Universidad Zaragoza

Trabajo Fin de Grado

Sistema de información integrada de plagas

Pest integrated information system

Autor

Eduardo Gimeno Soriano

Director

Javier Lacasta Miguel

Ponente

Francisco Javier Zarazaga Soria

Escuela de Ingeniería y Arquitectura

2020/2021

Sistema de información integrada de plagas

Resumen

Para facilitar la gestión de plagas en cultivos, el Ministerio de Agricultura, Ganadería y Pesca proporciona múltiples documentos describiendo los diferentes aspectos de las plagas y sus posibles tratamientos. Sin embargo, dicha información, en vez de proporcionarse de forma interoperable, está descrita principalmente en ficheros PDF, cada uno centrado en un aspecto concreto de la gestión de plagas.

Como primer paso para facilitar la integración de toda la información disponible en un sistema de información moderno, este trabajo de fin de grado se centra en el desarrollo de un sistema de extracción de la información contenida en un subconjunto de la colección de documentos PDF disponibles (texto e imágenes) y su almacenamiento en una base de datos. La complejidad del proceso de extracción viene marcada por la heterogeneidad y problemas de calidad de los documentos tratados. Como forma de validación de la información extraída, se desarrolla una aplicación web que presenta la información guardada en la base de datos de forma sencilla y rápidamente accesible a través de una serie de parámetros de búsqueda.

Dada la complejidad de la extracción de información de los PDFs, este paso requiere una investigación previa en herramientas y métodos existentes y su comparación para la elección del conjunto más apropiado.

La implementación de la aplicación web se realiza con el stack MERN (MongoDB, Express, React y NodeJS), ampliamente utilizado actualmente.

ÍNDICE

AGRADECIMIENTOS

Quiero agradecer a los miembros del grupo de investigación IAAA, Francisco Javier Zarazaga Soria, Javier Lacasta y Jorge Pinilla acompañarme en este trabajo de fin de grado. El cual ha supuesto un gran desafío y esfuerzo para culminar esta etapa en la Universidad de Zaragoza.

1 INTRODUCCIÓN

1.1 Contexto del Trabajo

Las plagas en los cultivos producen pérdidas económicas importantes en todo el mundo. Para tratar con ellas sin dañar a las personas y al medio ambiente, los gobiernos han establecido leyes y normas estrictas que describen los productos y procedimientos de uso. Sin embargo, dado que estas normas cambian con frecuencia para reflejar los avances tecnológicos y científicos, es necesario realizar una revisión frecuente de las normas afectadas. Esta tarea no es fácil porque generalmente están orientadas al ser humano, por lo que se requiere un trabajo manual intensivo.

El Ministerio de Agricultura, Ganadería y Pesca [1] ofrece múltiples documentos para la gestión integrada de las plagas presentes en el territorio español, describiendo distintos aspectos de las mismas y cómo tratarlas. Esta información, en lugar de proporcionarse de una forma interoperable, se proporciona mediante una gran cantidad de documentos PDF, cada uno centrado en un aspecto concreto.

Estos documentos están divididos en distintas colecciones. Por ejemplo, las guías de gestión integrada de plagas [2]. Estas guías, se centran cada una en un tipo de cultivo presente en España y describen las distintas plagas que le afectan. Se caracterizan por ser documentos muy largos, del orden de más de 100 páginas. Otra colección ofrecida sería una serie de fichas [3], tratando cada una de una plaga en concreto y dando información sobre su identificación, síntomas, cultivos afectados, etc. Se caracterizan por ser cortas, del orden de 1 o 2 páginas. Otra fuente disponible, es el registro de productos fitosanitarios [4], la cual ofrece documentos con información referente a productos de este tipo permitidos o su composición.

1.2 Motivación y problema que se aborda

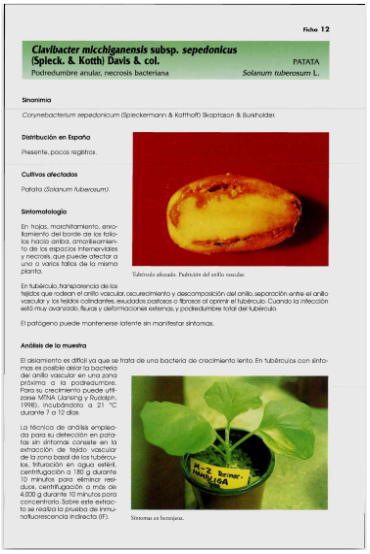
Para un agricultor puede resultar complicado encontrar toda la información necesaria sobre qué plagas afectan a sus cultivos, resultando costoso, en tiempo, encontrar dicha información, pudiendo esto provocar diversos problemas a sus explotaciones.

Para gestionar y facilitar el uso de la información disponible es necesario un sistema de información moderno que integre las colecciones existentes, permita una búsqueda unificada y simplifique la presentación de los datos.

Un primer paso en el desarrollo de este sistema y el objetivo de este proyecto es desarrollar un sistema automático de extracción, limpieza e integración de información

existente en el Ministerio de Agricultura, Ganadería y Pesca. A su vez, se ha creado una aplicación web que permite validar los resultados de este sistema de extracción.

Como fuente de información a integrar, se ha seleccionado la colección de fichas [3] mencionada en el apartado 1.1. Esta colección contiene aproximadamente 400 fichas en formato PDF, centradas en una plaga en concreto. Tiene su contenido dividido en diferentes secciones, las cuales pueden variar en cierta medida de un documento a otro. En primer lugar se encuentra un recuadro con los nombres científicos y comunes de la plaga sobre la que trata el documento y el/los cultivo/s que se ven afectados por ella. El contenido se divide en las secciones Sinonimia, Distribución en España, Cultivos afectados, Sintomatología, Análisis de la muestra, Transmisión, Identificación y Bibliografía. Además contiene imágenes sobre la plaga de la que trata y cada una va acompañada de una descripción. Como punto importante, cabe destacar que una gran cantidad de estas fichas son documentos PDF generados a partir de elementos escaneados (figura 1), frente al resto que son generados a partir de un documento de texto (figura 2). Esta diferencia entre los documentos de la colección implica que no se pueden procesar con las mismas técnicas según se ha visto en el proceso previo de investigación descrito en el Anexo I. Esto requiere distinguirlos para aplicar determinadas acciones según sean de qué tipo sean dentro del sistema automático de extracción.


Ficha 12

Clavibacter michiganensis subsp. sepedonicus (Spieck. & Koth) Davis & col.

PATATA

Podredumbre anular, necrosis bacteriana — *Solanum tuberosum* L.

Sinonimia

Corynebacterium sepedonicum (Spieckermann & Kotthoff) Skaptason & Burkholder.

Distribución en España

Presente, pocos registros.

Cultivos afectados

Patata (*Solanum tuberosum*).

Sintomatología

En hojas, marchitamiento, enrollamiento del borde de los folíolos hacia arriba, amarilleamiento de los espacios internerviales y necrosis, que puede afectar a uno o varios folíos de la misma planta.

Tubérculo afectado. Podrición del anillo vascular.

En tubérculo, transparencia de los tejidos que rodean el anillo vascular, oscurecimiento y descomposición del anillo, separación entre el anillo vascular y los tejidos colindantes, exudados pastosos o floroce al oprimir el tubérculo. Cuando la infección está muy avanzada, fisuras y deformaciones externas, y podredumbre total del tubérculo.

El patógeno puede mantenerse latente sin manifestar síntomas.

Análisis de la muestra

El aislamiento es difícil ya que se trata de una bacteria de crecimiento lento. En tubérculos con síntomas es posible aislar la bacteria del anillo vascular en una zona próximo o la podredumbre. Para su crecimiento puede utilizarse MTNA (Lansing y Rudolph, 1998), incubándolos a 21 °C durante 7 o 12 días.

La técnica de análisis empleada para su detección en patata sin síntomas consiste en la extracción de tejido vascular de la zona basal de los tubérculos, trituración en agua estéril, centrifugación a 180 g durante 10 minutos para eliminar residuos, centrifugación a más de 4.000 g durante 10 minutos para concentrarlo. Sobre este extracto se realiza la prueba de inmunofluorescencia indirecta (IF).

Síntomas en hortaliza.


Ficha 371

***Naohidemyces vacciniorum* (Schröt.) Spooner** — ERICÁCEAS

Roya del Arándano — *Ericaceae*

Sinonimia

Melampsora vaccinii Winter, *Thekopsora vaccinii* (Jørst.) Hirats., *Pucciniastrum vaccinii* (Winter) Jørst., *Naohidemyces vaccinii* (Jørst.) Sato, Katsuya & Hirats.

Distribución en España

Presente, pocos registros.

Cultivos afectados

Afecta a especies de varios géneros dentro de la familia *Ericaceae*, siendo los más importantes *Rhododendron* y *Vaccinium*.

Sintomatología

Los primeros síntomas son manchas cloróticas en las hojas más viejas, esas manchas se tornan pardo rojizas en el haz de la hoja, y si el ataque es importante, las manchas fienden a unirse y se vuelven necróticas, lo que hace que las hojas se sequen y finalmente caigan. En el envés de la hoja, debajo de estas manchas, aparecen los uredinios: son pústulas pulverulentas de color anaranjado principalmente, aunque pueden ser amarillentas.

En general es una enfermedad de poca importancia, con escaso efecto en el rendimiento. Las hojas infectadas se caen prematuramente, pero sólo si la defoliación es severa se podría ver afectado el cultivo.

Síntomas en el haz. Síntomas en el envés. Detalle de las pústulas.

Análisis de la muestra

Al ser un parásito estricto, las observaciones se hacen directamente sobre el material vegetal. Se realizan montajes de las pústulas del envés y se observan al microscopio.

Figura 1: Ejemplo de ficha generada a partir de elementos escaneados

Figura 2: Ejemplo de ficha generada a partir de de un documento de texto

1.3 Alcance, objetivos y limitaciones

El objetivo principal de este proyecto ha sido desarrollar el sistema automático de extracción, limpieza e integración de la colección de documentos seleccionada. A su vez, con el objetivo de validar este sistema, se desarrolla la aplicación web que muestra la información obtenida de forma sencilla y accesible por parámetros de búsqueda.

Como primer paso para realizar el objetivo principal, se ha realizado un estudio del estado del arte en tecnologías de extracción de información de documentos PDF. El proceso de dicha investigación queda explicado en el Anexo I. Una vez realizado, se han seleccionado las que mejores resultados presentan respecto a calidad de la extracción. Como segundo paso, se ha elaborado el sistema de extracción, haciendo uso de dichas tecnologías. Cuando se ha finalizado el proceso de extracción de la colección de documentos, con el sistema creado, y los resultados se encuentran guardados en la base de datos, se ha llevado a cabo el último paso, la creación de la aplicación web para mostrar los resultados. Dicha aplicación cuenta con Back-End, el cual ofrece una serie de operaciones teniendo en cuenta el modelo de datos creado, y con Front-End, el cual hace uso de estas operaciones para poder mostrar los resultados según se desea.

La principal limitación del proyecto ha venido por la complejidad en el proceso de extracción, debido a la heterogeneidad y calidad de los documentos PDF. Como se ha explicado en el apartado anterior, algunos documentos han sido generados a partir de elementos escaneados, mientras que otros lo han sido a partir de documentos de texto, lo cual ha obligado a aplicar técnicas diferentes a cada tipo. Además algunos documentos presentan secciones diferentes a otros, presentando problemáticas como que unos documentos comienzan por una sección y otros por otra o que algunos de ellos tengan más secciones. Además, cada documento tiene sus imágenes en posiciones diferentes.

1.4 Herramientas de trabajo

Como se puede ver en la tabla 1, las tecnologías utilizadas se han seleccionado según distintos criterios. Para el sistema de extracción, se han seleccionado PDFBox y EasyOCR porque han mostrado los mejores resultados de todas las tecnologías probadas y el uso de la máquina Endurance proviene de la necesidad de tener una gran capacidad de procesamiento (Anexo II). En el caso de la base de datos y la aplicación web se han seleccionado por ser tecnologías con un amplio uso en el ámbito empresarial y por tener un conocimiento previo sobre ellas.

Proceso	Herramienta
Sistema de extracción	PDFBox, EasyOCR
Procesado de la colección de documentos	Máquina Endurance
Base de datos	MongoDB
Aplicación web	Node.js, Express, React, http.server

Tabla 1: Herramientas utilizadas en las distintas partes del proyecto

En el Anexo II se realiza una descripción más detallada de cada una de estas tecnologías.

1.5 Esquema general de la memoria del proyecto

En los siguientes apartados, se describe el proceso seguido, así como resultados y conclusiones extraídas. El apartado 2 contiene todo el trabajo desarrollado, desde los requisitos del sistema de extracción y la aplicación web (2.1) hasta los resultados obtenidos y problemas encontrados (2.3), pasando por el diseño del sistema (2.2), en el que se explica el desarrollo del sistema de extracción y la aplicación web.

El apartado 3, contiene los conocimientos adquiridos tras la realización de este proyecto (3.1), ideas futuras (3.2), donde se describen posibles añadidos que se podrían realizar en un futuro. Por último, están las conclusiones obtenidas (3.3).

Pasando a los anexos, el Anexo I contiene la investigación llevada a cabo sobre las tecnologías para la extracción de información de documentos PDF. Se describen las distintas tecnologías que se han probado, así como resultados obtenidos y cuáles se han seleccionado finalmente. En el Anexo II se describen las tecnologías que se han utilizado para el desarrollo tanto del sistema de extracción, como de la aplicación web. El modelo de datos utilizado para guardar la información extraída de los documentos PDF se describe en el Anexo III. Finalmente, en el Anexo IV se describe el logging que se lleva a cabo en el sistema de extracción para comprobar el funcionamiento del mismo.

2 TRABAJO DESARROLLADO

2.1 Requisitos del sistema

En esta sección se listan los requisitos del sistema de extracción y de la aplicación web.

2.1.1 Requisitos del sistema de extracción

El sistema de extracción presenta los siguientes requisitos:

- Debe descargar los documentos a procesar de forma automática de la fuente.
- Debe guardar los resultados de la extracción en una base de datos.
- Debe extraer los nombres científicos y comunes de la plaga sobre la que trate cada documento.
- Debe extraer los nombres científicos y comunes de los cultivos a los que afecta la plaga sobre la que trate cada documento.
- Debe extraer el contenido de las secciones contenidas en el documento: Distribución en España, Sintomatología, Análisis de la muestra, Transmisión, Identificación y Bibliografía.
- Debe extraer las imágenes contenidas en los documentos.
- Debe extraer las descripciones de las imágenes.
- Debe ser sencillo poder añadir más fuentes de información.

2.1.2 Requisitos de la aplicación web

La aplicación web presenta los siguientes requisitos:

- Debe permitir realizar búsqueda por cultivo mostrando como resultado un listado con las plagas que lo afectan.
- Debe permitir realizar búsqueda por plaga mostrando como resultado la información detallada de la misma.

2.2 Diseño del sistema

2.2.1 Sistema de extracción

Teniendo en cuenta la problemática que presenta la colección de documentos, siendo una parte de ellos generados a partir de documentos de texto y otra parte a partir de elementos escaneados, obliga a hacer una diferenciación en cómo se extrae su información. Como se describe en el Anexo I, no es posible obtener los mismos resultados aplicando las mismas técnicas.

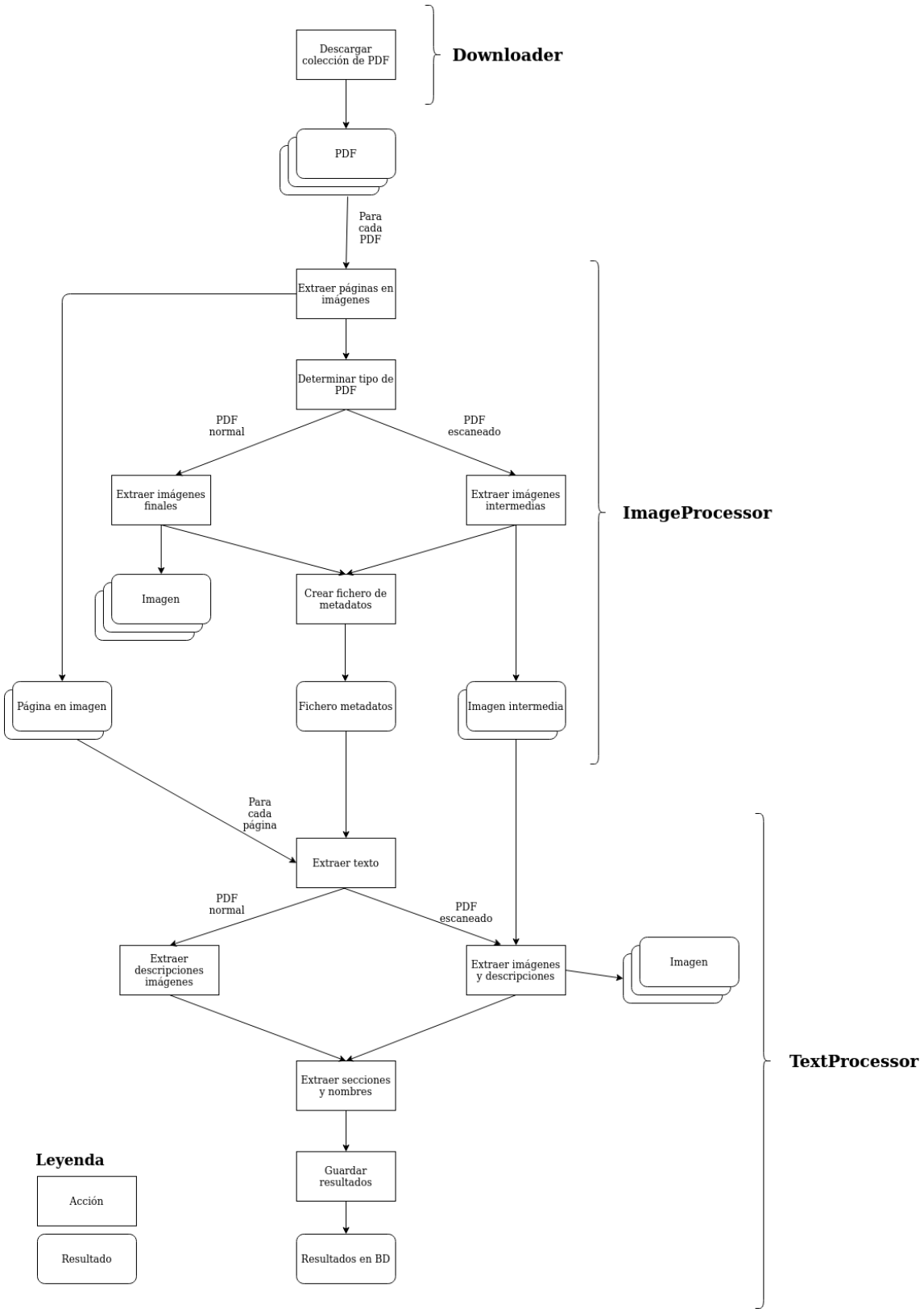


Figura 3: Esquema del sistema de extracción

En la figura 3 se describen los pasos que se siguen para obtener la información contenida en los documentos PDF. A continuación se van a describir los distintos pasos.

2.2.1.1 Descargar colección de PDF

En primer lugar, se descarga la colección de documentos. El módulo Downloader se encarga de realizarlo. Escrito con Python, debido a que resulta más sencillo y rápido de implementar, realiza una petición GET a cada una de las url para obtener los documentos. Cada documento se encuentra accesible a través de la url *https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_XXX*, siendo XXX el identificador de cada documento (001 a 399). Una vez obtenido el documento se guarda en el subdirectorio Short_Guides del directorio Data. Este directorio se encuentra un nivel por encima del sistema de extracción porque la aplicación web accede a él para obtener las guías descargadas.

2.2.1.2 Extraer páginas en imágenes

Con la colección descargada, sin diferenciar por el tipo de documento se obtiene una imagen de cada página para cada documento. Estas imágenes se usan posteriormente para obtener el texto.

Este paso se implementa en el módulo ImageProcessor, a través de la clase PageToImage. El módulo está escrito en Java porque se hace uso de la biblioteca PDFBox y de sus operaciones, es este caso, para obtener una página de un documento PDF en una imagen. PDFBox permite indicar el DPI con el que obtener la imagen, por las pruebas realizadas, descritas en el Anexo I, se ha observado que cuanta mayor calidad tiene la imagen que EasyOCR usa para obtener el texto, mejor resultados proporciona. Por tanto, es necesario indicarle a PDFBox un DPI (dots per inch) alto para generar una imagen de buena calidad. Se ha seleccionado un DPI de 990 porque es lo suficientemente alto para generar una imagen con la que EasyOCR proporciona resultados sin apenas errores y porque al superar esta cantidad de DPI indica un aviso de posible ataque DDOS. La imagen obtenida se guarda en formato PNG siguiendo la nomenclatura guía-página (ej. fd_001-1.png) para facilitar el procesamiento en el módulo TextPorcessor y se guardan en el directorio Pages dentro del sistema de extracción. Como inconveniente, al obtenerse con un DPI alto, el tamaño de la imagen también es elevado.

2.2.1.3 Determinar tipo de PDF

Para saber qué imágenes se van a extraer de un documento, es necesario determinar primero si se trata de un PDF generado o escaneado.

Se ha observado que los PDF generados presentan una versión diferente, concretamente la 1.6 (figura 5), frente a la 1.3 de los escaneados (figura 4). Esto es debido a que estos últimos (2007-2008) tienen una fecha de creación muy anterior a los primeros (2012). Teniendo esto en cuenta, basta con leer la primera línea del documento y ver si corresponde con %PDF-1.3 para determinar si se trata de un PDF escaneado o %PDF-1.6 para un PDF generado.

Figura 4: PDF escaneado

Figura 5: PDF generado

2.2.1.4 Extraer imágenes finales

Habiendo determinado que el documento es un PDF generado, se pueden extraer directamente las imágenes finales utilizando PDFBox.

Este paso se implementa en el módulo ImageProcessor, en la clase ImageExtractor. PDFBox ofrece operaciones para extraer directamente las imágenes, pero también se quiere obtener las coordenadas de los vértices de estas para poder obtener posteriormente las descripciones del texto. Para ello, es necesario extender una clase en concreto de PDFBox, PDFStreamEngine, y hacer un override de la operación processOperator. En esta operación, se pueden obtener las coordenadas del vértice superior izquierdo y calcular los restantes conociendo la anchura y altura de la imagen.

Las unidades de medida dentro de un PDF están en puntos, una unidad de medida tradicional de la industria gráfica. EasyOCR utiliza esta medida para las coordenadas de las bounding box que proporciona y varían en función del DPI con el que ha sido generada la imagen. PDFBox proporciona las coordenadas a partir de 72 DPI (estándar adoptado por Adobe), por tanto, es necesario escalar las coordenadas que proporciona PDFBox a 990 DPI. Teniendo en cuenta que 1 punto es equivalente a 1/72 pulgadas, de ahí los 72 DPI, se aplica la fórmula (coordenada*DPI)/72 en ambos ejes.

Otro problema a resolver ocurre con el eje Y. Para PDFBox el punto (0, 0) se encuentra en la esquina inferior izquierda de la página, sin embargo, para EasyOCR se encuentra en la esquina superior izquierda, como se puede ver en la figura 6. Previamente se ha obtenido la altura total de la página en puntos y se ha escalado a 990 DPI. Esto permite realizar la corrección de las coordenadas de los vértices en el eje Y simplemente restando a la altura la coordenada en dicho eje.

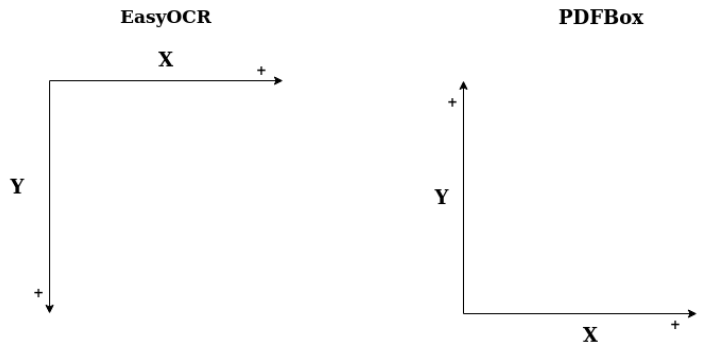


Figura 6: Distribución de los ejes en EasyOCR y PDFBox

Con estos problemas solucionados, finalmente se obtienen las coordenadas de los vértices superior izquierdo, inferior izquierdo e inferior derecho de las imágenes presentes en el documento, así como las propias imágenes, guardandolas siguiendo la nomenclatura guía-página-número (ej. fd_399-1-1) en el directorio Images, un nivel por encima del sistema de extracción porque la aplicación web accede a ellas. El número es el orden en el que aparecen en el documento de arriba a abajo de la página.

2.2.1.5 Extraer imágenes intermedias

No es posible obtener directamente las imágenes finales de los PDF escaneados en un único paso. Debido a como están creados, PDFBox no puede obtenerlas, sin embargo sí genera un producto intermedio útil del que poder obtenerlas posteriormente. La descripción detallada de este problema se encuentra en el Anexo I.

Como se puede ver en la figura 7, estas imágenes intermedias consisten en el contenido de la página sin el texto. El hecho de que el texto no esté presente facilita poder aplicar posteriormente técnicas que permiten obtener las imágenes finales que interesa guardar y que ya han podido ser obtenidas para los PDF generados. Estas imágenes intermedias se obtienen de la misma forma con PDFBox que las imágenes finales de los PDF generados, el único paso extra necesario es filtrar los resultados porque además de devolver estas imágenes, PDFBox también devuelve imágenes que contienen el texto de la página y estas últimas no interesan de cara a la extracción.

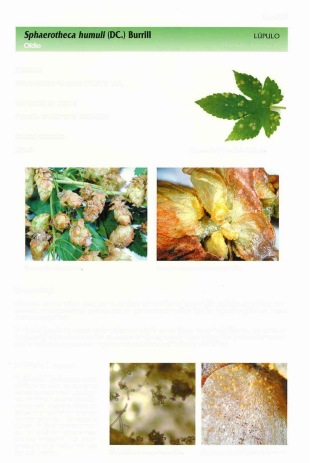


Figura 7: Imagen intermedia obtenida con PDFBox

Estas imágenes se guardan en el directorio Images, dentro del sistema de extracción, siguiendo la nomenclatura guía-página (ej. fd_001-1.jpg).

2.2.1.6 Crear fichero de metadatos

Al encontrarse el sistema de extracción dividido en dos módulos independientes, ImageProcessor implementando todo lo relacionado con PDFBox y TextProcessor implementando la parte de EasyOCR, es necesario un mecanismo de comunicación entre ambos lados. Para ello, se crea un fichero de metadatos, el cual varía dependiendo del tipo de PDF. Si se trata de un PDF escaneado presenta la estructura que se ve en la figura 8, dos líneas, la primera de ellas contiene el total de páginas y la segunda el tipo de PDF. Si se trata de un PDF generado, presenta la

estructura que se ve en la figura 9. Las dos primeras líneas son iguales, después presenta una estructura que se sigue para cada página del documento, indicativo de la página y en las líneas sucesivas las coordenadas de los vértices superior izquierdo, inferior izquierdo e inferior derecho, en ese orden, de las imágenes contenidas en dicha página.

Estos ficheros de metadatos se guardan en el directorio Metadata, dentro del sistema de extracción.

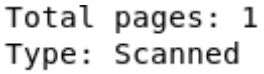


Figura 8: Fichero de metadatos de un PDF escaneado

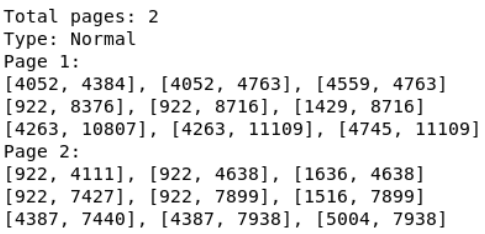


Figura 9: Fichero de metadatos de un PDF generado

2.2.1.7 Extraer texto

Primer paso que se implementa en el módulo TextProcessor. Todo el módulo está implementado en Python debido al uso de EasyOCR. La extracción del texto no presenta ninguna dificultad, simplemente se utiliza las funcionalidades que EasyOCR ofrece y de los productos generados por el módulo ImageProcessor se utiliza las imágenes de las páginas, de las cuales el OCR obtiene el texto y del fichero de metadatos el total de páginas para saber cuántas de estas imágenes hay que utilizar. EasyOCR devuelve como resultado un vector, conteniendo cada índice un sub-vector con las coordenadas de los vértices de la bounding box que ha identificado, el contenido y la probabilidad de que dicho resultado sea correcto.

2.2.1.8 Extraer descripciones imágenes

Una vez leído del fichero de metadatos de que tipo de PDF es el documento que se está procesando, en caso de tratarse se un PDF generado, solo es necesario extraer del texto las descripciones de las imágenes.

Para obtener la descripción de cada imagen del texto, se itera sobre el resultado devuelto por EasyOCR para la página en la que se encuentra la imagen, calculando en primer lugar el centro de la bounding box. Para calcular este centro (figura 10), se aplica la fórmula xc = ((x1 - x0) / 2) + x0 para el eje X, siendo x1 la coordenada en el eje X del vértice superior derecho y x0 la coordenada del vértice superior izquierdo. Para la coordenada en el eje Y la fórmula es similar, yc = ((y2 - y0) / 2) + y0, siendo y2 la coordenada en el eje Y del vértice inferior izquierdo e y0 la coordenada del vértice superior izquierdo. En la figura 10 se muestra de una forma gráfica.

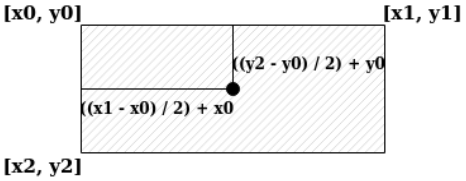


Figura 10: Cálculo del centro de una bounding box

Tomando como referencia las coordenadas contenidas en el fichero de metadatos y el centro calculado, se comprueba que la coordenada Y del centro se encuentra por debajo de la coordenada Y del vértice superior izquierdo, que la coordenada X se encuentra más a la derecha que la coordenada X del vértice inferior izquierdo y más a la izquierda que la coordenada X del vértice inferior derecho, es decir, en las dos primeras comparaciones se comprueba que sea mayor y en la última menor. En caso de que se cumpla lo anterior, se pasa a comprobar que haya una distancia pequeña en el eje Y, inferior a la del texto en los párrafos, entre el centro de la bounding box y el vértice inferior izquierdo. Esta distancia se ha refinado mediante sucesivas pruebas porque no es exacta sino una aproximación, llegando hasta un valor de 50 puntos (figura 11). En la figura 12 se muestra de una forma gráfica.

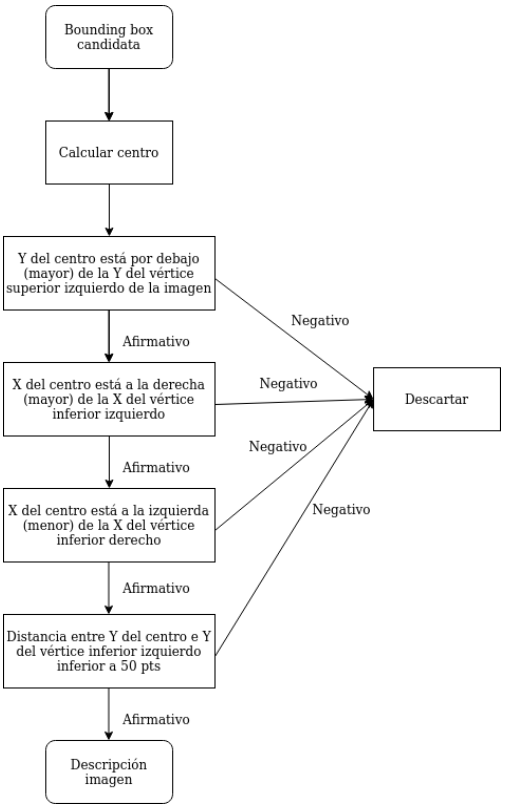


Figura 11: Pasos para la identificación de una descripción de imagen

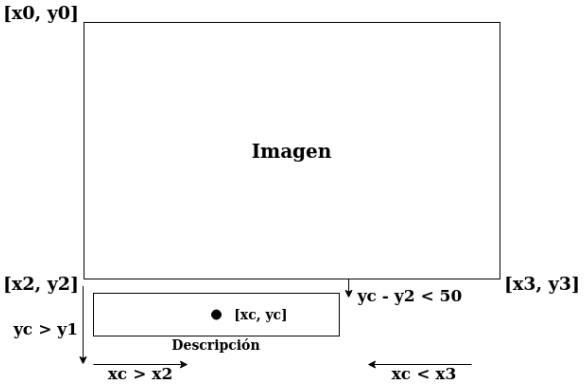


Figura 12: Obtención de la descripción de una imagen

Es posible que EasyOCR no identifique toda la descripción en una única bounding box, por tanto, una vez se ha identificado en el texto extraído una descripción, se comprueba si el vértice superior izquierdo de la bounding box que se está analizando y el vertice superior derecho de la anterior se encuentran a una distancia muy pequeña en el eje X. Esta distancia se ha refinado mediante sucesivas pruebas porque no es algo exacto, sino una aproximación, llegando hasta un valor de 30 puntos. En caso de que esto suceda el contenido de estas bounding box se junta como una única descripción.

Una vez se han identificado las descripciones de las imágenes, estas se guardan y se retiran del vector devuelto por EasyOCR porque posteriormente se utilizará para extraer el contenido de las secciones y nombres.

2.2.1.9 Extraer imágenes y descripciones

En caso de tratarse de un PDF escaneado, además de las descripciones, también se tienen que extraer las imágenes finales que se quieren guardar. Para ello, se utilizan las imágenes intermedias.

Partiendo de la imagen intermedia, se transforma a una representación HSL (Hue, Saturation, Lightness), se detecta y transforma el fondo a negro para, finalmente, transformar a blanco todo aquello que no sea el fondo, asegurando asi un contraste máximo, permitiendo así una fácil detección de los contornos de las imágenes finales. En el Anexo I se encuentra una descripción más detallada de este proceso.

Una vez obtenido el contorno de la imagen, es posible obtener las coordenadas del vértice superior izquierdo, así como la anchura y la largura de cada imagen contenida en la página que se tiene en la imagen intermedia. Con estos valores, se calculan las coordenadas de los vértices inferior izquierdo y derecho y se guarda la imagen de la misma manera que la descrita en el apartado 2.2.1.4.

Para extraer las descripciones de las imágenes en el texto se aplica el mismo procedimiento que se aplica a los PDF generados, descrito en el apartado 2.2.1.8. Se tienen los mismos vértices para las imágenes, variando la forma con la que estas han sido obtenidas. En ambos casos, una vez identificadas las descripciones y guardadas a parte, se eliminan del vector del texto devuelto por EasyOCR para que no estén presentes una vez se procesan las distintas secciones.

2.2.1.10 Extraer secciones y nombres

Una vez se han extraído del texto las descripciones de las imágenes, los vectores devueltos por EasyOCR para cada página se juntan en un único vector para facilitar la extracción de las secciones porque es posible que una sección quede dividida en distintas páginas.

La estructura de las secciones es siempre la misma, estando en primer lugar el título de la sección y después el contenido. Además, el orden de aparición de las secciones no varía, con la salvedad de que las secciones Identificación y Transmisión pueden no estar presentes.

Para extraer el contenido de una sección, se busca en el vector del texto el índice del título de la misma y el índice del título de la siguiente sección. En caso de que la sección a extraer o la siguiente sea una de las dos secciones que pueden no estar presentes, se comprueba primero su existencia. Conociendo estos dos índices, simplemente se extrae del vector el contenido existente entre ambos. En el caso de la última sección del documento, Bibliografía, únicamente se busca el índice del título de la sección y se extrae hasta el final del vector.

El caso de los nombres es completamente diferente al de las secciones, por lo que se aplica un método diferente. Estos se encuentran en un recuadro al comienzo del documento (figura 13). Se distinguen cuatro tipos de nombres: nombre científico de la plaga, nombre común de la plaga, nombre científico del cultivo y nombre común del cultivo. En el lado izquierdo del recuadro aparecen los nombres de la plaga y en el derecho los del cultivo. Existe la posibilidad de que haya varios nombres de un mismo tipo y en el caso del nombre común del cultivo puede aparecer varios cultivos.

En primer lugar, se calcula la mitad de la página en el eje X. Para ello, se divide el valor de la coordenada X del vértice superior derecho de la bounding box que contiene el identificador de la guía por dos, porque es el punto más a la derecha en el eje X de la página. En segundo lugar se busca el índice en el que se encuentra el título de la sección Sinonímia, la cual es la primera sección después de la sección de nombres. Sabiendo que el índice del identificador de la guía es el primero, los nombres están comprendidos entre dicho índice y el del título de la sección de Sinonímia.

Iterando entre estos dos índices, se calcula el centro de la bounding box del nombre que se está analizando, del mismo modo que se describe en el apartado 2.2.1.8. Se comprueba si el nombre se encuentra a la izquierda o derecha de la página, viendo si la coordenada X del centro es menor o mayor que la mitad de la página calculada previamente.

En caso de encontrarse a la izquierda, se trata de un nombre de la plaga. Para diferenciar si es científico o común se utiliza un detector de lenguaje. Si se detecta que el lenguaje es español, se trata de un nombre común porque los nombres científicos están en latín. Es posible que EasyOCR identifique el nombre en distintas bounding boxes, por tanto, se aplica el mismo método que se aplica a las descripciones de las imágenes para juntar ambas partes como un único nombre.

En caso de encontrarse a la derecha, se trata de un nombre del cultivo. Para diferenciar si es científico o común, resulta más sencillo porque el nombre común del cultivo está en mayúsculas. Es posible que una plaga afecte a varios cultivos, por lo que una vez identificado el nombre común, se comprueba que sea un único cultivo o varios.

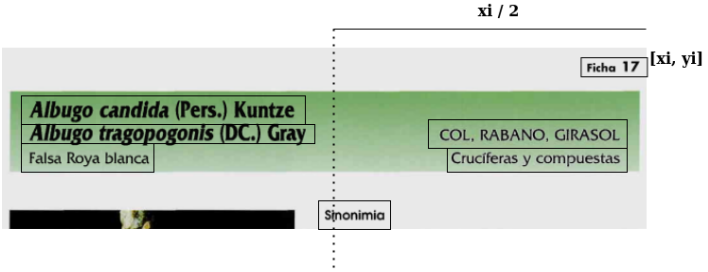


Figura 13: Distribución de los nombres

Dado que el OCR no proporciona ningún resultado respecto al tipo de letra del texto o tamaño de la misma, para distinguir los nombres de la plaga solo es posible aplicar un método que analice el significado del nombre. En el caso del cultivo es más sencillo debido a que el nombre común está enteramente en mayúsculas.

2.2.1.11 Guardar resultados

Como último paso, se guardan los resultados obtenidos en la base de datos MongoDB, siguiendo el modelo de datos descrito en el Anexo III. En caso de que la plaga de la que trate el documento afecte a más de un cultivo y, por tanto, haya varios nombres comunes, cada uno de ellos se guarda por separado.

2.2.2 Aplicación web

La aplicación web está compuesta por cuatro elementos (figura 14), la base de datos, MongoDB, almacena la información obtenida con el sistema de extracción siguiendo el modelo de datos descrito en el Anexo III. Ofrece los documentos en formato BSON, el cual presenta la misma estructura que el formato JSON. El servidor, implementado con Node.js y Express, atiende las peticiones del cliente a través de tres controladores, uno para cada tipo de documento que se almacena en la base de datos, ofreciendo los resultados en formato JSON. El servidor de recursos (http.server), atiende las peticiones del cliente para obtener los PDF en caso de descarga y las imágenes finales obtenidas con el sistema de extracción. Debido a que los documentos BSON de MongoDB están limitados a 16 MB se ha decidido desacoplar estos archivos para evitar posibles problemas de almacenamiento y guardar en estos documentos las rutas en las que están disponibles. Por último, el cliente, implementado con React, muestra la información final al usuario.

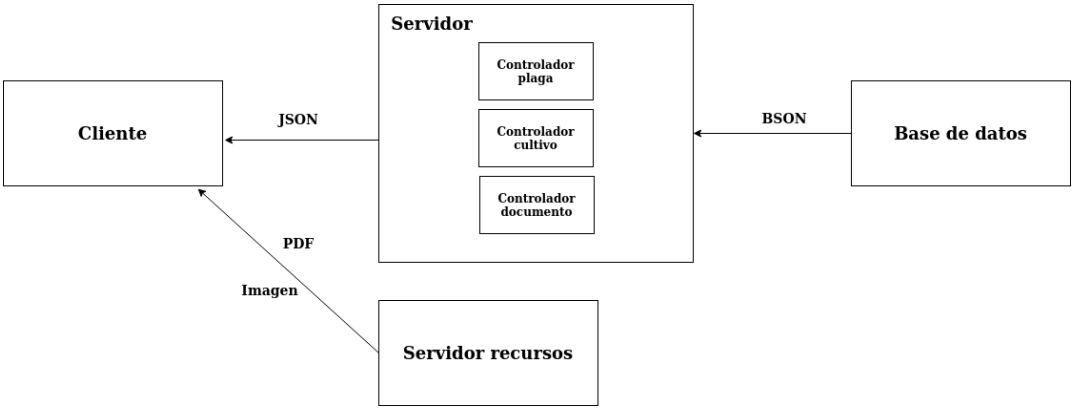


Figura 14: Estructura de la aplicación web

Para facilitar el uso de la API ofrecida por el servidor, se ha documentado utilizando Swagger.

Al acceder a la aplicación se muestra la vista de la figura 15. En caso de seleccionar un cultivo del desplegable de la parte izquierda, se muestra el listado de plagas que le afectan (figura 16), mostrando para cada plaga del listado su nombre común, científico y distribución en España. Si se selecciona el botón Ver presente en cada plaga del listado o una plaga del desplegable de la izquierda, se muestra la información detallada de la misma (figuras 17, 18 y 19), mostrando los distintos campos y finalmente las imágenes con sus descripciones. Además, al principio, se tiene la opción de descargar el documento seleccionando el botón con el icono en forma de PDF.

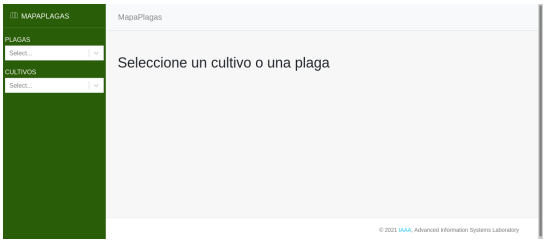


Figura 15: Vista inicial

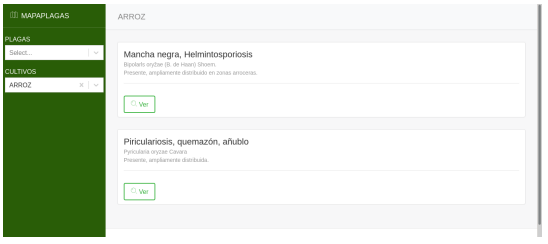


Figura 16: Listado de plagas que afectan a un cultivo

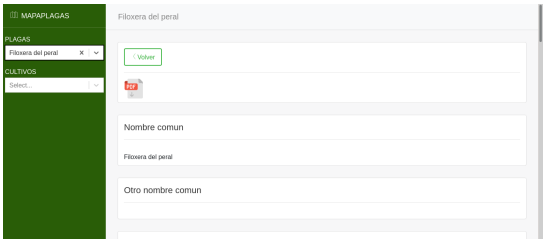


Figura 17: Información detallada de una plaga (1)

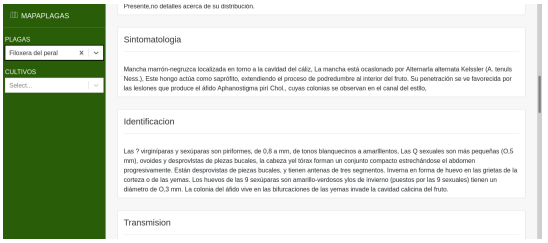


Figura 18: Información detallada de una plaga (2)

Figura 19: Información detallada de una plaga (3)

2.3 Problemas encontrados y resultados

Debido a la heterogeneidad de los documentos de la colección, su procesamiento ha resultado complejo. El hecho de que haya dos tipos de documentos ha hecho que sea necesario aplicar distintos pasos en determinados puntos del sistema, como es el caso de las imágenes. En primer lugar, del total de 400 documentos que componen la colección, 6 no estaban disponibles, por lo que no han podido ser descargados. Una vez procesada la colección, se ha observado que algunas guías no tienen nombre científico de los cultivos y otras no comienzan por la sección Sinonimia, sino por Distribución en España. Estos dos aspectos generan problemas a la hora de procesar la parte de los nombres, no pudiendo procesar la guía en cuestión correctamente. Por otra parte, aunque el nombre científico de la plaga esté en latín, al no tratarse de un latín clásico sino de un latín científico [23] proveniente del uso moderno del latín, provoca que la biblioteca utilizada para la detección del lenguaje arroje falsos negativos, no pudiendo realizar un procesamiento correcto. Para una parte de la colección, 50 guías consecutivas, las cuales son PDF escaneados, no se ha podido obtener las imágenes intermedias necesarias para extraer posteriormente las imágenes finales.

Tras procesar la colección de documentos, debido a los problemas mencionados, se han procesado completamente el 30% de las guías. Respecto al coste en tiempo, todo el proceso ha conllevado un tiempo de ejecución de 4 horas, desde la descarga de la colección hasta que la información se encuentra almacenada en la base de datos. Pasando por los productos intermedios generados en el sistema de extracción, las imágenes intermedias ocupan 50,8 MB para un total de 540 elementos, los ficheros de metadatos 0,276 MB para 394 elementos y las imágenes de las páginas 28,5 GB para 784 elementos. Respecto a los productos finales del sistema, la colección de documentos ocupa 307,9 MB para un total de 394 elementos, las imágenes finales que se muestran en la aplicación web, 121,1 MB para 1705 elementos y la base de datos ocupa un total de 6,36 MB.

Con el objetivo de mejorar el porcentaje obtenido, se han llevado a cabo mejoras en el sistema de extracción teniendo en cuenta los problemas surgidos. Se tiene en cuenta que una guía puede no tener nombre científico del cultivo o cultivos y se comprueba si existe la sección Sinonimia. Respecto a la problemática del latín, para identificar los nombres de la plaga se tiene en cuenta que se trate de español o no. En caso de identificarse como español, se considera nombre común y en caso contrario nombre científico. Para las 50 guías consecutivas de las que no es posible obtener imágenes intermedias, no ha sido posible encontrar una solución.

Una vez aplicadas estas mejoras, se ha conseguido procesar completamente el 52% de las guías, un 34% parcialmente y un 14% de las que no se ha podido obtener nada. Las guías que han podido ser procesadas parcialmente han presentado fallos mayoritariamente en la identificación de los nombres y en menor medida en la extracción de las secciones, pudiendo guardar de estas imágenes y la información referente a los cultivos.

3 LECCIONES APRENDIDAS Y CONCLUSIONES

3.1 Conocimientos adquiridos

Tras la realización de este proyecto, se han adquirido conocimientos sobre el procesamiento de documentos PDF mediante el uso de herramientas como PDFBox y EasyOCR. El trabajo con unidades de medida utilizadas en imágenes, puntos por pulgada (DPI), y en PDF (puntos). Extracción de imágenes contenidas en otras imágenes aplicando la representación HSL. También se ha profundizado en el uso del lenguaje Python. Respecto a la creación de una aplicación web, se ha aprendido a crear un cliente web utilizando un entorno como React, el cual no tiene una curva de aprendizaje sencilla.

3.2 Ideas Futuras

Una ampliación interesante a explorar sería la inclusión de nuevas fuentes de información. Una de ellas podría ser la colección de guías de gestión integrada de plagas [2]. Su tratamiento no variaría demasiado de la colección de fichas tratada en este proyecto aunque cada guía está centrada en un tipo de cultivo en lugar de una plaga. Otra posible fuente de información podría ser Wikispecies [21], se trata de un proyecto de la Fundación Wikipedia como un directorio libre de especies, posee información de animales, plantas, hongos, bacterias, arqueas, protistas y otras formas de vida. Es considerablemente mayor que la anterior y requeriría de mayor tratamiento para filtrar la información deseada. De forma similar a Wikispecies, está EUNIS (Sistema Europeo de Información sobre la Naturaleza) [22], el cual proporciona acceso a datos, disponibles públicamente, para especies, tipos de hábitats y lugares protegidos en toda Europa.

3.3 Conclusiones

Este proyecto ha resultado un gran desafío, trabajar con documentos PDF para extraer su contenido no ha resultado nada sencillo. En lo que respecta a herramientas de uso público, es un campo poco trabajado y la mayoría de ellas presentan deficiencias, únicamente PDFBox, al tener a Apache detrás, proporciona lo esperado. De los PDF escaneados resulta un quebradero de cabeza extraer su contenido correctamente, lo que ha llevado a necesitar utilizar más herramientas como en este caso un OCR. Trabajar con imágenes, principalmente para extraer otras contenidas dentro de estas, ha requerido comprender conceptos nuevos, no relacionados con la especialidad de este trabajo de fin de grado, como es la representación HSL, para poder aplicar nuevas técnicas correctamente. Respecto a la aplicación web, React no

ha resultado fácil de aprender a utilizar, pero una vez se ha comprendido su funcionamiento crear un cliente web se ha podido realizar rápidamente.

4 BIBLIOGRAFÍA

[1] Gobierno de España. “Ministerio de Agricultura, Ganadería y Pesca.” *Ministerio de Agricultura, Ganadería y Pesca*, https://www.mapa.gob.es/es/ministerio/default.aspx.

[2] 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/.

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

[4] 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.

[5] Apache Software Foundation. “Apache PDFBox.” *Apache PDFBox*, https://pdfbox.apache.org/.

[6] PyPI. “PyPDF2.” *PyPDF2*, https://pypi.org/project/PyPDF2/.

[7] Mathieu Fenniak. “Mathieu Fenniak.” *Mathieu Fenniak Page*, https://mathieu.fenniak.net/.

[8] PyPI. “PDFMiner.six.” *PDFMiner.six*, https://pypi.org/project/pdfminer.six/.

[9] PDF Clown Team. “PDF Clown.” *PDF Clown*, https://pdfclown.org/.

[10] PyPI. “pdfrw.” *pdfrw*, https://pypi.org/project/pdfrw/.

[11] nassibnassar. “PJX.” *PJX*, https://github.com/nassibnassar/pjx.

[12] tesseract-ocr. “tesseract.” *tesseract*, https://github.com/tesseract-ocr/tesseract.

[13] sourceforge. “Tess4j.” *Tess4j - JNA wrapper for Java*, http://tess4j.sourceforge.net/.

[14] JaidedAI. “EasyOCR.” *EasyOCR*, https://github.com/JaidedAI/EasyOCR.

[15] Colaboradores de Wikipedia. “MongoDB.” *MongoDB - Wikipedia*, https://en.wikipedia.org/wiki/MongoDB.

[16] dockerhub. “mongo.” *mongo*, https://hub.docker.com/_/mongo.

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

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

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

[20] Python Software Foundation. “http.server.” *http.server - HTTP servers - Python 3.9.6 documentation*, https://docs.python.org/3/library/http.server.html.

[21] Fundación Wikipedia. “Wikispecies.” *Portada - Wikispecies*, https://species.wikimedia.org/wiki/Portada.

[22] Agencia Europea de Medio Ambiente. “EUNIS.” *EUNIS - Welcome to EUNIS Database*, https://eunis.eea.europa.eu/.

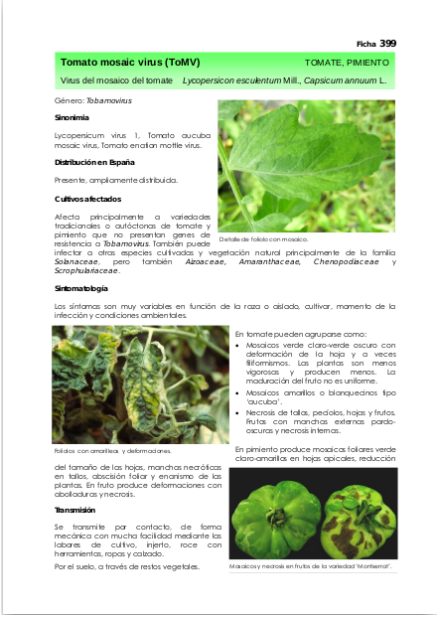
[23] Colaboradores de Wikipedia. “Latín contemporáneo.” *Latín contemporáneo - Wikipedia*, https://es.wikipedia.org/wiki/Lat%C3%ADn_contempor%C3%A1neo#Lat%C3%ADn_cient%C3%ADfico.

[24] Colaboradores de Wikipedia. “Modelo de color HSL.” *Modelo de color HSL - Wikipedia*, https://es.wikipedia.org/wiki/Modelo_de_color_HSL.

5 Anexo I. Investigación sobre tecnologías para la extracción de datos de documentos PDF

Como previo paso ha sido necesario realizar una investigación sobre las tecnologías existentes para la extracción de la información contenida en documentos PDF. Se han seleccionado una serie de documentos de los ofrecidos por el Ministerio de Agricultura, Ganadería y Pesca que presentan una serie de características en su contenido que conviene observar qué resultados proporcionan las tecnologías seleccionadas. En esta sección se describen las alternativas consideradas para la extracción de la información contenida en documentos PDF.

Se distinguen dos tipos de PDF, generados a partir de documentos de texto (figuras 20 y 22) y generados a partir de elementos escaneados (figura 21). Dentro del primer tipo, se analiza también distintos tipos de contenido, en párrafos (figura 20) y en tablas (figura 22). El segundo tipo únicamente contiene texto en párrafos. Por último, para ambos casos se comprueba la extracción de imágenes.



Ficha 399

Tomato mosaic virus (ToMV) — TOMATE, PIMIENTO

Virus del mosaico del tomate — *Lycopersicon esculentum* Mill., *Capsicum annuum* L.

Género: *Tobamovirus*

Sinonimia

Lycopersicum virus 1, Tomato aucuba mosaic virus, Tomato enation mottle virus.

Distribución en España

Presente, ampliamente distribuida.

Cultivos afectados

Afecta principalmente a variedades tradicionales o autóctonas de tomate y pimiento que no presentan genes de resistencia a *Tobamovirus*. También puede infectar a otras especies cultivadas y vegetación natural principalmente de la familia *Solanaceae*, pero también *Alsinaceae*, *Amaranthaceae*, *Chenopodiaceae* y *Scrophulariaceae*.

Sintomatología

Los síntomas son muy variables en función de la raza o aislado, cultivar, momento de la infección y condiciones ambientales.

En tomate pueden agruparse como:

- Mosaicos verde claro-verde oscuro con deformación de la hoja y a veces filiformismos. Las plantas son menos vigorosas y producen menos. La maduración del fruto no es uniforme.
- Mosaicos amarillos o blanquecinos tipo 'aucuba'.
- Necrosis de tallos, pecíolos, hojas y frutos. Frutos con manchas externas pardo-oscuras y necrosis internas.

En pimiento produce mosaicos foliares verde claro-amarillo en hojas apicales, reducción del tamaño de las hojas, manchas necróticas en tallos, abscisión foliar y enanismo de las plantas. En fruto produce deformaciones con abolladuras y necrosis.

Transmisión

Se transmite por contacto, de forma mecánica con mucha facilidad mediante las labores de cultivo, injerto, roce con herramientas, ropas y calzado.

Por el suelo, a través de restos vegetales.

Figura 20: PDF generado a partir de un documento de texto con párrafos

Ficha 1

***Aphanostigma piri* Chol.** — PERAL

Filoxera del peral — *Pyrus communis* L.

Sinonimia

(*Homoptera: Phylloxeridae*). *Phylloxera piri* Chol.

Distribución en España

Presente, no hay detalles acerca de su distribución.

Cultivos afectados

Peral.

Sintomatología

Mancha marrón-negruzca localizada en torno a la cavidad del cáliz. La mancha está ocasionada por *Alternaria alternata* Keissler (*A. tenuis* Nees.). Este hongo actúa como saprófito, extendiendo el proceso de podredumbre al interior del fruto. Su penetración se ve favorecida por las lesiones que produce el áfido *Aphanostigma piri* Chol., cuyas colonias se observan en el canal del estilo.

Análisis de la muestra

Tomar la zona del cáliz y buscar en su interior el áfido, ya que el hongo *Alternaria*, siempre presente, dificulta la localización del pequeño insecto (0,8-1 mm). Observando detenidamente, se aprecian masa de huevos, en algunos casos eclosionados, y adultos de *A. piri* Chol.

Identificación

Las ♀ virginíparas y sexúparas son piriformes, de 0,8 a 1 mm, de tonos blanquecinos a amarillentos. Las ♀ sexuales son más pequeñas (0,6 mm), ovoides y desprovistas de piezas bucales, la cabeza y el tórax forman un conjunto compacto, estrechándose el abdomen progresivamente. Están desprovistas de piezas bucales, y tienen antenas de tres segmentos. Invierna en forma de huevo en las grietas de la corteza o de las yemas. Los huevos de las ♀ sexúparas son amarillo-verdosos y los de invierno (puestos por las ♀ sexuales) tienen un diámetro de 0,3 mm. La colonia del áfido vive en las bifurcaciones de las yemas e invade la cavidad calicina del fruto.

Bibliografía

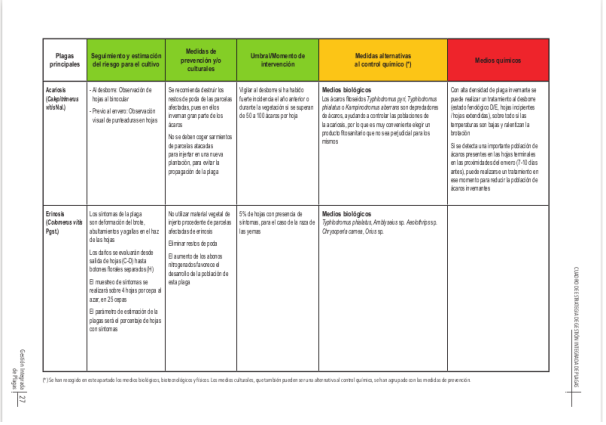
Cabeno, C.A., 1981: La phylaxéra du poirier. Un nouveau problème à considérer au Portugal. En: Les pucerons des cultures. ACTA. 247-248.

Goldasch, G., 1990: Manuale di patologia vegetale. Vol II. Ed. Edagricola.

Pozzi, A., 1989: La difesa delle piante da frutto. Ed. Edagricole.

GRUPO DE TRABAJO DE LABORATORIOS DE DIAGNÓSTICO
Laboratorio Agrario, Biología Vegetal, del Gobierno de Navarra.
Borrueel, M. y Plaza, B.

Figura 21: PDF generado a partir de elementos escaneados



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
Acariosis (Calepitrimerus vitis Nal.)	- Al desborre: Observación de hojas al trasluz - Previo al envero: Observación visual de punteaduras en hojas	Se recomienda destruir los restos de poda de las parcelas afectadas, pues en ellos invernan gran parte de los ácaros No se deben coger sarmientos de parcelas afectadas para injertar en una nueva plantación, para evitar la propagación de la plaga.	Vigilar al desborre si ha habido fuerte incidencia el año anterior o durante la vegetación si se superan de 50 a 100 ácaros por hoja	**Medios biológicos** Los ácaros fitoseidos *Typhlodromus pyri*, *Typhlodromus phialatus* o *Kampimodromus aberrans* son depredadores de ácaros, ayudando a controlar las poblaciones de la acariosis, por lo que es muy conveniente elegir un producto fitosanitario que no sea perjudicial para los mismos	Con alta densidad de plaga invernante se puede realizar un tratamiento al desborre (estado fenológico D/E, hojas incipientes /hojas extendidas), sobre todo si las temperaturas son bajas y ralentizan la brotación Si se detecta una importante población de ácaros presentes en las hojas terminales en las proximidades del envero (7-15 días antes), puede realizarse un tratamiento en ese momento para reducir la población de ácaros invernantes
Erinosis (Colomerus vitis Pgst)	Los síntomas de la plaga son deformación del brote, abultamientos y agallas en el haz de las hojas Los daños se evalúan desde salida de hojas (C-D) hasta botones florales separados (H) El muestreo de síntomas se realizará sobre 4 hojas por cepa a azar, en 25 cepas El parámetro de estimación de la plaga será el porcentaje de hojas con síntomas	No utilizar material vegetal de injerto procedente de parcelas afectadas de erinosis Eliminar restos de poda El aumento de los ácaros nitrogenados favorece el desarrollo de la población de esta plaga	5% de hojas con presencia de síntomas, para el caso de la raza de las yemas	**Medios biológicos** *Typhlodromus phialatus*, *Amblyseius* sp. *Aeolothrips* sp. *Chrysoperla carnea*, *Orius* sp.	

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

Figura 22: PDF generado a partir de un documento de texto con tablas

PDF generado a partir de un documento de texto

PDFBox

PDFBox es una biblioteca de código abierto para Java creada y mantenida por Apache [5].

Ofrece una serie de operaciones para extraer el texto de un documento, aunque la que resulta más interesante es la que permite indicar un número de página como parámetro y devuelve el contenido en un string. En primer lugar, siempre extrae el pie de página junto con el número de la misma, después el encabezado. Esto sucede independientemente del contenido de la página, ya sea párrafos o tablas. En el caso de los párrafos, respeta su separación y en el caso de las tablas, extrae el contenido por filas. Respeta mayúsculas, minúsculas, tildes y caracteres especiales, como paréntesis y comillas, además no presenta problemas con la letra ñ.

Respecto a las imágenes, tiene operaciones específicas, siendo la más interesante la que presenta un funcionamiento similar a la descrita anteriormente para el texto, permitiendo indicar el número de página. Permite extraer las imágenes en distintos formatos y presentan buena calidad.

Como observaciones que resulta interesante destacar, el hecho de que devuelva el texto de una página en un string requiere de un trabajo adicional sobre este en caso de que se quiera alguna parte en concreto. Algunas imágenes las extrae giradas 180 grados y volteadas, lo cual requiere también de un trabajo extra para corregirlas. Como aspectos positivos cabe destacar que requiere pocas líneas de código para realizar las distintas extracciones, está bien documentada, Apache sigue realizando mantenimiento sobre ella y existen multitud de tutoriales.

PyPDF2

PyPDF2 [6] es una biblioteca Pure-Python originada a partir del proyecto PyPDF creado por Mathieu Fenniak [7].

Al igual que en el caso anterior, presenta una operación que permite extraer el contenido de la página que se indique. Extrae el número de página, pero sin embargo no extrae ni el encabezado ni el pie de página, independientemente de si el contenido se encuentra en párrafos o tablas. Respecto a los párrafos, respeta su separación y respecto a las tablas, extrae su contenido por filas, aunque sin ofrecer ninguna separación entre el contenido de cada celda. Respeta mayúsculas, minúsculas, tildes y la letra ñ, sin embargo presenta problemas con la letra f y las comillas, confundiéndose con otros caracteres.

Pasando a las imágenes, el funcionamiento es similar a PDFBox, sin embargo presenta un error a la hora de extraerlas: *NotImplementedError: unsupported filter /DCTDecode*.

Como observaciones, cabe destacar los problemas que presenta con la letra f y las comillas mencionados anteriormente, así como con imposibilidad de extraer las imágenes. Además, es necesario desencriptar el PDF, lo cual requiere utilizar un software tercero. Se ha utilizado qpdf en este caso, pero esto requiere añadir más código y crea una copia nueva del documento. Por último, PyPDF2 ya no recibe mantenimiento.

PDFMiner.six

PDFMiner.six [8] es una biblioteca de Python creada y mantenida por la comunidad a partir del PDFMiner original.

Al contrario que los casos anteriores, no presenta una operación que permita extraer el contenido de una página en concreto directamente. Esto implica recorrer todas las páginas del documento hasta encontrar la que se desea. Respecto a la extracción del contenido, tanto para párrafos como para tablas, extrae en primer lugar el encabezado y pie de página. Para el caso concreto de los párrafos, no respeta la separación de los mismos, extrayendo todo junto. Para el caso de las tablas, extrae el contenido por filas, pero, al igual que sucede con los párrafos, no presenta ninguna separación entre el texto de cada celda. Presenta algún pequeño error pero por lo general respeta mayúsculas, minúsculas, tildes, caracteres especiales y la letra ñ.

Respecto a la imágenes, no ofrece ninguna operación que permita extraerlas a través de un programa, únicamente a través de línea de comandos.

Como observaciones, destacar la imposibilidad de extraer el contenido de una página directamente, lo cual obliga a recorrer todas las páginas del documento, lo cual reduce su rendimiento respecto a las anteriores bibliotecas, además de la imposibilidad de

extraer las imágenes. El hecho de que no respete la separación del texto, dificulta su tratamiento si se quiere alguna parte en concreto. Al igual que ocurre con PyPDF2, es necesario utilizar alguna herramienta externa para desencriptar el PDF primero. La documentación no es demasiado detallada.

PDF Clown

PDF Clown [9] es una biblioteca de código abierto para Java y .NET creada por Stefano Chizzolini.

Al igual que ocurre con PDFMiner.six, no ofrece una operación que permita extraer el contenido de una página directamente, por lo que es necesario recorrer todas las páginas del documento hasta encontrar la que se desea. Tanto si el contenido se encuentra en párrafos como en tablas, se extrae en primer lugar el número de página. Respecto a los párrafos, respeta la separación entre los mismos y para las tablas, extrae el contenido por filas, respetando también la separación entre celdas. No presenta fallos en mayúsculas, minúsculas, tildes, caracteres especiales y la letra ñ.

Respecto a las imágenes, únicamente las consigue extraer en formato jpeg. Además ocurre de forma similar al texto, no ofrece ninguna operación para obtener las de una página en concreto.

Como observaciones, presenta cadencias similares a PDFMiner.six, no ofrece una operación que permita acceder directamente a una página, lo cual merma su rendimiento respecto a las demás. A su vez, no es posible aplicar el mismo método para obtener una página en concreto a las imágenes, por lo que solo es posible obtener todas las presentes en el documento. No se ha encontrado ningún tutorial útil que explique su funcionamiento y la documentación no es muy abundante. Por otra parte, no está presente en Maven, por lo que hay que importarla manualmente.

Otras

Además de las anteriores, se han investigado otras en las que no se ha profundizado demasiado.

- **pdfrw**

  pdfrw [10] es una biblioteca para Python creada por Patrick Maupin. No se ha profundizado más en ella porque presenta muchas menos características que PyPDF2.

- **PJX**

  PJX [11] es una biblioteca para Java creada por Etymon Systems. No se ha profundizado más en ella porque no presenta ningún tipo de documentación y se encuentra abandonada desde hace bastante tiempo.

Conclusiones

	Texto	Tabla	Imagen	Mantenimiento	Documentación
PDFBox					
PyPDF2					
PDFMiner.six					
PDF Clown					

Tabla 2: Resultados que presenta cada tecnología

	No presenta problemas
	Presenta problemas parciales
	Presenta considerables problemas

Finalmente, se ha decidido utilizar PDFBox para extraer la información de documentos de este tipo, ya que es la que presenta mejores resultados conforme a los buscados para el proyecto, como se puede ver en la tabla 2.

PDF generado a partir de elementos escaneados

Para este tipo de documentos, se ha observado que las bibliotecas anteriores no son suficientes. Los resultados que ofrecen presentan múltiples fallos en el texto y en la extracción de las imágenes. Como solución a este problema se ha optado por investigar qué resultados se pueden obtener combinando la biblioteca PDFBox, ya que en el apartado anterior se ha visto que es la mejor opción, junto con un OCR.

El proceso seguido es el mismo para todas las opciones investigadas, con PDFBox se obtiene una imagen de la página a través de la operación que ofrece para ello. Esta imagen se pasa al OCR para que la analice y extraiga el texto contenido en la misma.

PDFBox y Tesseract (Tess4j)

Tesseract [12] es un OCR de código abierto, desarrollado por Google hasta 2018. Se ha utilizado a través del wrapper Tess4j [13] para Java.

Respecto a la extracción de texto, presenta peor calidad que el resultado obtenido únicamente con PDFBox. Presenta múltiples errores en la ortografía, confundiendo letras o directamente no pudiendo identificarlas, la mayor parte del texto resulta ininteligible.

Como observaciones, previamente se han tenido que descargar datos de entrenamiento en español y destacar, también, el proceso de configuración, el cual ha presentado diversos problemas, llegando a provocar fallos con la JVM. A PDFBox se le puede indicar la calidad con la que generar la imagen de la página. Teniendo en cuenta esto, se ha probado a mejorar la calidad de la imagen que debe analizar Tesseract, viendo así si se podían obtener mejores resultados, pero se ha llegado al mismo.

PDFBox y EasyOCR

EasyOCR [14] es un OCR *ready-to-use* con soporte para más de 80 lenguajes.

El texto extraído presenta una gran calidad en comparación con la opción anterior, aparecen algunos pequeños fallos, como confundir puntos con comas, pero que pueden ser solucionados aumentando la calidad de la imagen que debe analizar el OCR. Además junto a cada porción de texto, también devuelve en el resultado las coordenadas de los vértices de la bounding box donde lo ha identificado.

Como observaciones, destacar el elevado tiempo de ejecución del OCR. Desde la propia documentación del mismo se recomienda realizar la ejecución en GPU. Por otra parte, no requiere descargar ningún tipo de datos de entrenamiento.

Problemática con la extracción de las imágenes

Como se mencionó anteriormente, la extracción de las imágenes también supone un problema en este tipo de documentos. Aunque los resultados que ofrece PDFBox no son válidos en una primera instancia, si se pueden aplicar una serie de técnicas sobre ellos para obtener resultados finales válidos.

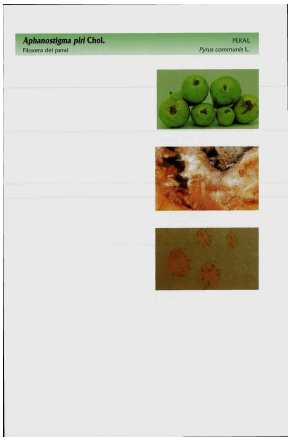


Figura 23: Imagen devuelta por PDFBox al extraer las imágenes de una página

Como se puede observar en la figura 23, PDFBox devuelve una imagen que no contiene el texto de la página, pero si las imágenes que interesa guardar, que en el caso mostrado son 3. Con el fin de obtener las imágenes finales, se transforma la imagen a una representación HSL [24], porque permite trabajar mejor con los contrastes de colores que la representación RGB. Teniendo la imagen en esta nueva representación, se detecta el fondo y se convierte a negro para asegurar el contraste máximo, estableciendo una franja de a qué color se considera blanco y por ende hay que transformar a negro. Con la imagen con el fondo negro (figura 24), se convierte en blanco las partes que no están en negro (figura 25), teniendo así una imagen binaria, permitiendo de este modo una detección de contornos sencilla. Los contornos detectados se filtran para eliminar el recuadro que contiene los nombres y finalmente se recortan las imágenes finales de la imagen de la que se ha partido utilizando las coordenadas de los contornos.

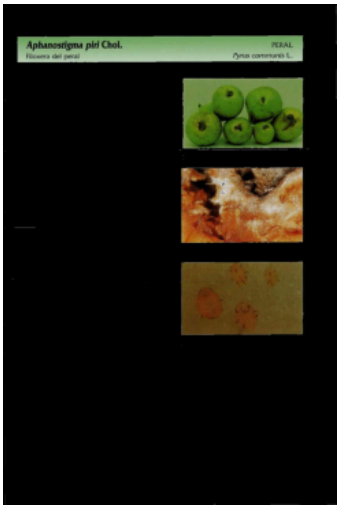


Figura 24: Detección del fondo y transformación a negro

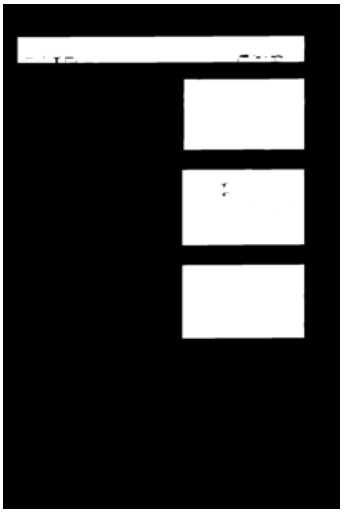


Figura 25: Transformación a blanco de las partes restantes

Conclusiones

Finalmente, se ha decidido utilizar la opción de PDFBox junto con EasyOCR, ya que además de los buenos resultados obtenidos, el hecho de que proporcione también las coordenadas de las bounding box que identifica puede facilitar la identificación de las distintas secciones del documento, así como la posibilidad de poder extraer también las descripciones de las imágenes. Como punto en contra está el elevado tiempo de ejecución, pero se puede disponer de una máquina con una GPU potente para poder realizarla dentro de unos términos menos elevados.

6 Anexo II. Descripción de las herramientas de trabajo

- **Herramientas software**
  - **Herramientas del proceso de extracción**
    - **PDFBox**

      Apache PDFBox [5] es una biblioteca de código abierto de Java que permite trabajar con documentos PDF. Permite la creación de nuevos documentos, la manipulación de documentos existentes y la capacidad de extraer el contenido. También incluye varias utilidades para línea de comandos. Se publica bajo la licencia Apache v2.0.

    - **EasyOCR**

      EasyOCR [14] es un OCR *ready-to-use* para Python, soporta más de 80 lenguajes y los alfabetos más populares. El hecho de que sea *ready-to-use*, implica que no requiere de ninguna carga de datos de entrenamiento. El proyecto está basado en investigaciones y códigos de varios artículos y repositorios de código abierto. Está publicado bajo licencia Apache v2.0.

  En el Anexo I se explica la investigación llevada a cabo para finalmente seleccionar estas dos tecnologías.

  - **Herramientas utilizadas para la base de datos**
    - **MongoDB**

      MongoDB [15] es una base de datos orientada a documentos multiplataforma. Se trata de una base de datos NoSQL que usa documentos similares a JSON, conocidos como BSON, con esquemas opcionales. Es desarrollado MongoDB Inc y tiene licencia SSPL. Para su uso, ha sido desplegada a través de Docker [16].

  - **Herramientas utilizadas para la aplicación web**
    - **Node.js**

      Node.js [17] es un entorno de ejecución de JavaScript para back-end, multiplataforma y de código abierto. Representa el paradigma *"JavaScript everywhere"*, que unifica el desarrollo de

aplicaciones web en torno a un solo lenguaje de programación, el lugar de diferentes lenguajes para el cliente y el servidor.

- **Express**

  Express es un framework de aplicación web para Node.js, lanzado como software gratuito y de código abierto bajo licencia MIT. Está diseñado para crear aplicaciones web y API. Se le considera el framework principal para Node.js.

- **React**

  React [19] es una biblioteca JavaScript para front-end de código abierto utilizada para construir interfaces de usuario o componentes. Es mantenida por Facebook y una comunidad de desarrolladores y empresas individuales.

- **http.server**

  http.server [20] es un módulo de Python que permite crear rápidamente servidores HTTP (servidores Web) a través de línea de comandos.

MongoDB, Node.js, Express y React conforman el stack MERN, el cual se ha decidido utilizar debido a su amplio uso en el ámbito empresarial hoy en día.

- **Herramientas hardware**
  - **Máquina Endurance**

    Máquina presente en el departamento de grupo de investigación IAAA. Cuenta con una tarjeta gráfica NVIDIA GP102GL y 31 GB de memoria VRAM, lo cual es más que suficiente para poder procesar la colección de documentos seleccionados. EasyOCR tiene un mejor funcionamiento utilizando una GPU, por este motivo se ha decidido utilizar esta máquina.

7 Anexo III. Modelo de datos

Por las características de la base de datos MongoDB utilizada, la información extraída se va a guardar en documentos. Por tanto, el modelo de datos creado consiste en la estructura que presentan los distintos documentos que se van a tener. Este modelo de datos es utilizado por el sistema de extracción y por la aplicación web.

Para guardar la información referente al propio PDF, se tiene la colección *Documents*, presentando cada documento la estructura que se puede ver en la tabla 3:

Campo	**Tipo**	**Descripción**
ID	Mongo ID	ID generado por MongoDB
Name	String	Nombre del PDF
Directory	String	Directorio donde está guardado

Tabla 3: Campos de un documento de la colección Documents

Para guardar la información referente al cultivo, se tiene la colección *Crops*, presentando cada documento la estructura que se puede ver en la tabla 4:

Campo	**Tipo**	**Descripción**
ID	Mongo ID	ID generado por MongoDB
CommonName	String	Nombre común del cultivo
ScientificName	String	Nombre científico del cultivo
OtherNames	String	Otros nombres científicos que pueda presentar el cultivo

Tabla 4: Campos de un documento de la colección Crops

Para guardar la información referente a la plaga, se tiene la colección *Plagues*, presentando cada documento la estructura que se puede ver en la tabla 5:

Campo	**Tipo**	**Descripción**
ID	Mongo ID	ID generado por MongoDB
CommonName	String	Nombre común de la plaga
OtherCommonName	String	Otros nombres comunes que pueda presentar la plaga

ScientificName	String	Nombre científico de la plaga
OtherScientificName	String	Otros nombres científicos que pueda presentar la plaga
Distribution	String	Contenido de la sección Distribución en España
Symptomatology	String	Contenido de la sección Sintomatología
Identification	String	Contenido de la sección Identificación
Transmission	String	Contenido de la sección Transmisión
Analysis	String	Contenido de la sección Análisis de la muestra
Sources	String	Contenido de la sección Bibliografía
Crops	Array	Array que contiene los ID de los cultivos a los que afecta la plaga
Document	Mongo ID	ID del PDF de la plaga
ImagesNames	Array	Array que contiene los nombres con los que se ha guardado las imágenes contenidas en el PDF de la plaga
Directory	String	Directorio donde están guardadas las imágenes
ImagesHeaders	Array	Array que contiene las descripciones de las imágenes

Tabla 5: Campos de un documento de la colección Plagues

8 Anexo IV. Logging en el sistema de extracción

Para poder tener conocimiento de posibles problemas durante el procesamiento de una guía y poder conocer los distintos elementos que se han podido extraer, se llevó a cabo una serie de logs para los distintos pasos del procesamiento de una guía.

A la hora de descargar la colección de documentos, se escribe en el fichero Download-SG.txt (figura 26) si se ha podido descargar una guía o no. Esto ha permitido saber que algunas guías no se encontraban disponibles.

```
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_108.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_109.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_110.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_111.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_112.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_113.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_114.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_115.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_116.pdf
Response error from https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_116.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_117.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_118.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_119.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_120.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_121.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_122.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_123.pdf
Downloading https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_124.pdf
```

Figura 26: Extracto del fichero Download-SG.txt

Cuando se obtienen las imágenes de las páginas de cada guía, en el fichero PDFBox-Pages.txt (figura 27), se escribe que guía se está procesando y las imágenes que se han guardado.

```
******************************
Processing guide 3
Saved image fd_003-1.png
Saved image fd_003-2.png
******************************
Processing guide 4
Saved image fd_004-1.png
Saved image fd_004-2.png
******************************
Processing guide 5
Saved image fd_005-1.png
Saved image fd_005-2.png
******************************
Processing guide 6
Saved image fd_006-1.png
Saved image fd_006-2.png
******************************
Processing guide 7
```

Figura 27: Extracto del fichero PDFBox-Pages.txt

En el fichero PDFBox-Images.txt (figuras 28 y 29) se registran para cada guía las imágenes que se han podido extraer. En caso de tratarse de un PDF escaneado, se registran las imágenes intermedias obtenidas y para un PDF generado, las imágenes finales.

```
******************************
Processing guide 1
Saved image fd_001-1.jpg
******************************
Processing guide 2
Saved image fd_002-1.jpg
Saved image fd_002-2.jpg
******************************
Processing guide 3
Saved image fd_003-1.jpg
Saved image fd_003-2.jpg
******************************
Processing guide 4
Saved image fd_004-1.jpg
Saved image fd_004-2.jpg
******************************
Processing guide 5
```

Figura 28: Extracto del fichero PDFBox-Images.txt (PDF escaneados)

```
******************************
Processing guide 330
Saved image fd_330-1-1.jpg
Saved image fd_330-2-1.jpg
Saved image fd_330-2-2.jpg
Saved image fd_330-2-3.jpg
******************************
Processing guide 331
Saved image fd_331-1-1.jpg
Saved image fd_331-1-2.jpg
Saved image fd_331-1-3.jpg
Saved image fd_331-2-1.jpg
Saved image fd_331-2-2.jpg
Saved image fd_331-2-3.jpg
******************************
Processing guide 332
Saved image fd_332-1-1.jpg
Saved image fd_332-1-2.jpg
Saved image fd_332-1-3.jpg
Saved image fd_332-2-1.jpg
Saved image fd_332-2-2.jpg
******************************
Processing guide 333
```

Figura 29:Extracto del fichero PDFBox-Images.txt (PDF generados)

Para la extracción del texto, en el fichero Text-Extraction.txt (figura 30) se registran los documentos que se han almacenado en la base de datos, permitiendo ver si una guía ha podido ser procesada completamente. Además, en caso de tratarse de un PDF escaneado, se registran también las distintas imágenes finales extraídas.

```
******************************
Processing guide 1
Saved document (PDF): 60a78d909a842db6e6ee17ce
Saved image fd_001-1-1.jpg
Saved image fd_001-1-2.jpg
Saved image fd_001-1-3.jpg
Saved crops: 60a78d979a842db6e6ee17cf
Saved plague: 60a78d979a842db6e6ee17d0
******************************
Processing guide 2
Saved document (PDF): 60a78d979a842db6e6ee17d1
Saved image fd_002-1-1.jpg
Saved image fd_002-1-2.jpg
Saved image fd_002-2-1.jpg
Saved image fd_002-2-2.jpg
Saved crops: 60a78da49a842db6e6ee17d2
Saved plague: 60a78da49a842db6e6ee17d3
******************************
Processing guide 3
Saved document (PDF): 60a78da49a842db6e6ee17d4
Saved image fd_003-1-1.jpg
Saved image fd_003-1-2.jpg
Saved image fd_003-2-1.jpg
Saved crops: 60a78db19a842db6e6ee17d5
Saved plague: 60a78db19a842db6e6ee17d6
******************************
```

Figura 30: Extracto del fichero Text-Extraction.txt

Finalmente, para tener constancia del tiempo de ejecución de cada parte del sistema de extracción, se registran en el fichero Times.txt (figura 31).

```
Download: 315.13 s
Extract images with PDFBox: 39 s
Page to image with PDFBox: 5858 s
Text extraction and saving: 8085 s
```

Figura 31: Fichero Times.txt