



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

Sistema de reconocimiento
automático en arqueobotánica
Documentación Técnica



Presentado por Jaime Sagüillo Revilla
en Universidad de Burgos — 26 de abril de 2017
Tutor: Álgvar Arnaiz González, José Francisco Díez
Pastor y Virginia Ahedo García

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	IV
Apéndice A Planificación	1
A.1. Introducción	1
A.2. Planificación temporal	1
Apéndice B Especificación de Requisitos	10
B.1. Introducción	10
B.2. Objetivos generales	10
B.3. Catalogo de requisitos	10
B.4. Especificación de requisitos	10
Apéndice C Especificación de diseño	11
C.1. Diseño de las interfaces	11
Apéndice D Documentación técnica de programación	14
D.1. Introducción	14
D.2. Estructura de directorios	14
D.3. Manual del programador	14
D.4. Compilación, instalación y ejecución del proyecto	14
D.5. Pruebas del sistema	14
Apéndice E Documentación de usuario	15
E.1. Introducción	15
E.2. Requisitos de usuarios	15
E.3. Instalación	15

<i>ÍNDICE GENERAL</i>	II
E.4. Manual del usuario	16
Bibliografía	23

Índice de figuras

A.1. Burndown del <i>sprint</i> 0	2
A.2. Burndown del <i>sprint</i> 1	3
A.3. Burndown del <i>sprint</i> 2	4
A.4. Burndown del <i>sprint</i> 3	4
A.5. Burndown del <i>sprint</i> 4	5
A.6. Burndown del <i>sprint</i> 5	6
A.7. Burndown del <i>sprint</i> 6	7
A.8. Burndown del <i>sprint</i> 7	8
A.9. Burndown del <i>sprint</i> 8	8
C.1. <i>Jupyter Notebook</i> para el reonomiento de caras	12
C.2. Prototipo del etiquetador de imágenes	13
C.3. Etiquetador de imágenes	13
E.1. Ejecución del <i>notebook</i>	16
E.2. Selección de la vista del <i>notebook</i>	17
E.3. Etiquetador de imágenes	18
E.4. Parte derecha del etiquetador	19
E.5. Ventana de subida de ficheros	19
E.6. Etiquetador imágenes con una imagen cargada	20
E.7. Ejemplo de etiqueta	20
E.8. Notificación en la carga de una imagen	22
E.9. Notificación en la carga incorrecta de un fichero que no sea una imagen	22
E.10. Notificación en el guardado de etiquetas	22

Índice de tablas

Apéndice A

Planificación

A.1. Introducción

Para llevar a cabo este proyecto vamos a aplicar una metodología llamada Scrum. Scrum es una metodología de desarrollo software ágil, es decir, durante cada *sprint*¹, generalmente cada semana, se asignarán unas determinadas tareas a cumplimentar, con un producto como consecuencia de estas tareas. Al final de cada *sprint* se realizará una reunión junto a los tutores para validar los avances realizados y determinar las tareas a realizar durante el siguiente *sprint*.

A.2. Planificación temporal

En esta sección podremos ver la planificación del proyecto subdividida en *sprints*, como previamente comentaba. En cada uno de los sprints se detalla las tareas a realizar, algunos detalles descriptivos y un gráfico *burndown*.

Sprint 0

Estas son las tareas a realizar durante este *sprint* 0:

- Probar L^AT_EX.
- Gestor de tareas/versiones: *Github* y *Zenhub*.
- Instalar *Anaconda* y *Jupyter*.
- Leer los artículos propuestos por los tutores.
- Comenzar a probar algunos algoritmos de binarización.

¹*Sprint*: es el período en el cual se lleva a cabo el trabajo en sí.[1]

Como se puede ver las tareas a realizar son básicas, puesto que es el *sprint* 0 y es un *sprint* de mera adaptación al entorno de trabajo. La única tarea que supone un esfuerzo de comprensión mayor es la lectura de los artículos propuestos sobre trabajos relacionados o con una problemática similar a la nuestra. A continuación, en la figura A.1, se muestra el diagrama *burndown* de este *sprint*.



Figura A.1: Burndown del *sprint* 0

Sprint 1

Estas son las tareas a realizar durante esta *sprint* 1:

- Documentar lo realizado durante el *sprint* 0.
- Documentar lo que se irá realizando durante este *sprint* 1.
- Continuar probando con algoritmos de procesamiento de imágenes.
- Probar una aproximación con clasificadores al problema.

Puesto que en el *sprint* anterior no se documentó lo realizado, durante este se pretende documentar todo lo realizado durante el *sprint* anterior y este. Además de continuar probando con algoritmos de procesamiento de imágenes y comenzar a probar con la aproximación al problema mediante clasificadores.

En este *sprint* me vi desbordado de trabajo debido a la subestimación del esfuerzo a empeñar en las distintas tareas. No siendo capaz de comenzar a probar una aproximación con clasificadores. Por ello la tarea «Probar una aproximación con clasificadores al problema» se vio movida al siguiente *sprint*.

A continuación, en la figura A.2, se muestra el diagrama *burndown* de este *sprint*. El cual tiene dicho aspecto debido a que muchas de las tareas se

trabajaron de manera paralela, no siendo acabadas hasta el final del *sprint*. Y, además, algunas de las tareas no fueron cerradas cuando se debió, aspecto que se corregirá en los siguientes *sprints*.



Figura A.2: Burndown del *sprint* 1

Sprint 2

Estas son las tareas a realizar durante este *sprint* 2:

- Probar una aproximación con clasificadores al problema.
- Aplicación del método "Non maximum suppression" sobre el clasificado.

Puesto que la aproximación mediante reconocimiento de imágenes no reflejaba unos resultados muy positivos, durante la reunión mantenida con los tutores se decidió el uso de una técnica distinta. Nos referimos a la utilización de un clasificador, junto a un descriptor visual.

Debido a que todavía no se poseían suficientes imágenes para el estudio del problema mediante esta técnica, lo que se decidió es aplicarla sobre otro problema de características similares, como es el reconocimiento de caras en imágenes. Con unos resultados bastante positivos debido a distintos razonamientos explicados en la Memoria, sección de Aspectos relevantes del proyecto.

A continuación, en la figura A.3, se muestra el diagrama *burndown* de este *sprint*.

Sprint 3

Estas son las tareas a realizar durante este *sprint* 3:

- Reorganizar los *Jupyter Notebooks*.

Figura A.3: Burndown del *sprint 2*

- Probar distintos clasificadores y métricas.
- Enviar fotos rotadas al clasificador.

Durante este *sprint*, primero, se reorganizó la estructura del proyecto. Aportando mucho más orden y claridad a nuestro proyecto. Después, se introdujeron múltiples clasificadores y métricas, los cuales introduciré en mayor medida en la memoria, como *Random Forest* o *Gradient tree boosting*. Por último, se enviaron imágenes rotadas al clasificador, con el fin de poder analizar una posible problemática.

A continuación, en la figura A.4, se muestra el diagrama *burndown* de este *sprint*.

Figura A.4: Burndown del *sprint 3*

Sprint 4

Estas son las tareas a realizar durante este *sprint* 4:

- Implementación de *Data Augmentation* en nuestro conjunto de entrenamiento.
- Implementación de controles de usuario.

Durante este *sprint* se aplicó en nuestro conjunto de entrenamiento la técnica *Data augmentation*. Esta técnica nos permitió aumentar el tamaño de nuestro conjunto de entrenamiento enormemente.

Además, se realizó un *notebook*², con controles de usuario, los cuales nos permiten escoger entre clasificadores, imágenes y probabilidades. Permitiendo la continua interacción entre el usuario y la clasificación de una imagen, sin la necesidad de modificar el código por parte del usuario del notebook para cambiar entre las distintas opciones.

A continuación, en la figura A.5, se muestra el diagrama *burndown* de este *sprint*.



Figura A.5: Burndown del *sprint* 4

Sprint 5

Estas son las tareas a realizar durante este *sprint* 5:

- Implementar un *file chooser*
- Añadir más clasificadores.

²Siempre que nos referimos a un *notebook*, a lo que nos referimos es a un *Jupyter Notebook*

- Correcciones en la documentación.
- Estudiar como implementar un etiquetador de imágenes.

Durante este *sprint* se añadieron los clasificadores que deseábamos, es decir, un clasificador bayesiano y un clasificador mediante regresión logística. Además, se añadió un *file chooser* que nos permitiría, desde ese momento, escoger la imagen que deseemos dentro de nuestro sistema operativo. En cuanto a la documentación, se corrigió toda la realizada hasta ese momento. Y, por último, se hizo un estudio básico sobre como implementar un etiquetador de imágenes mediante un *Widget* de *Python*. Aunque, esta última tarea no tuviese ningún producto resultante en este *sprint*.

A continuación, en la figura A.6, se muestra el diagrama *burndown* de este *sprint*.



Figura A.6: Burndown del *sprint* 5

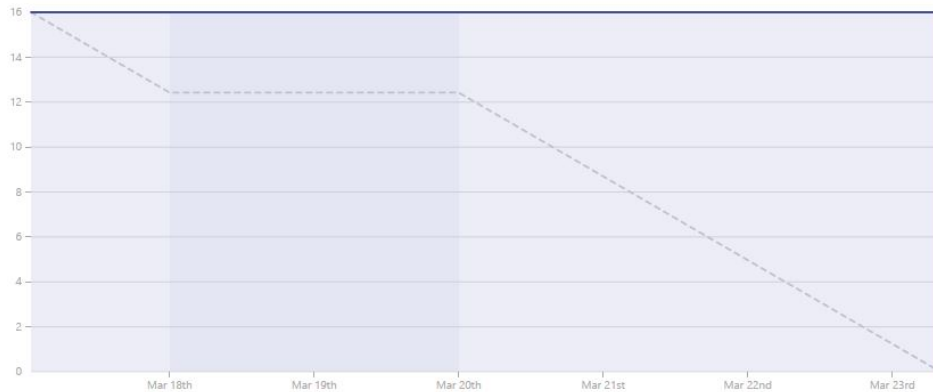
Sprint 6

Estas son las tareas a realizar durante este *sprint* 6:

- Estudiar los *Widgets* personalizados de *Jupyter Notebook* e *Ipython*.

Aunque este *sprint* se encuentre compuesto por una única tarea, no era menos complejo por ello. El objetivo de este *sprint* era obtener un *Widget* capaz de etiquetar imágenes. Pero en la realización de este se encontraron múltiples problemas. Obteniendo como producto resultante tres posibles alternativas con aspectos a corregir.

Por lo tanto, en la figura A.7 mostramos el diagrama *burndown*, poco esclarecedor, de este *sprint*.

Figura A.7: Burndown del *sprint* 6

Sprint 7

Estas son las tareas a realizar durante este *sprint* 7:

- Estudiar *Bag of Words*.
- Añadir mayor parametrización al *Jupyter Notebook UI*.
- Corregir *bugs* del Widget previamente implementado.

Durante este *sprint* se consiguió, en primer lugar, corregir una de las alternativas del etiquetador de imágenes, o *Widget*, desarrolladas durante el *sprint* anterior. Además, se corrigieron y añadieron múltiples parámetros en el *Jupyter Notebook UI* y en las clases utilizadas por este *Notebook*. Y, por ultimo, se realizó un estudio sobre un modelo ampliamente usado para tareas de clasificación, llamado *Bag of Words*.

A continuación, en la figura A.8, se muestra el diagrama *burndown* de este *sprint*.

Sprint 8

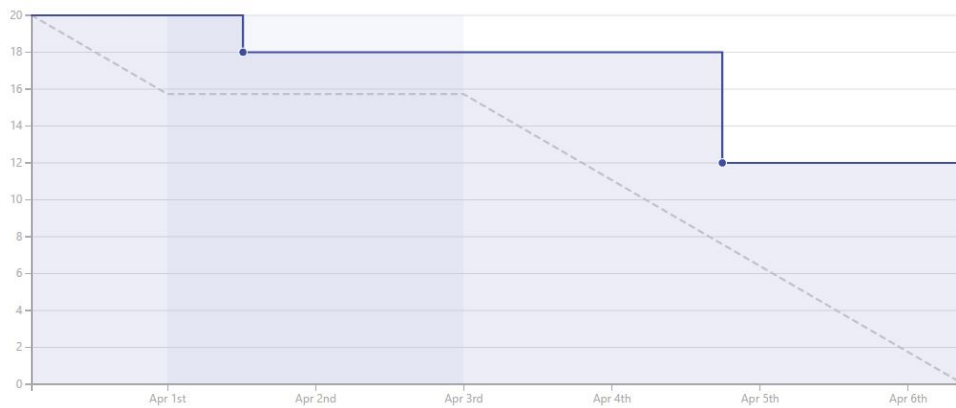
Estas son las tareas a realizar durante este *sprint* 8:

- Implementar la funcionalidad de obtención de imágenes en el etiquetador de imágenes, o *Widget*.
- Mejorar la interfaz del etiquetador de imágenes.
- Crear un primer prototipo de interfaz de usuario.

Figura A.8: Burndown del *sprint* 7

Durante este *sprint* se realizó un primer prototipo de interfaz de usuario. Partiendo de este prototipo, se mejoró la interfaz del etiquetador de imágenes. Consiguiendo, así, una interfaz adecuada para el cliente. Además, se implementó la funcionalidad que nos permitiría obtener una imagen resultante de cada etiqueta realizada en las distintas imágenes.

A continuación, en la figura A.9, se muestra el diagrama *burndown* de este *sprint*.

Figura A.9: Burndown del *sprint* 8

***Sprint* 9**

Este *sprint* tendrá una duración de dos semanas. Debido a la carga de trabajo asociada a este *sprint* y al ser días no lectivos por las vacaciones de Semana Santa.

Estas son las tareas a realizar durante este *sprint* 9:

- Añadir un texto a cada etiqueta que realizamos en una imagen.
- Añadir notificaciones al usuario en la carga y guardado de imágenes.
- Guardar las coordenadas de las etiquetas de cada imagen.
- Cargar las etiquetas de una imagen que haya sido previamente etiquetada.
- Controlar que el usuario no cree etiquetas en el SVG pero fuera de la imagen.
- Añadir la posibilidad de eliminar etiquetas previamente realizadas.
- Corregir los notebooks creados para la técnica Bag of Words.

Apéndice B

Especificación de Requisitos

- B.1. Introducción
- B.2. Objetivos generales
- B.3. Catalogo de requisitos
- B.4. Especificación de requisitos

Especificación de diseño

C.1. Diseño de las interfaces

En esta sección se explicarán las diferentes interfaces de los productos realizados en este trabajo fin de grado.

Jupyter Notebook para el reconocimiento de caras

Durante el primer mes se trabajó en un *Jupyter Notebook* que trataba de reconocer caras mediante un clasificador junto a un extractor de características, como ya hemos explicado en secciones anteriores. Con este *Jupyter Notebook* se trataba de facilitar la evaluación del rendimiento de los clasificadores y el cambio de las distintas variables de manera interactiva.

Por lo tanto, la interfaz de este *Jupyter Notebook* no ha sido trabajada para su uso por parte del usuario. Sino que fue creada para un uso más de experimentación. Es por ello que la interfaz no tiene un buen grado de usabilidad como podemos observar en la figura C.1.

Etiquetador de imágenes

La interfaz del etiquetador de imágenes ha sido desarrollada partiendo del prototipo mostrado en la ilustración C.2. Tratando de crear una interfaz lo más simple e intuitiva para el usuario.

El resultado tras la implementación de este producto es el mostrado en la C.3. Obteniendo una interfaz muy similar a la prototipada en un principio. Pero añadiendo algún elemento más, para facilitar su uso por parte del usuario.



Figura C.1: *Jupyter Notebook* para el reconocimiento de caras

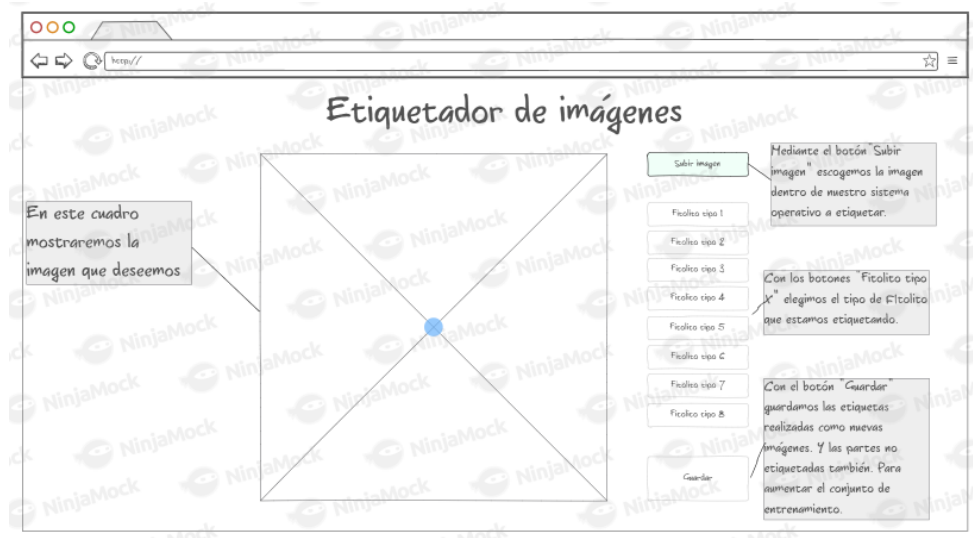


Figura C.2: Prototipo del etiquetador de imágenes

Etiquetador de imágenes



Figura C.3: Etiquetador de imágenes

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema

Documentación de usuario

E.1. Introducción

En esta sección se indicarán los distintos prerequisites de la herramienta. Así, como toda la información necesaria para la correcta utilización de la herramienta. O la solución ante posibles problemas en su utilización.

E.2. Requisitos de usuarios

El requisito básico necesario, por parte del usuario, es disponer de un ordenador con un sistema operativo moderno. Pudiendo, así, llevar a cabo la instalación que será explicada en la siguiente sección.

E.3. Instalación

Los requisitos necesarios son:

- Anaconda¹.
- IPython File Upload. Siguiendo los pasos de instalación en <https://github.com/peteut/ipython-file-upload>.
- Jupyter Dashboards. Siguiendo los pasos de instalación en <https://github.com/jupyter/dashboards>.
- Este proyecto descargado o clonado. Cuyo enlace al repositorio es <https://github.com/jasag/Phytoliths-recognition-system>.

¹Es altamente recomendable utilizar *Anaconda* para la instalación de *Python* junto a algunos paquetes necesarios. Puesto que la instalación aislada de *Python* llevaría al usuario a tener que instalar manualmente múltiples paquetes que ya vienen preinstalados con *Anaconda*.

E.4. Manual del usuario

En esta sección se explicarán los distintos aspectos a tener en cuenta en el uso de los distintos productos.

Manual del usuario: etiquetador de imágenes

Una vez completados satisfactoriamente los pasos anteriores, ejecutamos *Jupyter Notebook* desde *Anaconda*². Y desde esta aplicación, abrimos el *notebook Image_Labeler.ipynb*, en la carpeta *code/notebooks* dentro del repositorio previamente descargado.

Con *Image_Labeler.ipynb* ya abierto tendremos que llevar a cabo dos pasos:

1. Ejecutar todas las celdas del *notebook*.
2. Activar *Dashboard Preview*.

Para llevar a cabo los dos siguientes pasos basta con navegar por la barra del *notebook*, como muestro a continuación.

Para la ejecución del primer paso, navegamos por *Cell* y clicamos en *Run All*. Como se puede observar en la figura E.1.

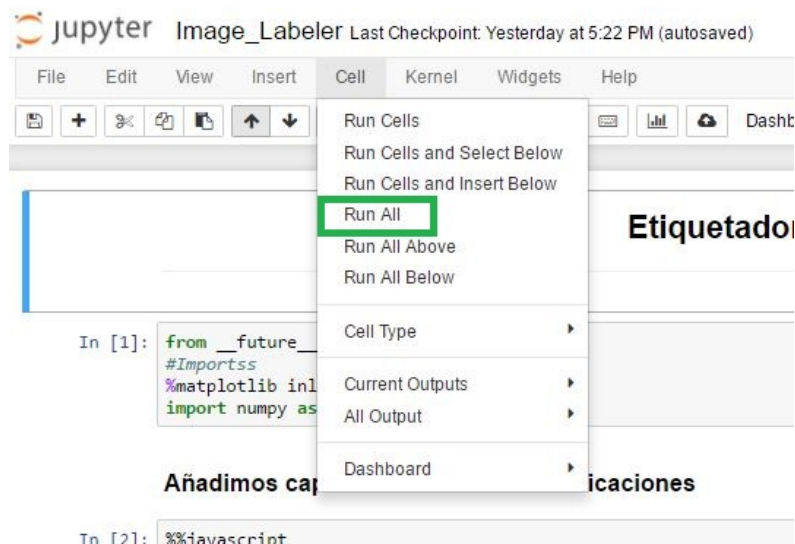
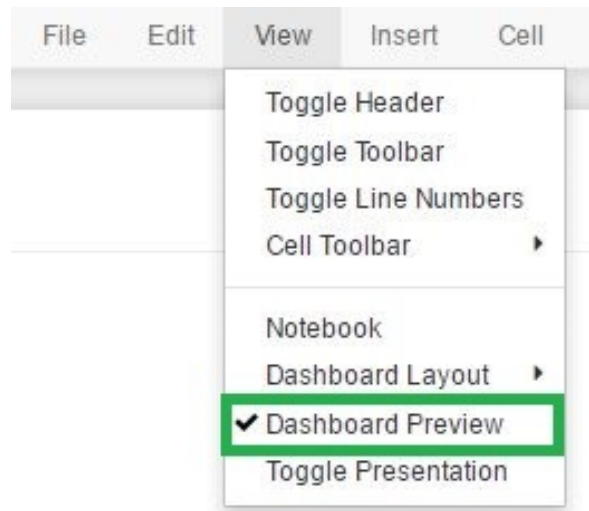


Figura E.1: Ejecución del *notebook*

Para la ejecución del segundo paso, navegamos por *View* y clicamos en *Dashboard Preview* E.2.

² Si es la primera vez que utilizas Anaconda es recomendable utilizar Anaconda Navigator. Ya que este nos facilitará ejecutar *Jupyter Notebook*, sin el uso de la línea de comandos.

Figura E.2: Selección de la vista del *notebook*

Una vez realizados los pasos anteriores, tendremos como resultado la figura [E.3](#).

Uso del etiquetador

Como resultado de los pasos previos, el etiquetador de imágenes estará listo para su funcionamiento.

El etiquetador está compuesto por dos partes principales:

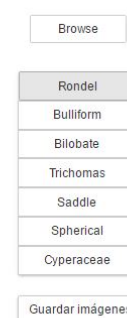
- Parte izquierda: imagen.
- Parte derecha: «botones».

La parte izquierda mostrará la imagen a ser etiquetada. Y la parte derecha mostrará los botones que nos permitirán realizar distintas elecciones. Como podemos ver en la figura [E.3](#).

Parte derecha del etiquetador

La parte derecha del etiquetador esta compuesta por tres grupos de botones. El botón para la carga de una imagen, los botones de selección de fitolito y el botón de guardado. Como podemos observar en la figura [E.4](#).

En las secciones siguientes explicaré sus funciones.

Etiquetador de imágenes

Browse
Rondel
Bulliform
Bilobate
Trichomas
Saddle
Spherical
Cyperaceae
Guardar imágenes

Figura E.3: Etiquetador de imágenes

Carga de imagen

Para etiquetar una imagen el primer paso será escoger una imagen en nuestro ordenador mediante el botón *Browse*, situado en la esquina superior derecha del etiquetador. Como se puede ver en la figura E.3.

Una vez pulsado, se mostrará una ventana, como la que podemos ver en la figura E.5, en la que escogeremos la imagen que deseemos.

Tras escoger la imagen, esta se cargará en la parte izquierda del etiquetador, dando lugar a algo similar a lo mostrado en la figura E.6.

Etiquetar distintos tipos de fitolitos

Antes de explicar como se crean las etiquetas, debemos de tener en cuenta los botones para la selección de tipos de fitolito.

Cada vez que se quiera etiquetar un tipo de fitolito distinto tendremos que clicar en el botón correspondiente al tipo de fitolito objetivo.

A su vez, y para facilitarnos distinguir a que tipo de fitolito pertenece cada etiqueta, se escribe encima de cada etiqueta el tipo de fitolito que ha sido etiquetado, como podemos ver en la figura E.7.



Figura E.4: Parte derecha del etiquetador

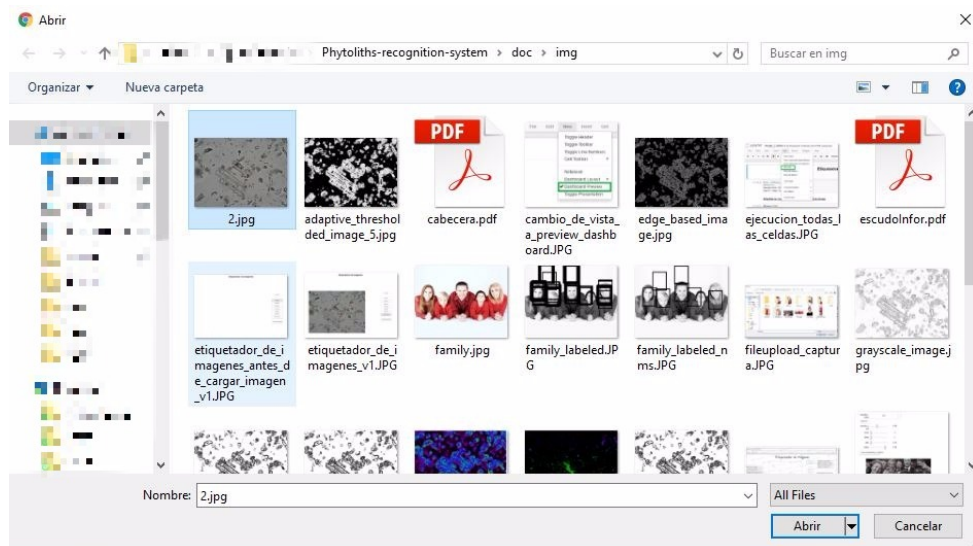


Figura E.5: Ventana de subida de ficheros

Realizar etiquetas

Una vez cargada la imagen en el etiquetador y comprendido el uso de los botones de selección de tipo de fitolito, podemos comenzar a etiquetar fitolitos. Para etiquetar basta con clicar en la imagen, mover el ratón hacia donde deseemos y volver a clicar. Para, así, definir una etiqueta como la que se puede ver en la figura E.7.

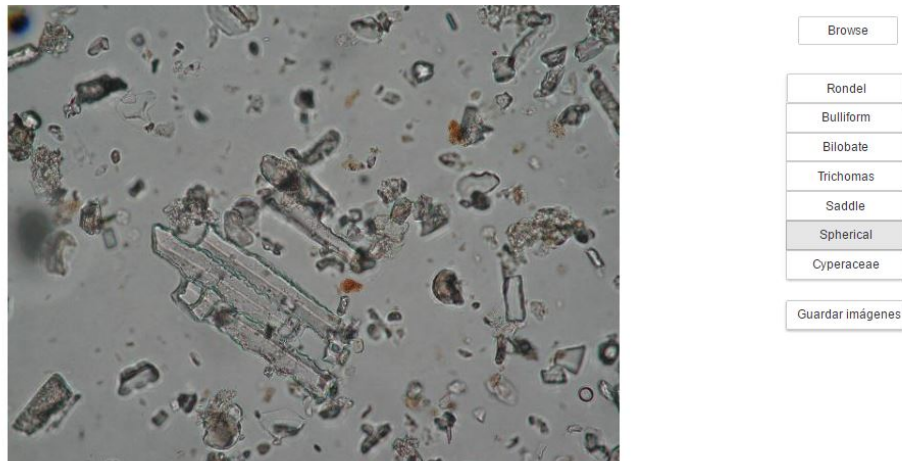
Etiquetador de imágenes

Figura E.6: Etiquetador imágenes con una imagen cargada

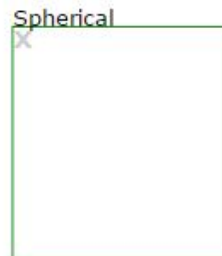


Figura E.7: Ejemplo de etiqueta

Eliminar etiquetas

Como es normal, un usuario puede equivocarse realizando una etiqueta que no debería. Por ello, en la esquina superior izquierda de cada etiqueta, disponemos de una «X», como podemos observar en la figura E.7, que nos permitirá eliminar etiquetas previamente realizadas. Posibilitando así revertir un error del usuario.

Guardar etiquetas

Después de realizar las etiquetas, y finalizar de etiquetar una imagen, solo queda guardar las etiquetas. Para ello basta con clicar el botón «Guardar imágenes» en la esquina inferior derecha del etiquetador, véase [E.3](#).

Es **importante** destacar que no sirve de nada etiquetar una imagen, sino pulsamos más tarde en el botón guardar. Debido a que las etiquetas realizadas no habrán sido guardadas. Y, por lo tanto, todas las etiquetas realizadas sobre una imagen se perderán.

Etiquetas realizadas fuera de la imagen

En la versión actual del etiquetador, este te permite realizar etiquetas fuera de la imagen. Pero estas etiquetas no serán guardadas.

Por lo tanto, debemos de tener en cuenta que las etiquetas que no tengan todas sus aristas dentro de la imagen no se guardan.

Aun, así, en la siguiente sección explicaré como comprobar que las etiquetas previamente realizadas han sido guardadas.

Carga de etiquetas previamente realizadas

Las imágenes etiquetadas se guardan en la carpeta `/code/rsc/img` dentro del repositorio. Al cargar imágenes que se encuentren dentro de esta carpeta, las etiquetas previamente realizadas sobre estas imágenes se cargarán con ellas³. Pudiendo ser modificadas, eliminándolas o creando nuevas etiquetas.

Notificaciones

Se han incluido en el etiquetador una serie de notificaciones, las cuales nos permitirán cercionarnos de distintos eventos.

Las notificaciones que se muestran son las siguientes:

1. Notificación en la carga de una imagen. Véase la figura [E.8](#).
2. Notificación en la carga incorrecta de un fichero que no sea una imagen. Véase la figura [E.9](#).
3. Notificación en el guardado de etiquetas. Véase la figura [E.10](#).

Gracias a estas notificaciones podremos saber si la imagen que hemos seleccionado era correcta, o no, y si las etiquetas ya han sido guardadas.

³Notese que los nombres de las imágenes no son los cargados originalmente, ya que todas las imágenes etiquetadas son renombradas para no sobrescribir imágenes previamente etiquetadas con nombres iguales.

A green rectangular notification box with a thin border. It contains the text "La imagen 2.jpg ha sido cargada correctamente!" in a dark green font. To the right of the text is a small, faint "x" icon for closing the notification.

La imagen 2.jpg ha sido cargada correctamente!

Figura E.8: Notificación en la carga de una imagen

A pink rectangular notification box with a thin border. It contains the text "El fichero cabecera.pdf no tiene formato de imagen. Ejemplo: jpeg, jpg, tif, tiff, png" in a dark pink font. To the right of the text is a small, faint "x" icon for closing the notification.

El fichero cabecera.pdf no tiene formato de imagen. Ejemplo: jpeg, jpg, tif, tiff, png

Figura E.9: Notificación en la carga incorrecta de un fichero que no sea una imagen

A green rectangular notification box with a thin border. It contains the text "Las etiquetas para la imagen 2.jpg han sido guardadas correctamente" in a dark green font. To the right of the text is a small, faint "x" icon for closing the notification.

Las etiquetas para la imagen 2.jpg han sido guardadas correctamente

Figura E.10: Notificación en el guardado de etiquetas

Bloqueo del etiquetador

En algunas ocasiones, y por razones que desconozco hasta el momento, el etiquetador puede bloquearse alguna vez al cargar imágenes previamente etiquetadas. Para solucionar este problema basta con recargar el etiquetador, como se vio en la parte inicial de esta sección.

Bibliografía

- [1] Wikipedia. Scrum — Wikipedia, the free encyclopedia, 2017. [Online; accedido 12-Marzo-2017].