado negativo; Al tratarse de un trabajo de investigación, el resultado puede ser que no se ha logrado la integración deseada.

d) RA_1. Riesgo de alcance "Tamaño estimado". Tiene que ver con el hecho de que el proyecto resulte mucho más grande de lo estimado inicialmente, y por diferentes circunstancias no se llegue a finalizar.

e) RA_6. Riesgo de alcance "Número de cambios". Otro riesgo es el hecho de que los requisitos cambien constantemente, bien porque los clientes lo solicitan bien porque las propias tecnologías lo imponen.

f) RE_9. Riesgo de entorno de desarrollo "Formación". Este riesgo trata con el hecho de que el alumno disponga de la formación necesaria y suficiente para lograr los objetivos propuestos.

4. **Planificación de la gestión de riesgos**

En esta fase se recogen las conclusiones mitigadoras acerca de los riesgos "preocupantes"del proyecto, en relación a su factor de gravedad:

a) RG_1: Plazos - Como medida mitigante, el alumno deberá dedicar un horario de jornada completa a la realización del proyecto durante el verano del año 2017.

b) RT_2: Software no probado - La contrapartida y defensa de este riesgo es desarrollar o experimentar primero con una prueba de concepto para validar que las tecnologías en su conjunto funcionen correctamente.

c) RT_6: Inalcanzable - La mitigación este riesgo está ligada al anterior. Por lo tanto, primero se hará una prueba de concepto para demostrar si las tecnologías empleadas se pueden usar en conjunto.

d) RA_1: Tamaño estimado - La solución desde un principio debe definir bien el alcance y determinar aquellas cosas que formarán parte de la solución y aquellas que no lo harán.

e) RA_6: Número de cambios - El alumno y el profesor deben acordar al principio unos requisitos fijos que no podrán ser modificables, en conjunto con el hecho de definir claramente el alcance de la solución.

f) RE_9: Formación - Para mitigar este riesgo, el alumno debe estar en constante aprendizaje, utilizando los manuales y tutoriales de las diferentes herramientas de las que va a hacer uso durante el proyecto. Además, el alumno tendrá a su disposición al director del proyecto para consultar dudas y a los diferentes foros tecnológicos de Internet.

C.2. Análisis de diseños alternativos

Al igual que los requisitos, el diseño de la solución también ha sufrido constantes cambios. Inicialmente la propuesta de trazabilidad de la solución es la que se puede observar en la figura 1.

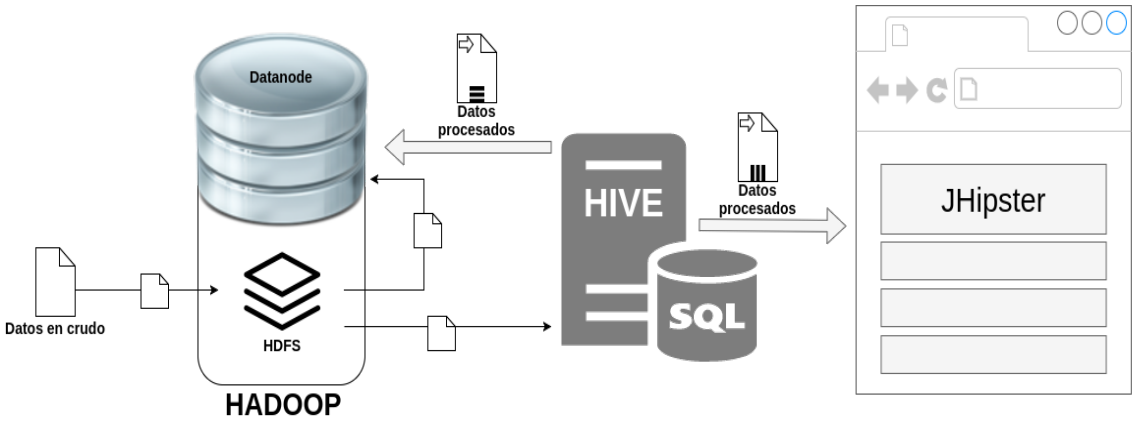

Figura C.1: Diseño primitivo del sistema.

Como se puede observar, inicialmente el concepto giraba alrededor de las tres tecnologías core: Hadoop, Hive y JHipster. No se tuvo en consideración otras

herramientas puesto que se pensaba que era suficiente para resolver el problema. Los datos en crudo, extraídos de la web del magrama o de otras fuentes heterogéneas, serían almacenados en Hadoop, en un nodo local mediante el sistema de ficheros HDFS, y posteriormente sería HIVE el encargado de procesarlos en su totalidad hasta conseguir almacenarlos en un esquema común. Además, la misma "base de datos" de Hive funcionaría como base de datos para la aplicación desarrollada con JHipster, sirviendo en todo momento ese esquema único para la visualización del mismo en un navegador web. Este diseño, conceptualmente fue la solución ideal para el problema planteado; no obstante, debido a que JHipster no ofrece soporte para cambiar la base de datos con la que se construye la aplicación y mucho menos soporte para Hive o Hadoop, tras muchos intentos frustrados de conseguir esta conectividad directa, se optó por una solución diferente, alejada de este diseño ideal pero rebelde, en contra de los paradigmas de programación que JHipster impone. La alternativa a este diseño que dió resultados y eliminó complicaciones fue la que se observa en la figura 2.

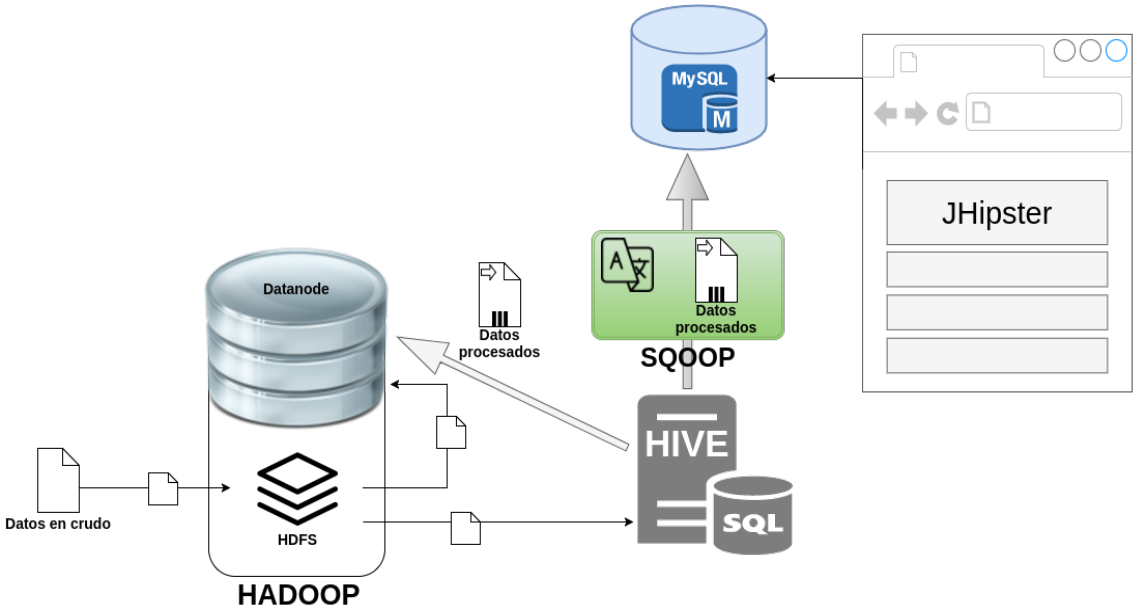

Figura C.2: Segunda iteración del diseño del sistema.

En la segunda fase del diseño, se observa la evolución de la idea, condicionada por los problemas anteriormente mencionados. En primer lugar, se conservó la base de datos nativa de JHipster, en este caso una base de datos MySQL convencional. En ella se almacenaria únicamente la estructura final del esquema unificado, con los datos finales. Dichos datos tendrían que ser pasados desde Hive mediante una herramienta de transformación y transporte de datos. En este caso se usó Sqoop, una herramienta gratuita que permite transportar los datos desde Hive a MySQL, puesto que ofrece soporte tanto para Hadoop, HDFS y Hive como para MySQL.

Una vez resuelto el problema de la visualización de los datos, lo próximo que se detectó fue esa necesidad de procesamiento de los datos en crudo antes de incluso exponerlos como un esquema relacional en Hive. Para eso, lo mejor era hacer uso de algún programa de procesado de ficheros y una de las mejores opciones aparentes era Pentaho, un programa completo de transformación de ficheros. Pentaho venía con soporte para Hadoop y HDFS, por lo que gracias a eso se pudo trabajar con ficheros directamente extraídos de Hadoop, y almacenarlos directamente en Hadoop. Los cambios se pueden observar en la figura 3.

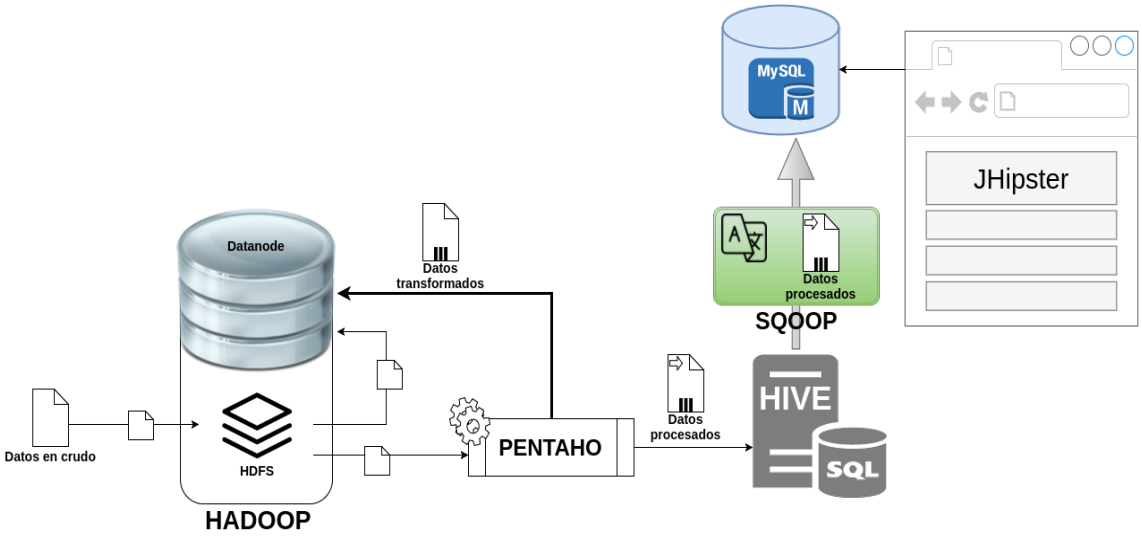

Figura C.3: Tercera iteración del diseño del sistema.

INACABADO