Universidad Zaragoza

Trabajo Fin de Grado

Sistema ETL para integración de datos de plagas agrícolas

ETL system for integration of agricultural pest data

Autor

Sergio García-Campero Hernández (721520)

Director

Javier Lacasta Miguel

Ingeniería del software
Escuela de Ingeniería y Arquitectura

25/11/2022

Sistema de información integrada de plagas

Resumen

El Ministerio de Agricultura, Ganadería y Pesca proporciona documentos PDF que describen las plagas de los cultivos, sus características y posibles tratamientos. Esta información está contenida en documentos en formato PDF. En este proyecto se parte de un trabajo previo en el cual se extrajo la información de los PDF a documentos HTML.

Con el objetivo de que estos datos sean más accesibles, este trabajo consiste en crear un sistema de extracción, procesado y carga de datos en una base de datos documental para su publicación en una aplicación web. Los documentos están repartidos en tres colecciones y contienen texto e imágenes, la dificultad reside en la extracción de la máxima cantidad de información posible, dado que los documentos son muy heterogéneos y no siguen una estructura común. Se han analizado los documentos para identificar los datos importantes y descartar aquellos que sean prescindibles.

Se ha realizado una labor de investigación de las diferentes tecnologías disponibles con el fin de elegir las más adecuadas para poder llevar a cabo este sistema de extracción de datos. El proceso de extracción se ha desarrollado en Java con Spring Batch y la web ha sido realizada con el stack MERN debido a su gran soporte y uso extendido.

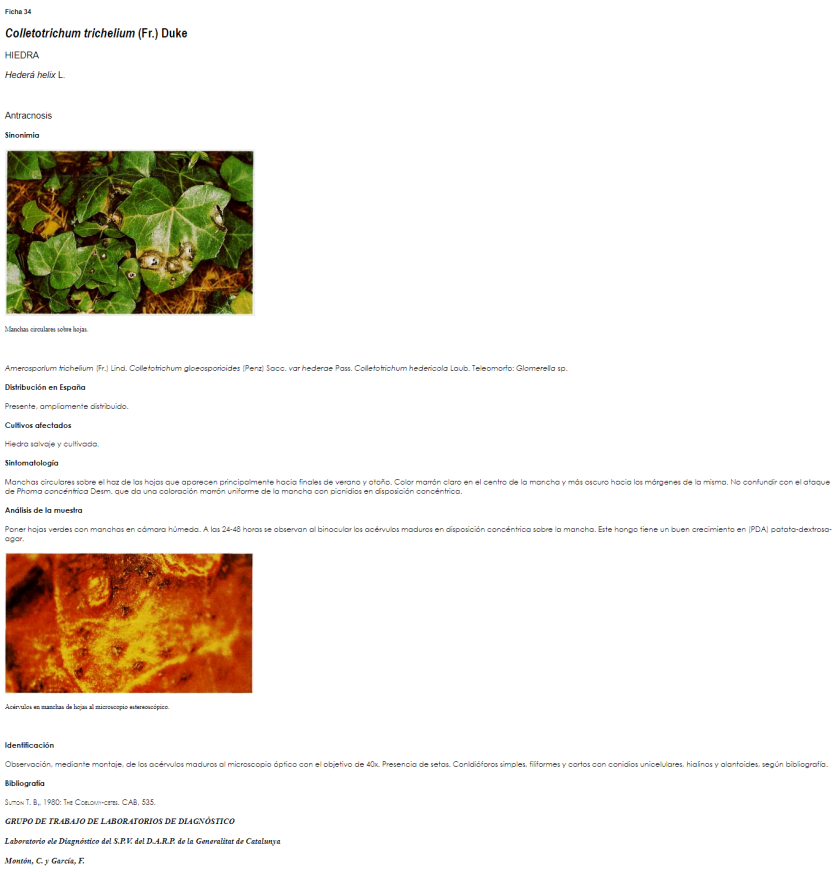
ÍNDICE

1. Introducción

1.1 **Contexto**

El Ministerio de Agricultura, Ganadería y Pesca aporta guías con información sobre las plagas en los cultivos y cómo tratarlas. Esta información se encuentra en miles de PDFs, lo que hace difícil encontrar la información deseada. Resulta muy tedioso para una persona (p.e. un agricultor) buscar manualmente, por lo que es de gran utilidad usar una herramienta que aporte una mayor accesibilidad a estas guías. Este proyecto se basa en un trabajo previo que consistía en transformar esos PDFs a documentos fácilmente procesables, generando las mismas guías en formato HTML.

Las guías están divididas en tres colecciones. Una de ellas describe las plagas [1], donde cada documento corresponde a una plaga específica. Son las guías más cortas. Otra colección describe los cultivos de España [2], en concreto aporta información de cuáles son las plagas que les afectan. Estos documentos son los más extensos ya que cuentan con anexos con información de posibles plagas que se pueden encontrar en el cultivo. Por último, la guía de productos fitosanitarios [3], cuyos documentos aportan información de cada producto y sus aplicaciones. Estos documentos son de tamaño mediano en comparación con los de las otras colecciones. A continuación se puede observar una guía describiendo una plaga en formato HTML.



Ficha 34

***Colletotrichum trichelium* (Fr.) Duke**

HIEDRA

Hederá helix L.

Antracnosis

Sinonimia

Manchas circulares sobre hojas.

Amerosporium trichellum (Fr.) Lind, *Colletotrichum gloeosporioides* (Penz) Saco. *var hederae* Pass. *Colletotrichum hedericola* Laub. Teleomorfo: *Glomerella* sp.

Distribución en España

Presente, ampliamente distribuido.

Cultivos afectados

Hiedra salvaje y cultivada.

Sintomatología

Manchas circulares sobre el haz de las hojas que aparecen principalmente hacia finales de verano y otoño. Color marrón claro en el centro de la mancha y más oscuro hacia los márgenes de la misma. No confundir con el ataque de *Phoma concéntrica* Desm. que da una coloración marrón uniforme de la mancha con picnidios en disposición concéntrica.

Análisis de la muestra

Poner hojas verdes con manchas en cámara húmeda. A las 24-48 horas se observan al binocular los acérvulos maduros en disposición concéntrica sobre la mancha. Este hongo tiene un buen crecimiento en (PDA) patata-dextrosa-agar.

Acérvulos en manchas de hojas al microscopio estereoscópico.

Identificación

Observación, mediante montaje, de los acérvulos maduros al microscopio óptico con el objetivo de 40x. Presencia de setas. Conidióforos simples, filiformes y cortos con conidios uniceluiares, hialinos y alantoides, según bibliografía.

Bibliografía

Sutton T. B., 1980: The Coelom-ceres. CAB, 535.

GRUPO DE TRABAJO DE LABORATORIOS DE DIAGNÓSTICO

Laboratorio ele Diagnóstico del S.P.V. del D.A.R.P. de la Generalitat de Catalunya

Montón, C. y García, F.

Figura 1: HTML de la guía de plagas (ficha 34)

Estos documentos tienen una interfaz muy simple en la que sólo se distinguen los títulos, el texto y las imágenes sobre un fondo blanco. Con este proyecto se pretende que la información de las guías pueda estar almacenada en una base de datos de manera que los datos puedan ser utilizados por aplicaciones y se puedan consultar fácilmente.

1.2 **Problema que se aborda**

Al tratarse de colecciones muy grandes y documentos extensos, para facilitar su uso es necesario crear un proceso que automatice la extracción de los datos. Sin embargo, los documentos son bastante heterogéneos. Los documentos de una colección tienen una estructura diferente a los de las demás. A su vez, un documento puede variar su estructura respecto de otros documentos de su misma colección. Por esa razón, el proceso ha de tener en cuenta estos factores para poder sacar la máxima información sin pérdidas de datos. El hecho de crear un proceso único independiente de la colección a procesar implicaría una inversión de tiempo y esfuerzo muy elevado, algo que queda fuera del alcance de este trabajo. Es por eso que el proceso de extracción desarrollado en este trabajo tiene en cuenta la distinción de colecciones.

Otro problema es que las colecciones se solapan entre sí, de modo que en una colección se puede encontrar información perteneciente a otra de las colecciones. Por ejemplo, las guías de cultivos contienen tablas y anexos con información sobre las plagas. Las guías de productos sanitarios contienen la relación de cultivos y plagas en los que pueden usarse. Las guías de plagas tienen un listado de cultivos afectados. Es necesario un proceso que integre estos datos de forma que la información se encuentre completa y relacionada.

1.3 **Alcance, objetivos y limitaciones**

El objetivo de este proyecto es hacer las guías accesibles mediante la transformación de los documentos HTML a un modelo de base de datos para que la información pueda ser operable de forma sencilla. Con los datos almacenados en una base de datos se pueden crear aplicaciones futuras que usen como entrada la información de las guías de plagas. A modo de validación se ha creado una aplicación web, de manera que no haya que ir buscando manualmente la información entre multitud de documentos. Con una búsqueda sencilla por palabras clave y filtros para seleccionar la colección se puede encontrar la información deseada de forma rápida. Además se puede acceder de una guía a otra de diferente colección si aparece mencionada en el documento. El número de documentos por colección se puede observar en la tabla 1.

Colección	Número de documentos
Guías de plagas	394
Guías de cultivos	34
Guías de productos fitosanitarios	2051

Tabla 1: Requisitos funcionales del proceso de extracción

Por otro lado, este sistema de extracción no es perfecto, ya que parte de unos documentos que pueden contener errores de caracteres, falta de datos respecto de las guías originales y heterogeneidad en la estructura del HTML. En la sección 2.6 se describen todos los problemas identificados en las fuentes de datos y cómo se han tratado.

1.4 **Metodología**

Se ha seguido una metodología iterativa. Inicialmente se han estudiado las diferentes tecnologías para abordar el problema. Este análisis puede encontrarse en el Anexo I. Una vez escogidas las tecnologías se ha llevado a cabo el desarrollo del proceso de extracción de los datos. Durante esta etapa se han realizado pruebas (Anexo II) para verificar el correcto funcionamiento del mismo. Posteriormente, se ha tenido en cuenta la duplicidad de los datos y se ha creado un proceso integrador para filtrar los datos y eliminar duplicados en la base de datos. Por último, se ha creado una aplicación web a modo de verificación de los resultados.

1.5 **Estructura del proyecto**

El proyecto se compone de tres módulos fundamentales: el proceso de extracción de los datos, la base de datos y la aplicación web que los muestra. El proceso de extracción es un proceso que se ejecutará únicamente cuando haya nuevos datos en la colección original ya que se actualizan las guías con poca frecuencia. Este proceso único está subdividido en fases (extracción, transformación o procesamiento y carga de los datos) que se aplican a cada colección por separado. A su vez, también se compone de un proceso integrador que unifica aquellos datos que relacionan las colecciones. Este proceso puede verse con más detalle en el apartado 2.4.

La base de datos es no relacional y está desplegada en el servicio Multi-Cloud de MongoDB Atlas [21]. Este servicio se encarga de automatizar el despliegue y proporcionar las herramientas para poder establecer la conexión y realizar las consultas. Los datos están en tres Schemas de MongoDB, uno para cada colección. Para relacionar las colecciones entre sí se guardan los identificadores de los documentos.
Por último, la aplicación web se encarga de conectar con la base de datos y mostrar la información de las guías. Es una aplicación sencilla que cuenta con dos vistas, una para el buscador y otra para la información detallada. Se puede navegar entre colecciones relacionadas. Las tecnologías utilizadas son ReactJS [9] para el front y NodeJS [7] y Express [8] para el backend.

1.6 **Tecnologías utilizadas**

Para desarrollar este trabajo se pueden utilizar multitud de tecnologías, pero es importante identificar qué es lo que nos ofrece cada una y cuál se ha de usar en cada situación. Para ello, se ha hecho una distinción de los diferentes módulos de los que se compone el proyecto:

- **Proceso ETL (extracción, transformación y carga) [4]:** Éste es el módulo más importante del proyecto, ya que se va a tratar toda la información y se va a insertar en la base de datos. Para este proceso se ha realizado un análisis de tecnologías y se ha optado por Java Spring Batch.
- **Base de datos:** Se necesita una base de datos capaz de almacenar mucha información de manera sencilla, rápida y fácilmente accesible, por lo que se ha optado por MongoDB.
- **Servidor Backend API:** Sirve para realizar las peticiones a la base de datos desde la aplicación web. La tecnología empleada es la definida en el stack MERN: NodeJS y Express.
- **Frontend de la aplicación web:** Debido a su extendido uso, la experiencia previa y su concordancia con las tecnologías anteriormente mencionadas, se ha utilizado ReactJS.

El análisis detallado de la elección de tecnologías se puede encontrar en el Anexo I.

2. Trabajo realizado

Se ha desarrollado un sistema de extracción de datos que lee unos documentos HTML de guías de plagas y guarda sus contenidos en una base de datos documental. Este sistema de extracción cuenta también con un proceso de integración que relaciona guías entre distintas colecciones. Además se ha desarrollado una aplicación web que permite visualizar y navegar entre estas guías.

2.1 **Análisis de requisitos**

En este apartado se detallan los requisitos del proceso de extracción y de la aplicación web.

2.1.1 Requisitos funcionales del proceso de extracción

Identificador del requisito	Especificación del requisito
RF1-ETL	El sistema debe leer todos los datos de las colecciones de plagas y productos fitosanitarios.
RF2-ETL	El sistema debe leer los contenidos de las secciones “Cuadro de gestión integrada de plagas” y “Anexo fichas de plagas“ de la colección de cultivos.
RF3-ETL	El sistema debe guardar los datos en una base de datos documental.
RF4-ETL	El sistema debe guardar tanto el texto como las imágenes extraídos de los documentos.
RF5-ETL	El sistema debe integrar los datos de diferentes colecciones relacionados entre sí.

Tabla 2: Requisitos funcionales del proceso de extracción

Aclaración del RF5: en el contenido de una guía se pueden encontrar menciones a otras guías de otras colecciones, por lo que se necesita guardar esa relación. P.e. las guías de cultivos contienen tablas y anexos con información extra sobre las plagas existentes, por lo que esa información se puede agregar a la plaga en cuestión en la base de datos.

2.1.2 Requisitos no funcionales del proceso de extracción

Identificador del requisito	Especificación del requisito
RNF1-ETL	Los datos a procesar están en formato HTML.

Tabla 3: Requisitos no funcionales del proceso de extracción

2.1.3 Requisitos funcionales de la aplicación web

Identificador del requisito	Especificación del requisito
RF1-App	La aplicación debe permitir mostrar el listado de documentos de la base de datos filtrado por tipo de colección.
RF2-App	La aplicación debe permitir realizar una búsqueda de documentos por nombre de la colección seleccionada.
RF3-App	La aplicación debe permitir ver la información detallada de cada documento.
RF4-App	La aplicación debe permitir navegar entre documentos relacionados entre sí.

Tabla 4: Requisitos funcionales de la aplicación web

2.1.4 Requisitos no funcionales de la aplicación web

Identificador del requisito	Especificación del requisito
RNF1-App	La aplicación debe ser intuitiva.

Tabla 5: Requisitos no funcionales de la aplicación web

2.2 **Estructura del proyecto**

El proyecto cuenta con la siguiente estructura:

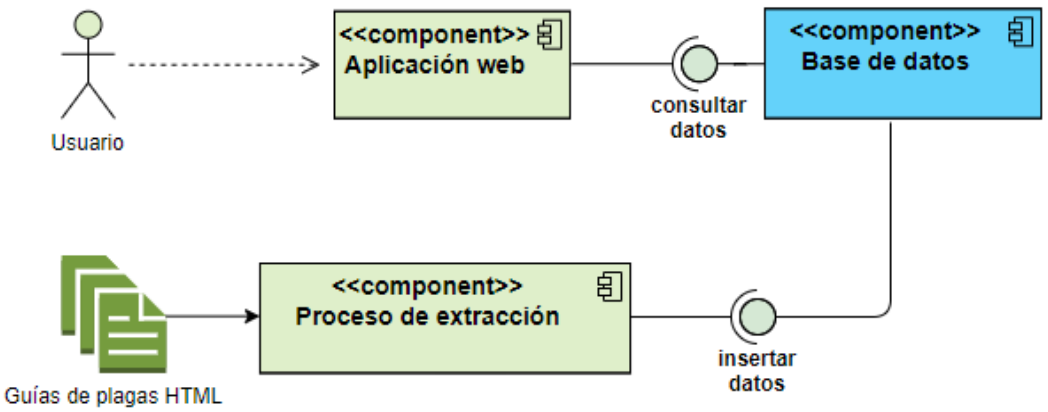


Figura 2: Diagrama de componentes del proyecto

El proyecto consta de tres módulos principales (Figura 2): la aplicación web, el proceso de extracción y la base de datos. La aplicación web se compone de dos elementos: la propia interfaz a la que accede el usuario y el servidor al que se realizan las peticiones de los datos. El servidor se encarga de procesar esas peticiones y extraer los datos solicitados de la base de datos.

Por otro lado, el proceso de extracción es un proceso largo, pensado para ejecutarse manualmente en periodos de tiempo grandes. Este proceso lee los documentos de entrada, los procesa y guarda los datos de salida en la base de datos.

2.3 **Proceso de extracción**

En este apartado se explica en detalle las características del proceso de extracción. Una vez analizada la estructura de los documentos HTML y sus contenidos se ha realizado un análisis de tecnologías cuya elección ha sido Spring Batch [5]. Esta tecnología permite definir una serie de funciones reutilizables para procesar grandes cantidades de información por lotes.

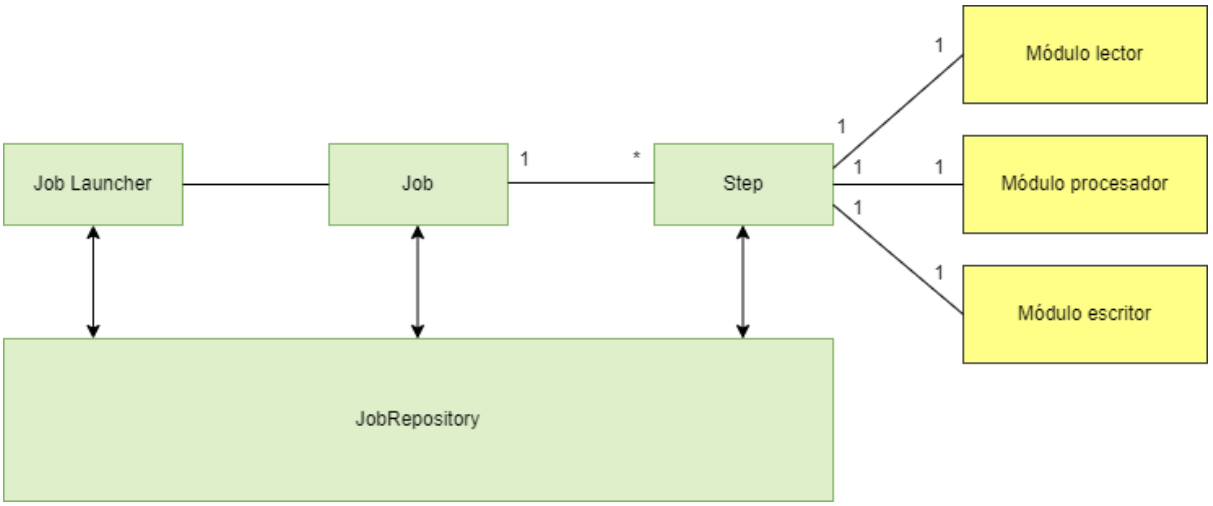


Figura 3: Arquitectura de Spring Batch

Esa es la arquitectura que proporciona Spring Batch (Figura 3). Por defecto vienen módulos de lectura, procesamiento y escritura de varios tipos de documentos (p.e. hojas de cálculo). Sin embargo, se han definido módulos personalizados que se detallan en los siguientes apartados. Existe un JobRepository que almacena información sobre la ejecución de todo el proceso. El JobLauncher es el encargado de lanzar una serie de jobs o tareas. Un job es una tarea en concreto, en este caso es la extracción de datos de las guías. Cada job cuenta con uno o varios pasos (steps), que a su vez cuentan con tres módulos bien definidos: el lector de los datos, el procesador que hace transformaciones con los datos y el escritor que los guarda en la base de datos. El esquema de Spring Batch adaptado a este proyecto se observa en la figura 4.

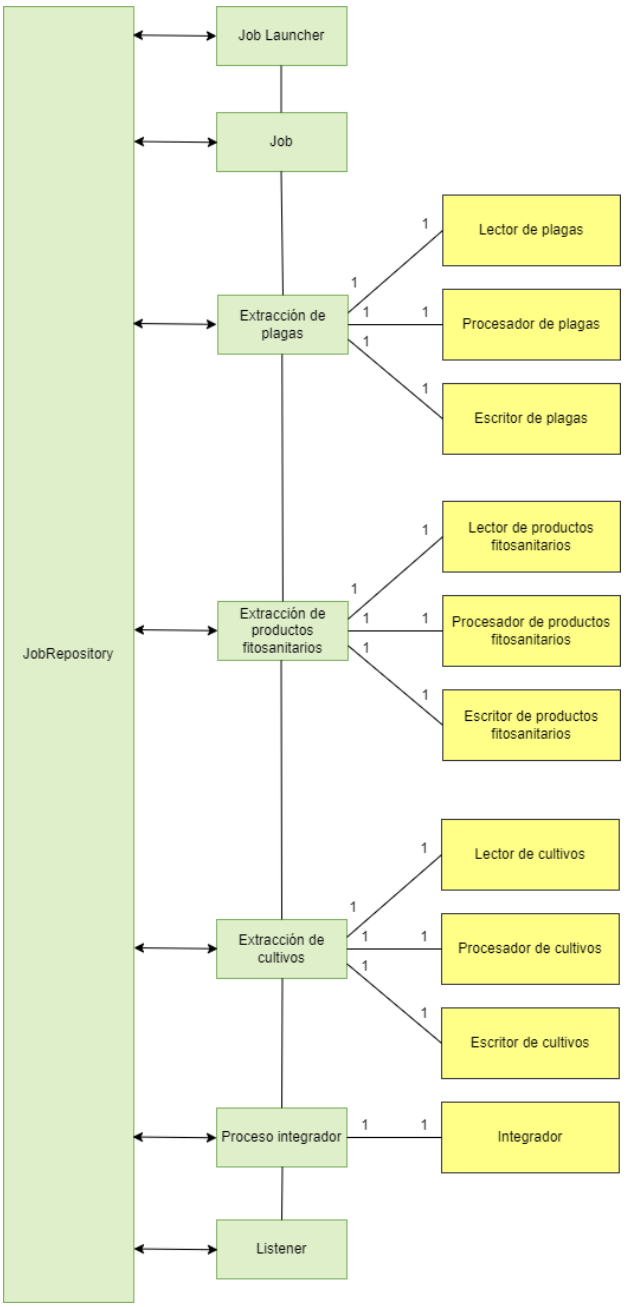


Figura 4: Esquema del proceso de extracción

Respecto al esquema original de Spring Batch se mantiene un solo Job, ya que sólo hay una tarea que es la de extracción de los datos de los documentos. Sin embargo, esta tarea cuenta con cuatro pasos, tres de ellos dedicados a la extracción de los documentos de las tres colecciones respectivamente, y una tarea de integración. En cada paso de extracción se han sobreescrito los módulos lector, procesador y escritor de Spring Batch para adaptarlos a las necesidades de cada colección. Por último, se ha añadido un elemento Listener que indica el estado final de la ejecución del proceso, si se ha completado correctamente o ha fallado.

El proceso de extracción sigue los siguientes pasos:

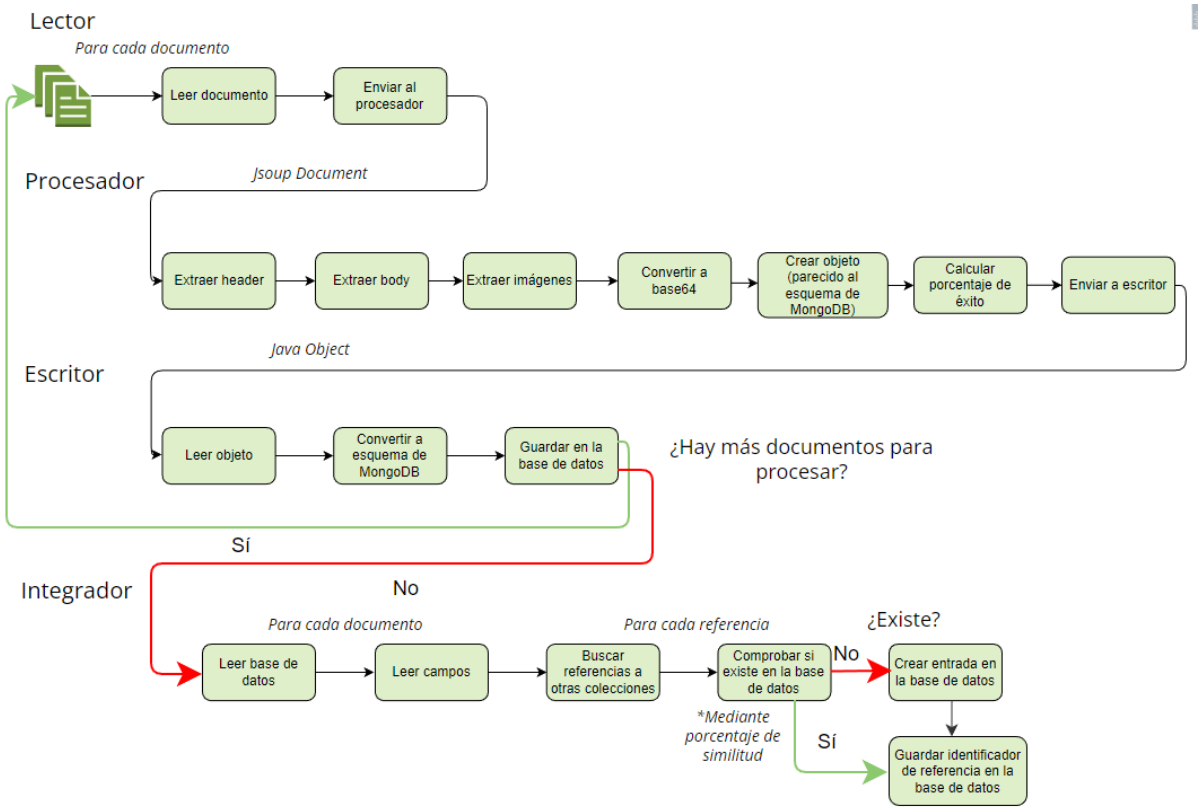


Figura 5: Esquema del proceso de extracción

En la figura 5 se ven las tareas que realiza el proceso de extracción. Cada módulo lector y procesador tiene su implementación propia según la colección que procesa, pero las tareas son las mismas. El lector itera sobre la colección y lee la información de cada documento para mandárselo al procesador.El procesador lee los datos y separa los campos, diferenciando entre encabezado y contenido. Esto se debe a que es más importante conseguir correctamente el identificador del documento que el resto de campos, así que el encabezado se procesa teniendo más en cuenta los casos de heterogeneidad. Con el texto ya procesado, se leen las imágenes y se convierten a base64 dada su facilidad de uso a la hora de almacenar en bases de datos y mostrar en aplicaciones web. Una vez obtenidos todos los datos se calcula el porcentaje de éxito y se crea el objeto que se envía al escritor. En el escritor se lee el objeto recibido y lo transforma en una plantilla de esquema para guardar los datos en MongoDB. El procesamiento de cada documento es independiente del resto, por lo que se ha paralelizado [10] para que sea más rápido. Por último, cuando ya se han procesado todos los documentos se ejecuta un proceso integrador que relaciona las colecciones entre sí.

2.3.1 Módulo lector

Este módulo se utiliza para leer el contenido de los documentos en las distintas colecciones. Dada una colección, se itera por todos los documentos guardando sus contenidos en objetos de tipo Document de Jsoup, una librería que facilita el procesamiento de HTML. Estos documentos se guardan en una lista que se irá vaciando conforme se envíen al

módulo procesador. Los lectores por defecto de Spring Batch [6] ofrecen una implementación del módulo lector para leer diversos tipos de documentos y bases de datos. Sin embargo, no incluye una implementación para procesar HTML, y menos para el problema de heterogeneidad de estas colecciones. En este caso se han tenido que crear tres módulos lectores distintos que implementan el módulo ItemReader de Spring Batch, uno para cada colección de documentos.

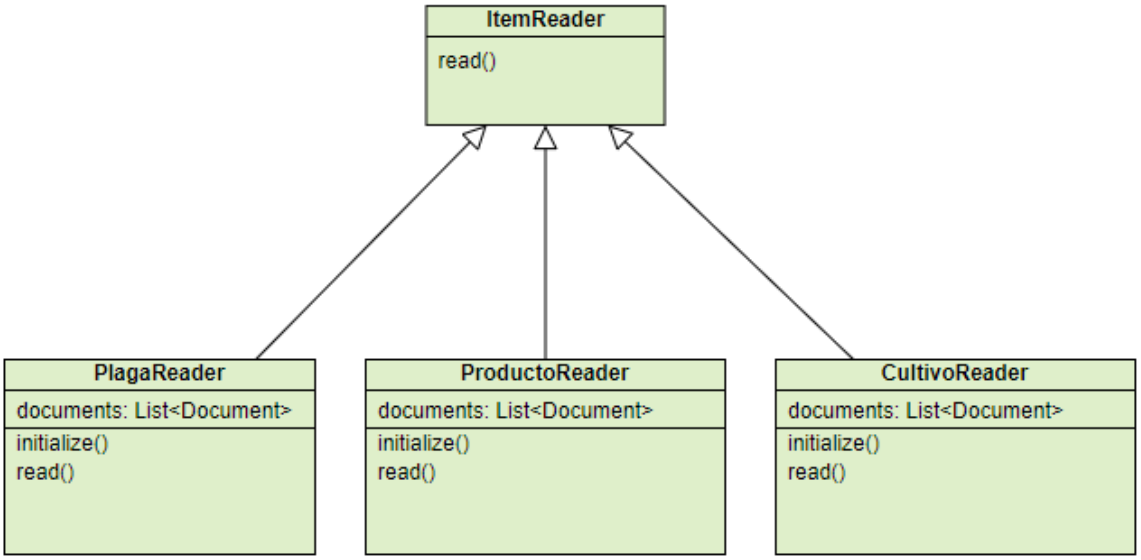


Figura 6: Diagrama de clases módulo lector

Este módulo hace una lectura secuencial de los documentos contenidos en una colección ya que Spring Batch no permite paralelizarlo. Este proceso lector es secuencial, es decir, sólo lee un documento a la vez. Sin embargo, tanto el procesador como la carga en la base de datos son independientes, por lo que se pueden paralelizar [10] reduciendo así el tiempo de ejecución del proceso.

2.3.2 Módulo procesador

De igual manera que el módulo lector, se ha implementado un módulo procesador personalizado para cada colección.

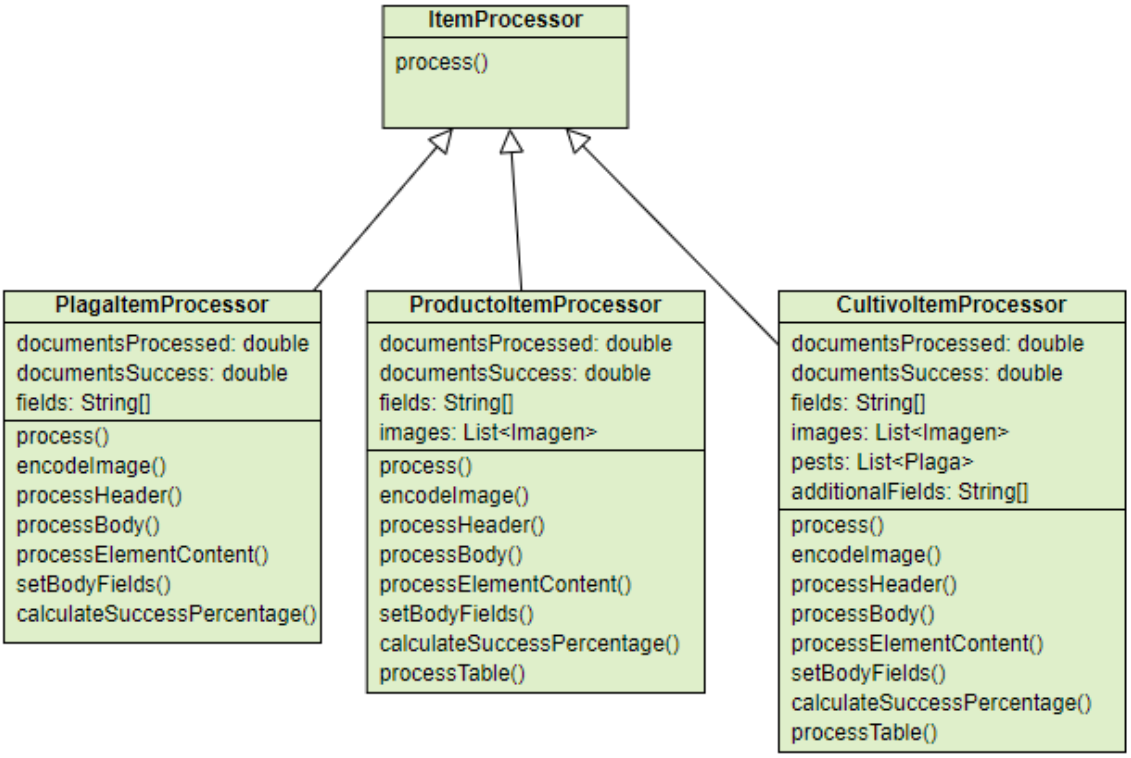


Figura 7: Diagrama de clases módulo procesador

Este módulo sirve principalmente para separar los distintos campos del documento y guardarlos como un objeto de Java. Esto se hace mediante la librería Jsoup mencionada anteriormente, la cual nos permite seleccionar en el documento el contenido de texto de cada etiqueta HTML, así como de sus atributos. Como se ve en la figura 7, cada implementación del módulo escritor tiene sus peculiaridades. Aparte de leer los campos del documento correspondientes a su colección, se han de tener en cuenta otros atributos. Para las plagas hay que identificar su tipo, ya que puede ser un insecto, una enfermedad o una mala hierba. Esta información nos la proporcionan las guías de los cultivos, por lo que se extrae este campo en el CultivoItemProcessor. Como las guías de plagas no especifican este atributo, se guarda como cadena vacía y no se muestra en la aplicación web. Además, la información de las plagas se recoge tanto en el módulo lector de plagas como en el de cultivos, ya que las guías de cultivos contienen información adicional de las plagas. Esta información adicional son atributos nuevos, por lo que se guardan para que en el módulo integrador se añadan a las plagas ya existentes.

Identificando las pautas comunes entre los documentos de una misma colección se ha escrito un algoritmo que intenta contemplar la mayoría de los casos de heterogeneidad con el fin de leer la mayor cantidad de datos posibles de los documentos. Esto implica que se pueden perder algunos datos debido a etiquetas no contempladas, estructura de las etiquetas muy heterogénea o atributos de etiqueta no encontrados.

Ficha 42

Exobasidium japonicum Shirai	AZALEA
Agallas	*Rhododendron* sp.

Ficha 34

***Colletotrichum trichelium* (Fr.) Duke**

HIEDRA

Hederá helix L.

Figura 8: Comparación de títulos de las guías de plagas (ficha 42 y ficha 34)

En la figura 8 se observa un ejemplo de la heterogeneidad entre documentos de una misma colección. En este caso es la forma de presentar el título, donde un documento lo muestra con una tabla mediante la etiqueta <table>, el otro lo muestra sin ella mediante etiquetas de párrafo <p>. Si el proceso sólo identificase títulos contenidos en una tabla podría no se guardaría correctamente el título de la ficha 34. Para evitar pérdida de datos o lectura de datos erróneos se han analizado previamente los casos más reiterados de heterogeneidad y se han tenido en cuenta mediante sentencias condicionales a nivel de código. A continuación se muestran otros ejemplos de heterogeneidad en los documentos:

USO	AGENTE	Dosis (ver nota)	FORMA Y ÉPOCA DE APLICACIÓN (Condic. Específico)
Cítricos	HERIDAS PODA		
	HERIDAS INJERTOS		

N° REGISTRO: 22750

N° REGISTRO: ES-00319

AFFIRM OPTI

USO	AGENTE	Dosis Kg/ha	N° Aplic.	Intervalos	Vol. Caldo	FORMA Y ÉPOCA DE APLICACIÓN (Condic. Específico)
Manzano	CYDIA ORUGAS MINADORAS ORUGAS ENROLLADORAS	2,06	3	7	500-1500 l/ha	Tratamientos al aire libre. No utilizar aceite como adyuvante. Aplicar el producto desde cuajado (diámetro del fruto hasta 10 mm) hasta madurez de consumo (frutos con aroma y firmeza típicos)(BBCH 71-89). Aplicar un máximo de 6.18 kg pf/ha y año.

Figura 9: Comparación de “Usos y dosis” de la guía de productos fitosanitarios (nº 22750 y nº ES-00319)

Plagas principales	Seguimiento y estimación del riesgo para el cultivo	Medidas de prevención y/o culturales	Umbral/Momento de intervención	Medidas alternativas al control químico (*)	Medios químicos

Plagas principales	Seguimiento y estimación del riesgo para el cultivo (1)	Medidas de prevención y/o culturales	Umbral/Momento de intervención	Medidas alternativas al control químico (2)	Medios químicos

Figura 10: Comparación de “Cuadro de estrategia” de la guía de cultivos (Forestales frondosas y Hortícolas fresa y fresón)

En la figura 9 aparecen en una tabla cuatro campos el número de registro, mientras que en la otra tabla hay siete campos y el número de registro aparece antes de la tabla. Esto se ha solucionado guardando todo el contenido de texto de la tabla directamente. A su vez, se han guardado por separado las columnas uso y agente para relacionarlas con el resto de colecciones. Otro ejemplo de heterogeneidad se encuentra en la figura 10, donde vemos que los campos son los mismos pero varían en un carácter y puede no detectar bien el campo a guardar.

Con Jsoup se leen los atributos de las etiquetas, por lo que se puede guardar la dirección del directorio en el que se encuentra cada imagen. Sin embargo, se quieren almacenar las imágenes en la base de datos sin depender de que las imágenes estén en un directorio en concreto. Para ello se han codificado mediante la librería de Java Base64, que permite almacenarlas como una cadena en la base de datos y a la aplicación web mostrarlas de manera sencilla. Una vez leída la información de un documento, se calcula el porcentaje de éxito (Anexo II) de los ya procesados y se envía el documento actual al módulo escritor.

Al tratarse de una de las partes con más carga de trabajo del proceso (sólo por detrás del módulo integrador) se ha paralelizado para reducir el tiempo de ejecución. Se han encapsulado los tres procesadores mediante el tipo de objeto AsyncItemProcessor. Esto hace que los procesadores sean asíncronos y, asignándole un TaskExecutor se ha creado una pool de workers a la que se le pueden asignar parámetros como el número de workers, si es flexible o fija, etc. Con esto podemos procesar más documentos al mismo tiempo y personalizarlo manualmente acorde a los recursos del equipo en el que se ejecute el proceso.

2.3.3 Módulo escritor

En los módulos anteriores se han sobreescrito el lector y procesador por defecto de Spring Batch para el objetivo de este proyecto. Sin embargo, no ha hecho falta sobreescribir el escritor por defecto ya que Spring Batch proporciona el MongoItemWriter. Se han creado tres MongoItemWriter, uno por colección. Se configura la plantilla indicando que los datos de entrada son clases de Java y se le pasa el tipo de objeto de cada colección, uno para cada ItemWriter. Éste transforma las clases creadas en Java a un esquema de MongoDB y guarda en la base de datos cada documento que le llega del procesador. Se ha configurado el acceso a la base de datos en el documento de propiedades de Spring Batch. Al igual que el procesador, también se ha paralelizado encapsulando este escritor en un AsyncItemWriter, el cual permite la asincronía y se complementa con el procesador. También se han asignado los objetos de Java referentes a cada colección como Future, que obtiene el valor cuando se termina de calcular.

2.3.4 Módulo integrador

Con el fin de mejorar la experiencia del usuario se ha relacionado aquellas menciones entre documentos de distintas colecciones para que pueda navegar entre resultados relacionados en la web. Por ejemplo, un producto fitosanitario tiene asociados unos cultivos en los que se puede usar para combatir unas plagas determinadas. El esquema del proceso integrador es el siguiente:

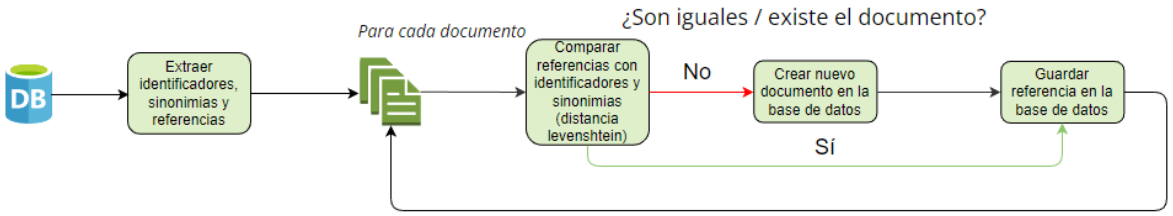


Figura 11: Diagrama de flujo del proceso integrador

Este algoritmo (Figura 11) lee los identificadores y sinonimias de la base de datos, así como de las referencias que pueda haber a documentos de las otras colecciones. Entonces compara si la referencia encontrada es un documento ya existente en la base de datos y, en caso de no serlo, lo crea. Teniendo en cuenta la posibilidad de que los textos pueden no ser exactos a la hora de compararlos, se ha creado una función que implementa la métrica de la distancia de Levenshtein [11]. Esta métrica compara si dos cadenas de texto hacen referencia a lo mismo midiendo su porcentaje de similitud. Se ha hecho un análisis para determinar el porcentaje de similitudes encontradas con éxito y así evitar falsos positivos.

Este módulo se encarga también de completar la información de las plagas. Las guías de cultivos presentan información adicional sobre las plagas, de modo que se busca en la base de datos la plaga y, en caso de que exista se añaden los atributos y si no existe se crea una nueva entrada con esos atributos. Este proceso no sigue la estructura de lector-procesador-escritor, por lo que se ha definido como un Tasklet de Spring Batch, que nos permite tener un proceso independiente que no sigue dicha estructura. Este proceso se ejecuta siempre al acabar la extracción, procesamiento y guardado de los datos, ya que depende de ellos. No se paraleliza porque se ejecuta únicamente al finalizar los procesos anteriores.

2.4 **Estructura de los datos**

Los datos utilizados en este proyecto provienen de tres colecciones extensas de documentos correspondientes a guías de plagas en la agricultura. En dichas guías existe información redundante o de poco valor para el proyecto, por lo que se han estudiado los datos de entrada y para poder definir correctamente los datos útiles. En esta sección se analiza la estructura de las colecciones base y la diseñada para guardar la información en la base de datos.

2.4.1 Estructura de los datos de entrada

Los datos a procesar vienen dados en tres colecciones distintas. Como se ha comentado en la tabla 1 de la introducción, estas tres colecciones tienen miles de documentos HTML cuyas etiquetas difieren entre colecciones e incluso entre documentos de una misma colección, haciendo que sea más complicado crear un proceso homogéneo para todas las colecciones. Existe una colección que describe cada plaga, otra que describe cada producto fitosanitario y otra que describe las plagas existentes en cada cultivo en concreto. De las dos primeras interesa guardar todo el contenido ya que aporta información útil y es único

para cada documento. De los cultivos, sin embargo, sólo interesa guardar las secciones abordan el contenido referente a las plagas, ya que el resto de información es común para todas las guías y no aporta información sobre el cultivo en cuestión.

La información relevante son los textos legibles por los usuarios y las imágenes. No es necesario guardar las etiquetas propias del lenguaje HTML de los documentos ya que se ha creado una nueva interfaz web de cero para mostrar los datos.

2.4.2 Estructura de los datos de salida

Debido a la escasez de relaciones de los datos y a las ventajas que aportan las bases de datos no relacionales para tratar grandes cantidades de documentos se ha optado por MongoDB. Cada documento contiene una serie de apartados distinguidos por un título. El diseño de la base de datos basa sus atributos en cada uno de estos apartados, añadiendo a su vez alguno adicional. A continuación se muestra el modelo de datos utilizado:

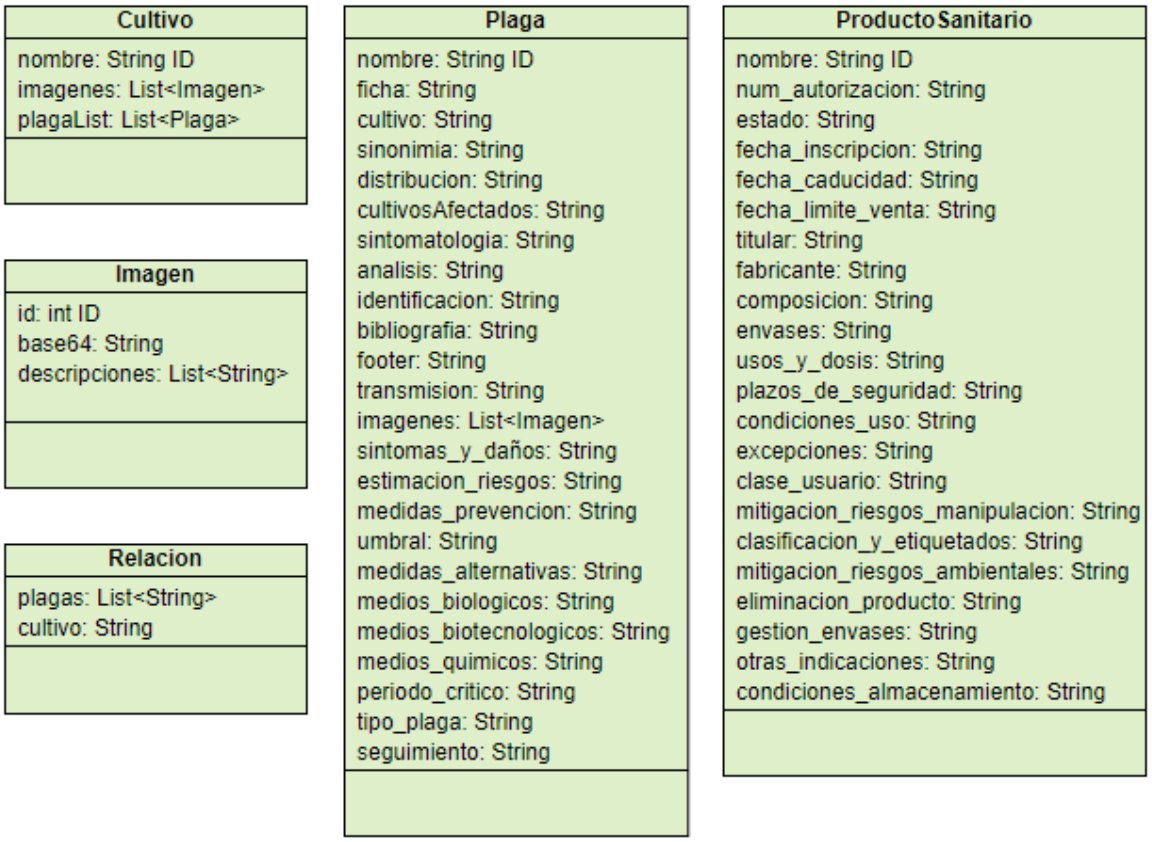


Figura 12: Modelo de datos

Como se puede observar en la figura 12, los datos están representados como documentos ya que la tecnología empleada es MongoDB, una base de datos documental. A continuación se van a explicar los aspectos más importantes de cada esquema:

- **Cultivo:** Correspondiente a la colección de guías de cultivos. Sólo se guarda el nombre del cultivo y las plagas relacionadas, ya que el resto de información es genérica para todos los documentos de los cultivos.
- **Plaga:** Correspondiente a la colección de guías de plagas. Destaca el campo de cultivos afectados, que permite relacionarlo con la colección de cultivos. La mitad de los atributos provienen de la colección de plagas y la otra mitad de la colección de cultivos.
- **ProductoSanitario:** Correspondiente a la colección de guías de productos fitosanitarios. Destaca el campo usos y dosis, en el que se encuentran referencias a plagas y cultivos.
- **Imagen:** Subdocumento que guarda la imagen en base64 y la/s descripciones asociadas.
- **Relación:** Subdocumento que guarda la relación de los productos fitosanitarios con las otras colecciones.

2.5 **Aplicación web**

La aplicación web sirve de utilidad para los usuarios finales y como validación del proceso. Esta aplicación sirve para comprobar que los datos se han guardado correctamente en la base de datos y dar facilidades de navegación. En ella podemos visualizar los datos más relevantes de las guías y navegar entre ellos, facilitando su búsqueda mediante un buscador y documentos relacionados. La tecnología empleada es ReactJS [9] para el frontend y NodeJS [7] para el backend. En esta sección se profundiza en su funcionamiento y las decisiones tomadas.

2.5.1 Objetivo de la aplicación

La aplicación tiene dos objetivos. El primero, servir de validación de que el proceso de extracción ha funcionado correctamente. Los textos se pueden observar tal y como se encuentran en la base de datos y las imágenes se visualizan correctamente. El segundo, proporcionar una utilidad para usuarios, principalmente del sector agrícola. Como la aplicación está enfocada al sector agrícola, los usuarios no necesitan tener conocimiento avanzado de las tecnologías. La aplicación se ha diseñado con el fin de ser intuitiva, sencilla y fácil de usar cuidando el tamaño de los textos.

2.5.2 Estructura

La web cuenta con dos módulos: el de interfaz visual y el servidor que se comunica con la base de datos. De acuerdo con el stack MERN se ha empleado ReactJS para lo visual y NodeJS y Express para el servidor backend. El funcionamiento es sencillo, el cliente hace peticiones HTTP al servidor mediante la librería Axios [22] para solicitar los datos y el servidor cuenta con cuatro controladores que gestionan las peticiones. Hay un controlador específico por cada colección y hay otro que devuelve la información de las tres colecciones juntas. Mediante estos endpoints se realizan consultas a la base de datos y se devuelven al cliente en formato JSON.

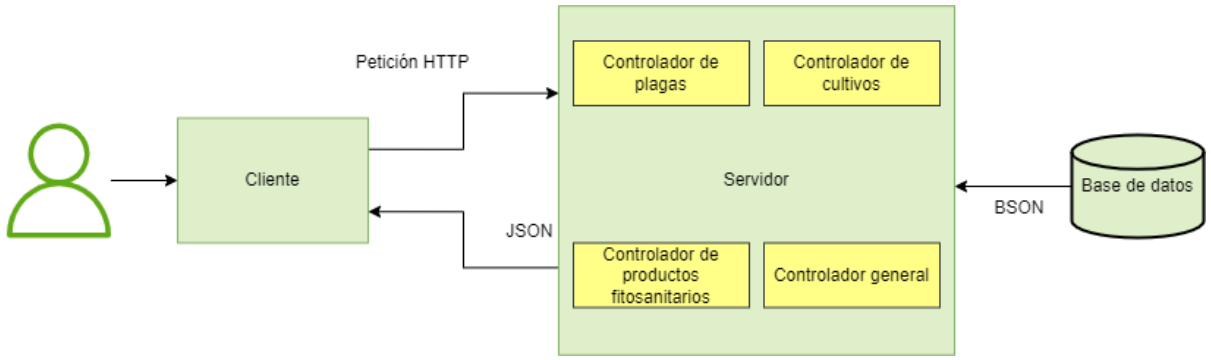


Figura 13: Estructura de la aplicación web

2.5.3 Interfaz

La web cuenta con un buscador con filtros que permite recabar las guías mediante palabras clave (Figura 14). La lista se filtra dinámicamente y muestra una foto y los campos más relevantes de cada documento. Haciendo clic en un elemento de la lista se puede observar toda la información de esa guía, así como de las guías relacionadas. De esta manera se puede navegar de una guía a otra si están relacionadas. Las imágenes se muestran con su respectiva descripción en un carrusel.

Figura 14: Pantalla principal de la aplicación web

Figura 15: Pantalla de detalles de una guía de plagas de la aplicación web

En la pantalla de detalles (Figura 15) se pueden ver las imágenes y toda la información de una de las guías de plagas. Esta ventana es equivalente para los cultivos y productos fitosanitarios, pero tiene ligeras variaciones según la colección, ya que los campos son distintos. En el caso de los productos sanitarios no hay imágenes, por lo que sólo se observan sus campos de texto y los documentos relacionados. En los cultivos no hay campos de texto,por lo que se muestran las imágenes y un listado de las plagas relacionadas.

2.6 **Problemas encontrados**

La mayor parte de los problemas han surgido en el proceso de extracción, ya que es el núcleo del proyecto. Principalmente la heterogeneidad de los documentos y la longitud de las colecciones han hecho que las pruebas sean costosas en tiempo. En esta sección se comentan algunos de los problemas encontrados a la hora de desarrollar el proyecto.

2.6.1 Adaptación a Spring Batch

Spring Batch era la mejor opción para el desarrollo del proyecto. No obstante, no había trabajado antes con esta tecnología, por lo que había que aprenderla desde cero. No ha supuesto un problema muy grande ya que es fácil de entender, pero hay que saber encontrar las herramientas que te permiten implementar la estructura adaptada a tu problema en concreto. En este caso Spring Batch me proporcionaba un módulo escritor para MongoDB, pero tanto el lector como el procesador han sido implementados desde cero.

2.6.2 Heterogeneidad de los documentos

Existe heterogeneidad entre documentos de distintas colecciones y, a su vez, entre colecciones diferentes.

2.6.2.1 Heterogeneidad entre colecciones

- Los campos de las guías de diferentes colecciones son distintos.
- La colección de cultivos muestra la información en tablas y anexos, mientras que la colección de plagas lo hace en campos y la de productos fitosanitarios en campos y tablas.
- Las colecciones de plagas y cultivos tienen imágenes variadas, mientras que la de productos tiene una sola imagen que se repite para todos los documentos de la colección.

2.6.2.2 Heterogeneidad entre documentos de una misma colección

- No aparecen siempre todos los campos.
- Las etiquetas usadas para definir el nombre del campo o el contenido varían. P.ej.: en un documento puede ser la etiqueta <h> y en otro la etiqueta <p>

La solución ha sido crear módulos de procesamiento específicos para cada colección y cubrir la mayoría de casos de heterogeneidad mediante condicionales a nivel de código.

2.6.3 Texto ilegible

Este problema viene dado por el proyecto anterior que transformaba las guías de PDF a HTML. Al escanearlas se producían errores resultantes en caracteres malformados. No hay una manera sencilla de solucionar este problema, por lo que pueden aparecer estos caracteres a la hora de visualizar los datos dificultando así la lectura de la información.

2.6.4 Relaciones entre documentos

Al tener una base de datos no relacional se ha buscado una forma de representar las relaciones entre documentos de distintas colecciones. La solución es sencilla, guardar el identificador del documento al que se está haciendo referencia. Esto se ve reflejado en la navegación por la web, en la que podemos saltar de un documento a otro desde la vista de detalles.

2.6.5 Proceso lento

Los procesos ETL suelen ser procesos largos. Esto tiene implicaciones a la hora de hacer pruebas, ya que cada ejecución puede tardar varios minutos. Para hacer las pruebas más rápido se indicaba en la configuración de Spring Batch que se ejecutase sólo el paso a testear, agilizando así el proceso. Además se han paralelizado los módulos procesador y escritor para conseguir mayor rendimiento.

2.6.6 Ralentización de las conexiones a la base de datos

A la hora de realizar pruebas, cada ejecución se hacía más lenta. Esto era debido a que cada vez se generaban más conexiones a la base de datos y no se cerraban. El servicio de MongoDB Atlas [21] no permite cortar las conexiones en su versión gratuita, por lo que había que borrar la base de datos y crearla de nuevo. Esto se ha solucionado cerrando correctamente las conexiones en el código.

3. Conclusiones

3.1 **Aprendizaje**

El aprendizaje se encuentra mayoritariamente en el proceso de extracción, ya que para el desarrollo de la aplicación web se contaba con experiencia previa. Se ha profundizado en los procesos ETL y sus implicaciones, las diferentes tecnologías que permiten implementarlos y sus características. Se ha descubierto una herramienta muy potente para este tipo de procesos como lo es Spring Batch, la cual te da el esqueleto del proceso y te permite personalizar los módulos base, además de otras opciones de configuración como lotes a procesar y paralelización. También se ha tenido que lidiar con problemas complejos como lo es la heterogeneidad en la lectura de los documentos, cuya solución no es perfecta ya que siempre hay casos muy difíciles de tratar.

Ha sido un proyecto muy completo en el que se ha desarrollado un sistema de extracción de datos con una tecnología desconocida para mí y una aplicación web aplicando todos los conocimientos adquiridos en la carrera, desde el análisis de los requisitos hasta la implementación y verificación de los resultados.

3.2 **Trabajo futuro**

Como el objetivo del proyecto era hacer la información de las guías más accesible, el tener todos los datos en una base de datos permite crear cualquier aplicación que los utilice. Gracias a esto se podría desarrollar una versión móvil para la aplicación web de este proyecto para aumentar la accesibilidad de los datos. Otra idea sería crear un sistema inteligente con sensores que detecte cuándo un cultivo tiene plagas y proporcione sugerencias de productos fitosanitarios para combatirlas. Una posibilidad más real podría ser añadir nuevas colecciones de guías, lo cual implicaría añadir otro paso al proceso de extracción o intentar crear un proceso único para todas, de manera que se puedan procesar independientemente de la heterogeneidad de las colecciones. Esto último requeriría de un esfuerzo mucho mayor.

4. Bibliografía

[1] Ministerio de Agricultura, Ganadería y Pesca. “Guías de Gestión Integrada de Plagas.” Guías de Gestión Integrada de Plagas, https://www.mapa.gob.es/es/agricultura/temas/sanidad-vegetal/productos-fitosanitarios/guias-gestion-plagas/.

[2] Ministerio de Agricultura, Ganadería y Pesca. “Fichas.” Fichas mapa, https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_399.pdf.

[3] Ministerio de Agricultura, Ganadería y Pesca. “Registro de Productos Fitosanitarios.” Registro de Productos Fitosanitarios, https://www.mapa.gob.es/es/agricultura/temas/sanidad-vegetal/productos-fitosanitarios/registro/menu.asp.

[4] Colaboradores de Wikipedia. “Extract, transform and load.” Extract, transform and load - Wikipedia, https://es.wikipedia.org/wiki/Extract,_transform_and_load.

[5] Spring Framework. “Spring Batch.” Documentación Spring Batch - Wikipedia, https://spring.io/projects/spring-batch.

[6] Diego Rubio Abujas. “Introducción a Spring Batch: qué es, ventajas y ejemplo real.” Introducción a Spring Batch - Profile, https://profile.es/blog/que-es-spring-batch-ejemplo/.

[7] Colaboradores de Wikipedia. “Node.js.” Node.js - Wikipedia, https://en.wikipedia.org/wiki/Node.js.

[8] Colaboradores de Wikipedia. “Express.js.” Express.js - Wikipedia, https://en.wikipedia.org/wiki/Express.js.

[9] Colaboradores de Wikipedia. “React (JavaScript library).” React (JavaScript library) - Wikipedia, https://en.wikipedia.org/wiki/React_(JavaScript_library).

[10] Syeda Famita Amber. “Spring Batch Parallel Processing and Scaling 101” Procesamiento paralelo Spring Batch - Hevo Data, https://hevodata.com/learn/spring-batch-parallel-processing/#ws.

[11] Colaboradores de Wikipedia. “Distancia de Levenshtein.”Distancia de Levenshtein - Wikipedia, https://es.wikipedia.org/wiki/Distancia_de_Levenshtein.

[12] Alistair Miles. “petl - Extract, Transform and Load” Petl documentación - Petl, https://petl.readthedocs.io/en/stable/.

[13] Lucas Merencia. “node-cron” Node-cron - GitHub, https://github.com/node-cron/node-cron.

[14] Chad Auld. “Nextract” Nextract - GitHub, https://petl.readthedocs.io/en/stable/.

[15] Agmen-hu. “node-datapumps” Datapumps - GitHub, https://github.com/agmen-hu/node-datapumps.

[16] TaskRabbit. “Empujar” Empujar - GitHub, https://github.com/taskrabbit/empujar.

[17] Aleksandar Gakovic. “Extract Transform Load with Pandas” Pandas ETL - Medium, https://algakovic.medium.com/extract-transform-load-etl-with-pandas-d9e52c309e82.

[18] Crummy. “Beautiful Soup Documentation” Beautiful Soup - Crummy, https://algakovic.medium.com/extract-transform-load-etl-with-pandas-d9e52c309e82.

[19] Apache. “Apache Airflow” Apache Airflow - Apache, https://airflow.apache.org/.

[20] Microsoft. “SQL Server Integration Services” SSIS - Microsoft, https://learn.microsoft.com/es-es/sql/integration-services/sql-server-integration-services?view=sql-server-ver16.

[21] MongoDB Atlas. “Deploy a multi-cloud database” MongoDB Atlas- MongoDB, https://www.mongodb.com/atlas/database.

[22] Reed Barger. “How To Use Axios With React: The Definitive Guide (2021)” Axios-FreeCodeCamp, https://www.freecodecamp.org/news/how-to-use-axios-with-react/.

5. Anexo I: Análisis de tecnologías

Existen numerosas tecnologías que permiten implementar procesos ETL. A continuación se muestran las ventajas y desventajas de cada una de ellas y el porqué se ha elegido Spring Batch finalmente.

Nodejs

En Nodejs encontramos variedad de librerías que permiten implementar nuestro ETL, pero no hay una que venga con todas las funcionalidades necesarias, por lo que hay que combinarlas.

Librerías
- **Node-cron [13]->** Para ejecutar tareas periódicas - **ETL ->** Permite bulk insert (insertar múltiples datos en tablas), permite leer csv y texto plano, compatible con mongodb, mysql y postgres, permite reintentos y establecer el tiempo de cada reintento. - **Nextract [14]->** Compatible con Postgres, MySQL, MSSQL(Microsoft SQL Server), MariaDB, Oracle y más. Funciona bien para CSV y JSON. Fácil de usar. Permite ordenar y filtrar los datos y realizar operaciones matemáticas sobre ellos. - **Datapumps [15]->** Sirve para transformación y encapsulación de datos, proporciona manejo y depuración de errores. Permite crear grupos de datos a exportar. Lee CSV y bases de datos. - **Empujar [16]->** Puede correr procesos en paralelo hasta el límite que le establezcas. MySQL, FTP, S3, Amazon Redshift. - **Extraload ->** Permite archivos XML y CSV.

Tabla 6: Librerías ETL para NodeJS

Ventajas
- **Fáciles de usar:** Son librerías sencillas en su mayoría dado que cada una aporta una funcionalidad del ETL. - **Uso de Javascript:** Al tener experiencia previa con Javascript se facilita el uso de estas librerías. - **Variedad de librerías:** Al ser Javascript un lenguaje muy extendido, existen muchas implementaciones hechas por usuarios.

Tabla 7: Ventajas de las librerías ETL para NodeJS

Desventajas
- **Librerías incompletas:** Son librerías hechas por usuarios y están en desarrollo o les falta documentación. - **Conectividad con empresas:** Las aplicaciones de grandes empresas no pueden conectar con Node.js Open Source ETL Tools por razones de incompatibilidad. Esto puede dar problemas con el servicio de base de datos elegido. - **Poca gestión de errores:** Al ser Open Source, la mayoría de estas librerías no tienen gestión de errores. - **Lotes pequeños:** Los lotes en los que se procesan los datos tienen poco tamaño. - **No sirven para transformaciones complejas:** Hay falta de soporte para realizar transformaciones complejas de los datos. Las librerías sirven cuando las transformaciones son sencillas. - **Falta de soporte al cliente:** Al ser herramientas Open Source no hay empresas que les den soporte. - **Características de seguridad pobres:** Son librerías con estructura pobre y propensas a ataques informáticos.

Tabla 8: Desventajas de las librerías ETL para NodeJS

Python

Librerías
- **Petl (Python ETL) [12]->** Simple, permite fuentes de CSV, XML, JSON, XLS, etc. No hace ningún tipo de análisis de datos y tiene problemas con grandes cantidades de datos. Es un subset de Pandas más enfocado a ETL. - **Pandas [17]->** Sencillo. Tiene problemas con grandes cantidades de datos. Permite gráficas y análisis de datos. - **BeautifulSoup [18]->** Para procesar HTMLs y JSON - **Apache airflow [19]->** Permite monitorizar progreso y ejecutar tareas periódicas

Tabla 9: Librerías ETL para Python

Ventajas
- **Fáciles de usar:** Son librerías sencillas en su mayoría. - **Uso de Python:** Python es un lenguaje con una curva de aprendizaje muy baja. - **Análisis de datos:** Tienen herramientas visuales para el análisis de los datos.

Tabla 10: Ventajas de las librerías ETL para Python

Desventajas
- **No leen HTML:** Hay que transformar los datos - **Lotes pequeños:** Los lotes en los que se procesan los datos tienen poco tamaño. - **No sirven para transformaciones complejas:** Hay falta de soporte para realizar transformaciones complejas de los datos. Las librerías sirven cuando las transformaciones son sencillas. - **Falta de soporte al cliente:** Al ser herramientas Open Source no hay empresas que les den soporte. - **Características de seguridad pobres:** Son librerías con estructura pobre y propensas a ataques informáticos.

Tabla 11: Desventajas de las librerías ETL para Python

SQL Server Integration Services (SSIS) [20] y similares

Permite configurar un proceso ETL de forma gráfica. No sirve para este proyecto ya que se necesita mucha personalización para solucionar los problemas de heterogeneidad y SSIS está pensado para procesar documentos con datos homogéneos.

Spring Batch

Ventajas
- **Fácil de usar:** Puede resultar complicado la primera vez que lo utilizas, pero una vez entendido es muy cómodo de usar. - **Uso de Java:** Al tener experiencia previa con Javascript se facilita el uso de Spring Batch. - **Muchas opciones de configuración:** Permite configurar el tamaño de los lotes, añadir la cantidad de pasos que se desee, añadir procesos intermedios que no siguen la estructura extracción-transformación-carga, paralelización de procesos, logging de eventos, inicio, detención y reanudación de procesos, etc. - **Mucha documentación** - **Muy modularizado:** Spring Batch tiene una estructura muy bien definida que permite identificar bien cada uno de los componentes.

Tabla 12: Ventajas de Spring Batch

Desventajas
- **No lee HTML:** Hay que crear un módulo lector personalizado para poder leer los documentos HTML. - **Aprendizaje costoso:** Pese a que una vez entendido es fácil de implementar, los primeros pasos pueden resultar un poco toscos a la hora de entender su funcionamiento.

Tabla 13: Desventajas de Spring Batch

En definitiva, la cantidad de herramientas que ofrece Spring Batch sin necesidad de combinar numerosas librerías y su amplio uso en el mundo de los procesos ETL hacen que exista mucha documentación sobre su uso.

6. Anexo II: Testing y validación

En este anexo se detallan las pruebas realizadas para comprobar el correcto funcionamiento del proceso de extracción, así como para tomar decisiones en su desarrollo.

6.1 **Cálculo de porcentajes de similitud**

Como se ha mencionado durante el documento se ha implementado un proceso de integración de los datos en el que se debían comparar dos cadenas de texto para verificar si una referencia se encuentra en la base de datos. Para ello se ha implementado la distancia de Levenshtein, que indica si dos cadenas de texto son similares dado un porcentaje mínimo. Si el porcentaje es mayor o igual que el porcentaje mínimo, se asume que las cadenas son similares. Esto tiene un problema si no se establece un porcentaje mínimo a conciencia, ya que puede resultar en falsos positivos o falsos negativos. Se ha hecho un análisis con las referencias encontradas en las guías de cultivos.

Porcentaje mínimo	Falsos positivos	Total positivos	Porcentaje de falsos positivos
0,62	41	290	14,1%
0,65	15	253	6%
0,7	4	231	2%

Tabla 14: Resultados del análisis de porcentaje de similitud entre cadenas de texto

Si bien la última opción tiene un porcentaje muy bajo de falsos positivos, genera 11 falsos negativos, por lo que la configuración elegida ha sido la de 0’65, ya que es la más balanceada. También se ha comparado la cantidad de relaciones encontradas con y sin la distancia de Levenshtein:

	Cadenas de texto estrictamente iguales (sólo nombres)	Cadenas de texto estrictamente iguales (nombres y sinonimias)	Con distancia Levenshtein (nombres y sinonimias)
Cadenas similares	11	102	253

Tabla 15: Resultados del análisis comparativo del uso de la distancia de Levenshtein

Lo que nos permite pasar de 11 relaciones a 253, sólo con la guía de cultivos.

6.2 **Análisis de rendimiento**

Un proceso ETL tiene un tiempo de ejecución elevado ya que procesa miles de documentos y hace conexiones con una base de datos. Como se ha mencionado anteriormente, el proceso se ha paralelizado en los módulos de procesamiento y escritura para reducir el tiempo de ejecución. Se ha hecho una prueba de rendimiento midiendo el tiempo que tarda el proceso en ejecutar los tres primeros pasos, que son los que cuentan con paralelización.

Cada paso tiene el módulo lector que es secuencial, ya que Spring Batch no permite paralelizarlo. El proceso integrador no se ha tenido en cuenta para este análisis ya que es un proceso dependiente, por lo que se ejecuta únicamente al terminar los anteriores. Hay que tener en cuenta que Spring Batch es procesamiento por lotes, por lo que dependiendo del tamaño de los lotes, los tiempos varían.

Tamaño de lote	Workers	Tiempo	Porcentaje de mejora
1	1	1 min 56 s	0%
1	4	1 min 56 s	0%
20	1	53 s	52,3%
20	4	39 s	66,4%
5000	1	47 s	59,5%
5000	4	27 s	76.7%

Tabla 16: Comparativa de rendimiento

El hecho de que el tamaño de lote sea 1 hace que el procesamiento sea secuencial, independientemente de que el procesador y escritor sean asíncronos ya que se está obligando a procesar un solo documento a la vez. Aumentando el tamaño de lote se mejora el rendimiento ya que el lector, el procesador y el escritor pueden estar trabajando al mismo tiempo. Si a esto se le añade la pool de workers la mejora de rendimiento es aún mayor. Se ha conseguido una mejora de hasta el 76,7% gracias a la paralelización, principalmente, del módulo procesador, que tiene una alta carga de trabajo.

6.3 **Cálculo de porcentaje de éxito**

Durante el desarrollo del sistema de extracción se ha empleado el sistema de logging para ver los documentos que se procesaban con éxito. Al principio era un porcentaje bajo de documentos procesados con éxito dado que no se cubrían todos los casos de heterogeneidad de los documentos. Sin embargo, se ha mejorado el proceso proporcionando los siguientes resultados:

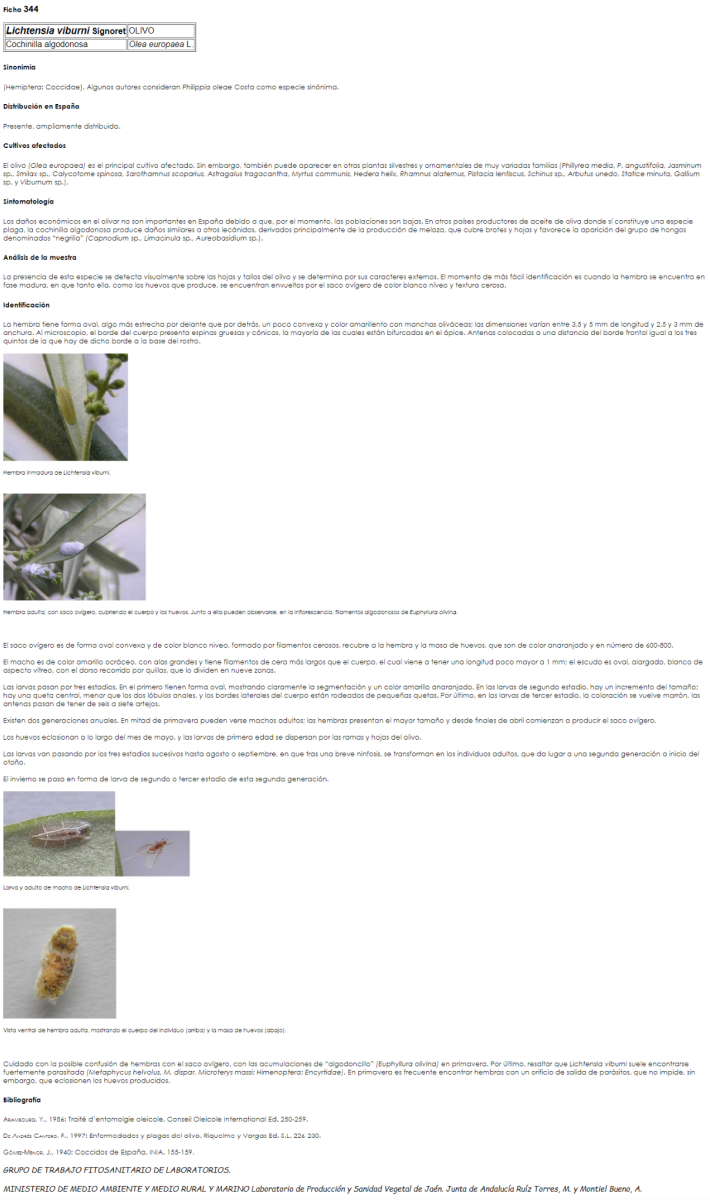
Colección	Nº de documentos	Documentos procesados con éxito	Porcentaje de éxito
Plagas	394	393	99%
Cultivos	34	33	97%
Productos fitosanitarios	2051	1974	96%

Tabla 17: Porcentaje de éxito en la extracción de documentos

Se entiende como éxito el que se haya podido leer correctamente su identificador y varios campos básicos como la ficha, el cultivo al que pertenece una plaga, las plagas relacionadas a un cultivo o las fechas de inscripción y caducidad de los productos fitosanitarios.

7. Anexo III: Estructura de los datos

En esta sección se listan algunos de los campos más comunes de los documentos de cada colección.



Ficha 344

Lichtensia viburni **Signoret**	OLIVO
Cochinilla algodonosa	*Olea europaea* L.

Sinonimia

(Hemiptera: Coccidae). Algunos autores consideran Philippia oleae Costa como especie sinónima.

Distribución en España

Presente, ampliamente distribuido.

Cultivos afectados

El olivo (*Olea europaea*) es el principal cultivo afectado. Sin embargo, también puede aparecer en otras plantas silvestres y ornamentales de muy variadas familias (*Phillyrea media, P. angustifolia, Jasminum* sp., *Smilax* sp., *Calycotome spinosa, Sarothamnus scoparius, Astragalus tragacantha, Myrtus communis, Hedera helix, Rhamnus alaternus, Pistacia lentiscus, Schinus* sp., *Arbutus unedo, Stafice minuta, Gallium* sp. y *Viburnum* sp.).

Sintomatología

Los daños económicos en el olivar no son importantes en España debido a que, por el momento, las poblaciones son bajas. En otros países productores de aceite de oliva donde sí constituye una especie plaga, la cochinilla algodonosa produce daños similares a otros lecánidos, derivados principalmente de la producción de melaza, que cubre brotes y hojas y favorece la aparición del grupo de hongos denominados "negrilla" (*Capnodium* sp., *Limacinula* sp., *Aureobasidium* sp.).

Análisis de la muestra

La presencia de esta especie se detecta visualmente sobre las hojas y tallos del olivo y se determina por sus caracteres externos. El momento de más fácil identificación es cuando la hembra se encuentra en fase madura, en que tanto ella, como los huevos que produce, se encuentran envueltos por el saco ovígero de color blanco níveo y textura cerosa.

Identificación

La hembra tiene forma oval, algo más estrecha por delante que por detrás, un poco convexa y color amarillento con manchas oliváceas; las dimensiones varían entre 3,5 y 5 mm de longitud y 2,5 y 3 mm de anchura. Al microscopio, el borde del cuerpo presenta espinas gruesas y cónicas, la mayoría de las cuales están bifurcadas en el ápice. Antenas colocadas a una distancia del borde frontal igual a las tres quintas de la que hay de dicho borde a la base del rostro.

Hembra inmadura de *Lichtensia viburni*.

Hembra adulta, con saco ovígero, cubriendo el cuerpo y los huevos. Junto a ella pueden observarse, en la inflorescencia, filamentos algodonosos de *Euphyllura olivina*.

El saco ovígero es de forma oval convexa y de color blanco níveo, formado por filamentos cerosos, recubre a la hembra y la masa de huevos, que son de color anaranjado y en número de 600-800.

El macho es de color amarillo ocráceo, con alas grandes y tiene filamentos de cera más largos que el cuerpo, el cual viene a tener una longitud poco mayor a 1 mm; el escudo es oval, alargado, blanco de aspecto vítreo, con el dorso recorrido por quillas, que lo dividen en nueve zonas.

Las larvas pasan por tres estadios. En el primero tienen forma oval, mostrando claramente la segmentación y un color amarillo anaranjado. En las larvas de segundo estadio, hay un incremento del tamaño; hay una queta central, menor que los dos lóbulos anales, y los bordes laterales del cuerpo están rodeadas de pequeñas quetas. Por último, en las larvas de tercer estadio, la coloración se vuelve marrón, las antenas pasan de tener de seis a siete artejos.

Existen dos generaciones anuales. En mitad de primavera pueden verse machos adultos; las hembras presentan el mayor tamaño y desde finales de abril comienzan a producir el saco ovígero.

Los huevos eclosionan a lo largo del mes de mayo, y las larvas de primera edad se dispersan por las ramas y hojas del olivo.

Las larvas van pasando por los tres estadios sucesivos hasta agosto o septiembre, en que tras una breve ninfosis, se transforman en los individuos adultos, que da lugar a una segunda generación a inicio del otoño.

El invierno se pasa en forma de larva de segundo o tercer estadio de esta segunda generación.

Larva y adulto de macho de *Lichtensia viburni*.

Vista ventral de hembra adulta, mostrando el cuerpo del individuo (arriba) y la masa de huevos (abajo).

Cuidado con la posible confusión de hembras con el saco ovígero, con las acumulaciones de "algodoncillo" (*Euphyllura olivina*) en primavera. Por último, resaltar que *Lichtensia viburni* suele encontrarse fuertemente parasitada (*Metaphycus helvolus, M. dispar, Microterys mazi*; Himenoptera: Encyrtidae). En primavera es frecuente encontrar hembras con un orificio de salida de parásitos, que no impide, sin embargo, que eclosionen los huevos producidos.

Bibliografía

Arambourg, Y., 1986: Traité d'entomologie oléicole. Conseil Oléicole International Ed. 250-259.

De Liñán Cabello, R., 1997: Entomofauna y plagas del olivo. Riquelme y Vargas Ed. S.L. 226 230.

Gómez-Menor, J., 1940: Coccidos de España. INIA. 155-159.

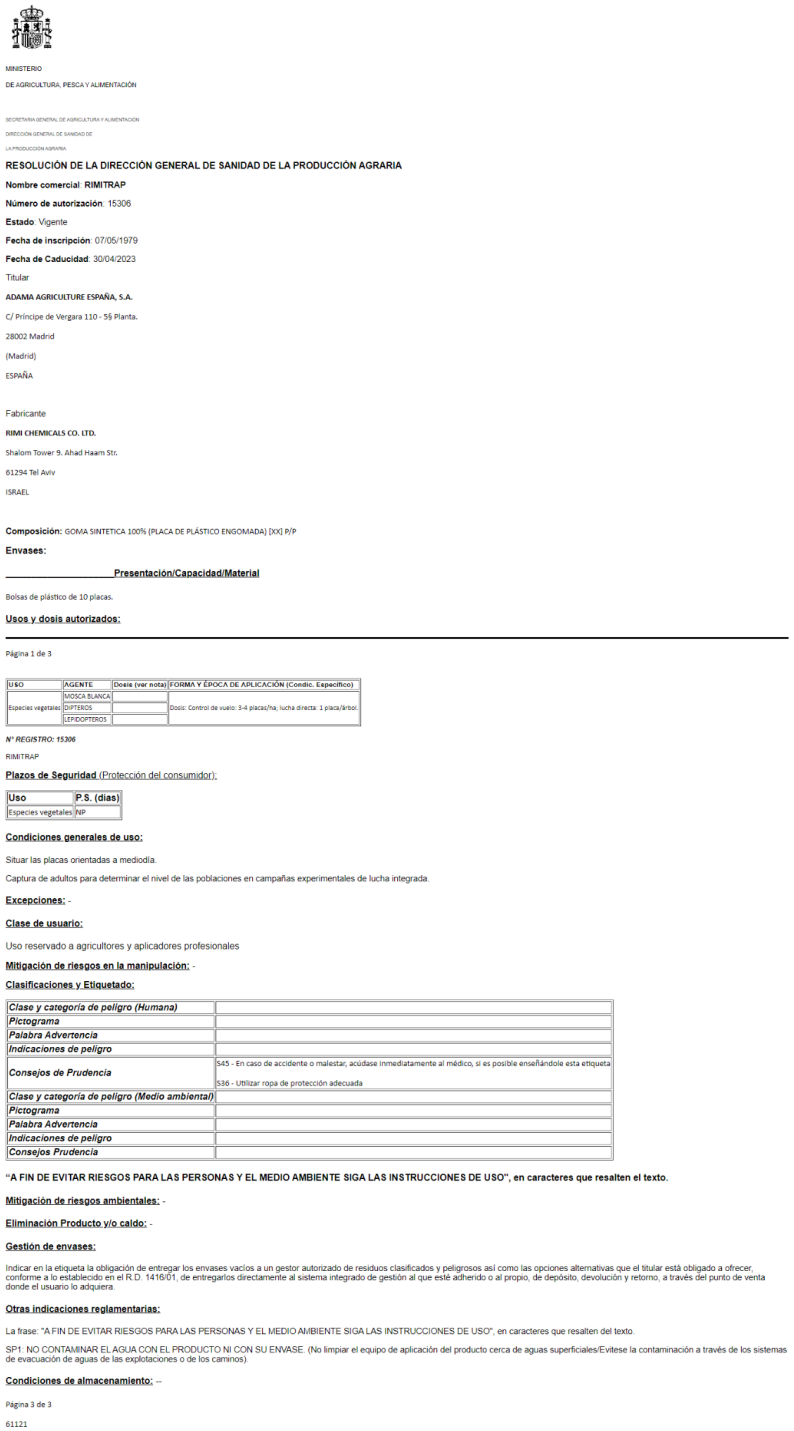
GRUPO DE TRABAJO FITOSANITARIO DE LABORATORIOS.

MINISTERIO DE MEDIO AMBIENTE Y MEDIO RURAL Y MARINO Laboratorio de Producción y Sanidad Vegetal de Jaén. Junta de Andalucía Ruiz Torres, M. y Montiel Bueno, A.

Figura 16: Guía de plagas (ficha 344)

Las guías de plagas cuentan con el número de ficha al inicio de la página (figura 16). Seguidamente aparece el nombre científico de la plaga y el cultivo al que pertenece. Esta información puede aparecer representada en forma de tabla o en texto plano. El cultivo es el dato que se usa para relacionarlo con la colección de cultivos. A continuación hay una serie de campos alternados con imágenes que tienen (normalmente) el título en negrita y el texto normal. Los campos más comunes son el de sinonimia (utilizado en el proceso de integración), la distribución en España, los cultivos afectados (también usado para

relacionarlos), la sintomatología y la identificación (la descripción de la plaga). Al final aparecen la bibliografía y algunas referencias. El resto de campos aparecen menos frecuentemente o vienen de las guías de cultivos.



MINISTERIO
DE AGRICULTURA, PESCA Y ALIMENTACIÓN

SECRETARÍA GENERAL DE AGRICULTURA Y ALIMENTACIÓN
DIRECCIÓN GENERAL DE SANIDAD DE
LA PRODUCCIÓN AGRARIA

RESOLUCIÓN DE LA DIRECCIÓN GENERAL DE SANIDAD DE LA PRODUCCIÓN AGRARIA

Nombre comercial: **RIMITRAP**

Número de autorización: 15306

Estado: Vigente

Fecha de inscripción: 07/05/1979

Fecha de Caducidad: 30/04/2023

Titular

ADAMA AGRICULTURE ESPAÑA, S.A.

C/ Príncipe de Vergara 110 - 5§ Planta.

28002 Madrid

(Madrid)

ESPAÑA

Fabricante

RIMI CHEMICALS CO. LTD.

Shalom Tower 9. Ahad Haam Str.

61294 Tel Aviv

ISRAEL

Composición: GOMA SINTETICA 100% (PLACA DE PLÁSTICO ENGOMADA) [XX] P/P

Envases:

Presentación/Capacidad/Material

Bolsas de plástico de 10 placas.

Usos y dosis autorizados:

Página 1 de 3

USO	AGENTE	Dosis (ver nota)	FORMA Y ÉPOCA DE APLICACIÓN (Condic. Específico)
Especies vegetales	MOSCA BLANCA		Dosis: Control de vuelo: 3-4 placas/ha; lucha directa: 1 placa/árbol.
	DIPTEROS		
	LEPIDOPTEROS		

Nº REGISTRO: 15306

RIMITRAP

Plazos de Seguridad (Protección del consumidor):

Uso	P.S. (días)
Especies vegetales	NP

Condiciones generales de uso:

Situar las placas orientadas a mediodía.

Captura de adultos para determinar el nivel de las poblaciones en campañas experimentales de lucha integrada.

Excepciones: -

Clase de usuario:

Uso reservado a agricultores y aplicadores profesionales

Mitigación de riesgos en la manipulación: -

Clasificaciones y Etiquetado:

Clase y categoría de peligro (Humana)	
Pictograma	
Palabra Advertencia	
Indicaciones de peligro	
Consejos de Prudencia	S45 - En caso de accidente o malestar, acúdase inmediatamente al médico, si es posible enseñándole esta etiqueta S36 - Utilizar ropa de protección adecuada
Clase y categoría de peligro (Medio ambiental)	
Pictograma	
Palabra Advertencia	
Indicaciones de peligro	
Consejos Prudencia	

"A FIN DE EVITAR RIESGOS PARA LAS PERSONAS Y EL MEDIO AMBIENTE SIGA LAS INSTRUCCIONES DE USO", en caracteres que resalten el texto.

Mitigación de riesgos ambientales: -

Eliminación Producto y/o caldo: -

Gestión de envases:

Indicar en la etiqueta la obligación de entregar los envases vacíos a un gestor autorizado de residuos clasificados y peligrosos así como las opciones alternativas que el titular está obligado a ofrecer, conforme a lo establecido en el R.D. 1416/01, de entregarlos directamente al sistema integrado de gestión al que esté adherido o al propio, de depósito, devolución y retorno, a través del punto de venta donde el usuario lo adquiera.

Otras indicaciones reglamentarias:

La frase: "A FIN DE EVITAR RIESGOS PARA LAS PERSONAS Y EL MEDIO AMBIENTE SIGA LAS INSTRUCCIONES DE USO", en caracteres que resalten del texto.

SP1: NO CONTAMINAR EL AGUA CON EL PRODUCTO NI CON SU ENVASE. (No limpiar el equipo de aplicación del producto cerca de aguas superficiales/Evítese la contaminación a través de los sistemas de evacuación de aguas de las explotaciones o de los caminos).

Condiciones de almacenamiento: --

Página 3 de 3

61121

Figura 17: Guía de productos fitosanitarios (nº 15306)

Las guías de productos sanitarios (Figura 17) no contienen más que una imagen inicial, la cual es común para todos los documentos de la colección. Los campos vienen separados

con títulos en negrita y/o subrayados. En la parte superior, el contenido de los campos no va acompañado de un salto de línea, mientras que en el resto del documento sí, dificultando así la extracción de los datos. Los campos más importantes son el nombre comercial (identificador del documento), el número de autorización, el estado, la fecha de inscripción, la fecha de caducidad y los usos y dosis (usados para relacionar los productos con las plagas y cultivos). Este último campo se presenta en formato de tabla. La mayoría de los campos de estas guías suelen aparecer siempre, por lo que están bastante completas.

Plagas principales	Seguimiento y estimación del riesgo para el cultivo	Medidas de prevención y/o culturales	Umbral/Momento de intervención	Medidas alternativas al control químico (*)	Medios químicos
Pammene fasciana **(TORTRÍCIDO PRECOZ DE LA CASTAÑA)**	Utilización de trampas de feromonas sexuales Complementar esta medida con la detección visual de síntomas sobre los erizos y el examen de los mismos para la detección de orugas	Colocación de bandas de cartón ondulado rodeando la base y las ramas más gruesas del castaño para capturar las larvas que se retiran a invernar, eliminándolas antes de la primavera	No existe un umbral de actuación establecido, la sola detección de la plaga debería ser suficiente para eliminar orugas y evitar daños al fruto	**Medios biológicos** Aunque hay diversos parasitoides de las larvas de *Pammene fasciana*, de momento no se han realizado experiencias de control	Se podrán utilizar, en el caso de que existan, los productos fitosanitarios autorizados para este uso en el Registro de Productos Fitosanitarios del Ministerio de Agricultura, Pesca y Alimentación
Cydia fagiglandana, Cydia splendana **(TORTRÍCIDOS INTERMEDIO Y TARDÍO DE LA CASTAÑA)**	Colocación de trampas de feromonas (1 trampa/1 ha) durante la época de vuelo, a partir de mayo para *C. fagiglandana* y de junio-julio para *C. splendana*	Instalación de mallas en el terreno para evitar que las larvas se entierren Recolección manual de erizos caídos para disminuir la población	No existe un umbral de actuación establecido, sin embargo, al menos para *C. splendana* debería actuarse antes de que las orugas penetren en el fruto	**Medios biológicos** Se está estudiando el uso de nematodos entomopatógenos (NEP's)	Se podrán utilizar, en el caso de que existan, los productos fitosanitarios autorizados para este uso en el Registro de Productos Fitosanitarios del Ministerio de Agricultura, Pesca y Alimentación
Zeuzera pyrina **(TALADRO AMARILLO O ZEUZERA)**	Instalar trampas cebadas con feromona a partir de mayo Con presencia de plaga, realizar un control visual en invierno al 20% de los árboles buscando orificios de entrada o detritus anaranjado	En pequeños ataques eliminación manual de las larvas Poda o destrucción de los árboles afectados	2% de árboles afectados, revisar al menos un centenar de árboles por parcela	**Medios biológicos** Los depredadores o parasitoides ejercen un grado de control insuficiente, igualmente sucede con nematodo entomopatógenos **Medios biotecnológicos** Uso de feromonas de confusión sexual colocadas en la parcela a una densidad de 10 trampas/ha poco antes del inicio del vuelo	Pulverización del tronco y ramas a final de la primavera, desde la eclosión de huevos y antes de que las larvas penetren en el interior de la madera Se podrán utilizar, en el caso de que existan, los productos fitosanitarios autorizados para este uso en el Registro de Productos Fitosanitarios del Ministerio de Agricultura, Pesca y Alimentación

(*) En este apartado se han recogido los medios biológicos, biotecnológicos y físicos. Los medios culturales, que también pueden ser alternativos al control químico, se han agrupado con las medidas de prevención.

CUADRO DE ESTRATEGIA DE GESTIÓN INTEGRADA DE PLAGAS

Figura 18: Guía de cultivos (Castaño)

En la figura 18 se muestra sólo una sección de una guía de cultivos, ya que el documento es muy extenso como para mostrarlo entero. La información de los cultivos es el nombre (identificador), el cuadro de gestión integrada de plagas (contiene información extra sobre las plagas y sirve para relacionar el cultivo con la colección de plagas) y anexos sobre plagas (más información de las plagas). La tabla mostrada siempre tiene los mismos campos, lo cual facilita su procesamiento. Los anexos de plagas siguen la misma estructura que las guías de plagas (figura).

8. Anexo IV: Diseño inicial de la GUI

A continuación se muestra el diseño inicial de la interfaz gráfica de la aplicación:

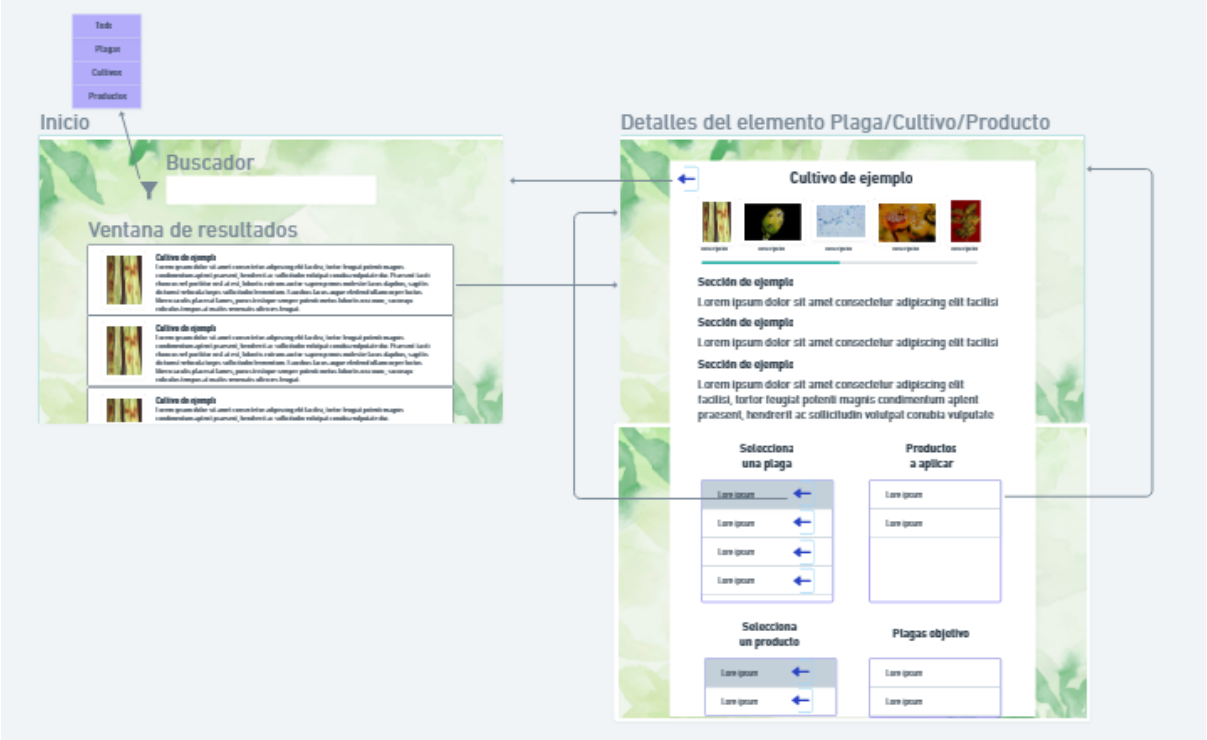


Figura 19: Diseño inicial de la GUI

Se trata de un diseño minimalista de dos ventanas. La ventana principal cuenta con un buscador y un listado de los documentos. La búsqueda es de los nombres y se puede filtrar según la colección deseada. Se puede navegar hacia la pantalla de detalles haciendo clic en un elemento de la lista. En esta pantalla se muestran todos los campos del documento seleccionado, con variaciones dependiendo de la colección a la que pertenece. Destaca la navegación, ya que se puede volver a la pantalla principal o se puede acceder a la pantalla de detalles de otro documento que esté relacionado.