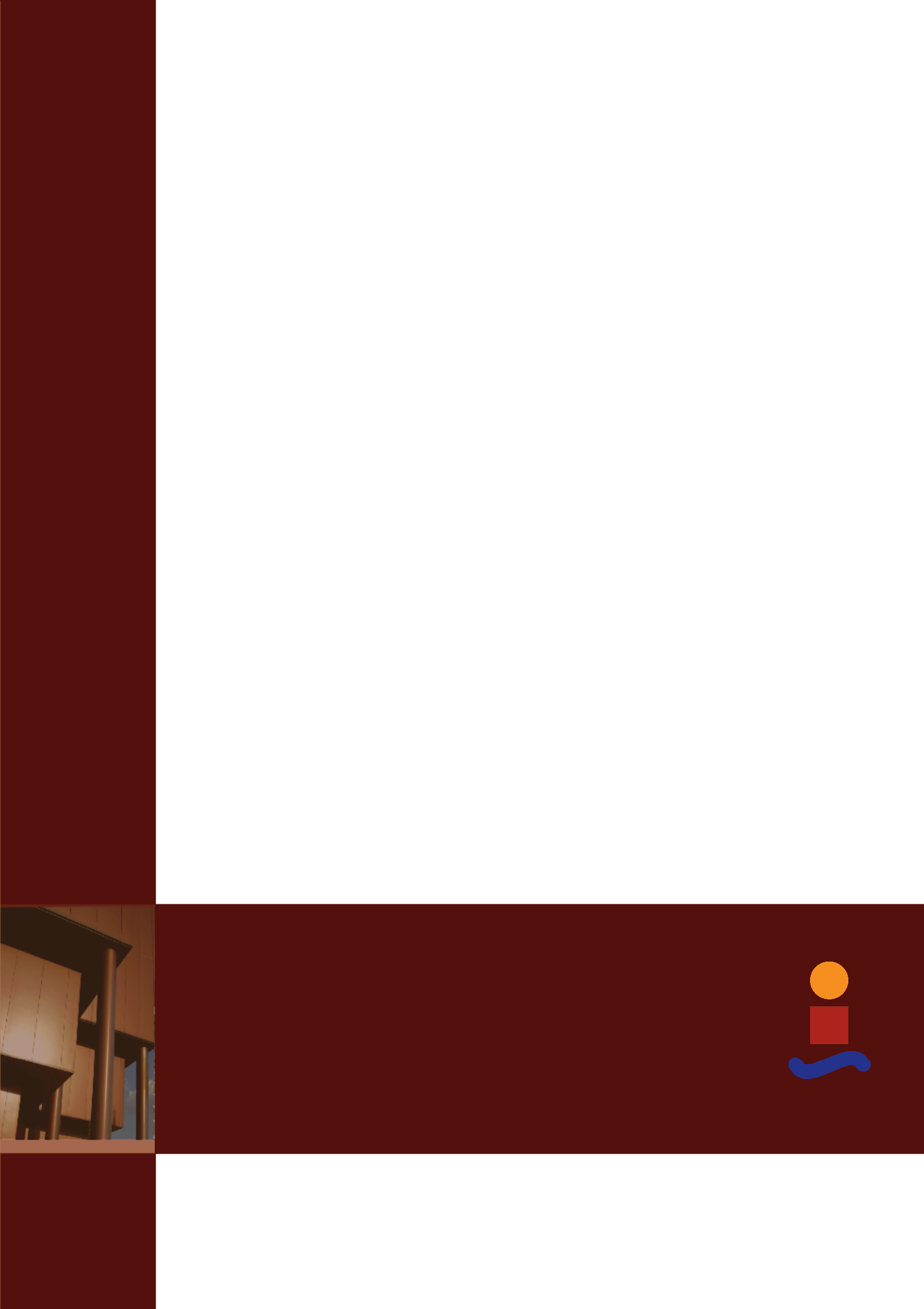
­

Autor: Miguel Ángel Rodrigo de Lisbona

Tutor: Daniel Limón Marruedo

Reconocimiento de Sellos en “Archivo Histórico de Osborne”

Trabajo de Fin de Grado

Ingeniería de Tecnologías Industriales

Dep. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Trabajo de Fin de Grado

Ingeniería de Tecnologías Industriales

**Reconocimiento de Sellos en “Archivo Histórico de Osborne”**

Autor:

Miguel Ángel Rodrigo de Lisbona

Tutor:

Daniel Limón Marruedo

Profesor titular

Dep. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Proyecto Fin de Carrera: Reconocimiento de Sellos en “Archivo Histórico de Osborne”

|  |  |
| --- | --- |
| Autor: | Miguel Ángel Rodrigo de Lisbona |
| Tutor: | Daniel Limón Marruedo |

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal

*A mi familia*

*A mis maestros*

Agradecimientos

Los estilos adoptados por nuestra Escuela y utilizada en este texto es una versión y adaptación a Word® del la versión LATEX que el Prof. Payán realizó para un libro que desde hace tiempo viene escribiendo para su asignatura. Por ello, la Escuela le está agradecida. Por otro lado, la adaptación se hizo sobre un formato que el prof. Aguilera arregló, basándose en su tesis doctoral. Su aportación ha sido muy relevante para que este formato vea la luz. Esta adaptación la llevamos a cabo el alumno Silvio Fernández, becario del Centro de Cálculo, y yo mismo, sobre un trabajo preliminar del alumno Julián José Pérez Arias.

A esta hoja de estilos se le incluyó unos nuevos diseños de portada. El diseño gráfico de las portadas para proyectos fin de grado, carrera y máster, está basado en el que el prof. Fernando García García, de la Facultad de Bellas Artes de nuestra Universidad, hiciera para los libros, o tesis, de la sección de publicación de nuestra Escuela. Nuestra Escuela le agradece que pusiera su arte y su trabajo a nuestra disposición.

*Juan José Murillo Fuentes*

*Subdirección de Comunicaciones y Recursos Comunes*

*Sevilla, 2013*

Resumen

La Fundación Osborne posee un archivo de documentos relacionados con la empresa que datan desde principios de 1800. Previas investigaciones han permitido descubrir información sobre la vida de Tolkien en el archivo. Se sabe además, que existe correspondencia con casas reales o el Vaticano. También existe correspondencia de personajes públicos como Washington Irving. Los inicios de la empresa están estrechamente relacionados con otros personajes históricos como Fernán Caballero, cuñada del fundador, o “Frasquita” Larrea, su suegra. Por ello, podría haber información importante por descubrir sobre dichos personajes que podría ser de valor histórico.

Para extraer dicha información, se ha optado por desarrollar una web donde se suben los archivos escaneados junto con su transcripción. Además de ello, se quiere desarrollar un sistema que realice dichas transcripciones de manera autónoma mediante detección de los caracteres manuscritos.

En este trabajo se describe el desarrollo de un sistema que reconozca los escudos en las cartas para clasificar su autor de manera automática. Además, el sistema debe calcular la posición de dichos escudos con el objetivo de eliminarlos. Esto es necesario ya que los sistemas de detección de caracteres necesitan que los elementos que no son texto sean eliminados del documento antes de empezar cualquier análisis.

Abstract

Fundación Osborne owns an archive of documents related to the company dated up to early 1800. Previous investigations on the archive have revealed information about the life of Tolkien. Moreover, it is known that the archive has correspondence with royal houses as well as The Vatican. There are also letters of historical characters such as Fernán Caballero – founder’s sister-in-law – or “Frasquita” Larrea – his mother-in-law. For those reasons, it is reasonable to think that there might be some historical value on the documents that remain not investigated.

In order to retrieve that information, there is a website in development where scanned documents will be uploaded with their respective transcription and metadata. Those transcriptions are currently being typed manually. A system to make those transcriptions automated is being developed too by recognizing handwritten characters on the documents.

The purpose of this work is to describe the development of a system which recognizes seals on letters to automatically classify them by author. Moreover, the system calculates the position of such seals to delete them from the document. This is a need because character recognition system needs every non-text element removed from the image before any analysis is performed.

Índice

Agradecimientos ix

Resumen xi

Abstract xiii

Índice xiv

Índice de Tablas xvi

Índice de Figuras xviii

Notación xx

1 La importancia de preservar nuestra historia 1

2 Descripción del problema 1

3 Construcción de un ground truth 1

4 Esquema general de un ocr 1

Binarización 1

4.1.1 Elección manual de umbral 1

4.1.2 Método de Otsu 1

4.1.3 Método adaptativo 3

Segmentación 3

4.1.4 Segmentación mediante enfoque frecuencial 3

4.1.5 Extracción y descripción de características 6

4.1.6 Segmentación mediante componentes conexas 8

4.1.7 Conclusiones sobre segmentación 10

4.1.8 Mención de honor a segmentación mediante transformada de Hough 10

Acerca de los otros pasos de un OCR estándar. Separación de líneas y palabras y sistema de reconocimiento 10

5 Elaboración de un *ground truth* 11

Preliminares 11

5.1.1 Cubierta 11

5.1.2 Portada 11

5.1.3 Resumen 11

5.1.4 Prefacio o Introducción 11

5.1.5 Índice y/o Índice general 11

5.1.6 Lista de ilustraciones y tablas 12

5.1.7 Lista de abreviaturas y símbolos 12

5.1.8 Glosario 12

Texto principal 12

Bibliografía 12

Anexos 12

6 Uso de Estilos 13

Secciones 13

6.1.1 Subsección 13

6.1.2 Otra subsección 13

Otra sección 14

7 Otro Capítulo 11

Estilos de un documento 11

Títulos y Referencias Cruzadas 12

Versiones y Sistemas Operativos 12

7.1.1 Macintosh 12

7.1.2 Linux 12

Texto en inglés 13

Elementos básicos de un libro 13

Símbolos y fórmulas 13

Ecuaciones y MathType® 13

7.1.3 Fuentes 14

7.1.4 Epígrafes o citas célebres 14

7.1.5 Figuras y tablas 15

7.1.6 Hiperenlaces 15

7.1.7 Tabla de contenido 16

7.1.8 Índice de figuras, tablas y otros elementos 16

7.1.9 Formatos de páginas 16

7.1.10 Teoremas y otros elementos similares 16

7.1.11 Ejemplos 17

7.1.12 Índices de palabras y glosarios 17

Antes del documento 17

Fuente del texto 18

Cubierta 18

Apéndices 11

Apéndice A. Código Ground Truth GUI 11

main.py 11

win\_pygame.py 12

pygame\_functions.py 14

win\_tkinter.py 15

widgets.py 18

database.py 19

seal.py 20

Referencias 11

Índice de Conceptos 13

Glosario 15

# **Índice de Tablas**

Tabla 2–1. Tipos de transmisión y frecuencia central 8

Tabla 3–1 Tipos de transmisión y frecuencia central 15

# **Índice de Figuras**

Figura 2‑1. Esto es el pie de la figura. 8

Figura 3‑1. Pie de figura 15

Notación

|  |  |
| --- | --- |
| A\* | Conjugado |
| c.t.p. | En casi todos los puntos |
| c.q.d. | Como queríamos demostrar |
| ∎ | Como queríamos demostrar |
| e.o.c. | En cualquier otro caso |
| e | número e |
| IRe | Parte real |
| IIm | Parte imaginaria |
| sen | Función seno |
| tg | Función tangente |
| arctg | Función arco tangente |
| sen | Función seno |
| sin*xy* | Función seno de *x* elevado a *y* |
| cos*xy* | Función coseno de *x* elevado a *y* |
| Sa | Función sampling |
| sgn | Función signo |
| rect | Función rectángulo |
| Sinc | Función sinc |
| ∂y ∂x  *x*◦ | Derivada parcial de *y* respecto  Notación de grado, *x* grados. |
| Pr(*A*) | Probabilidad del suceso *A* |
| SNR | Signal-to-noise ratio |
| MSE | Minimum square error |
| : | Tal que |
| < | Menor o igual |
| > | Mayor o igual |
| \ | Backslash |
| ⇔ | Si y sólo si |

# La importancia de preservar nuestra historia

Un pueblo sin conocimientos de su historia, origen y cultura es como un árbol sin raíces.

- Marcus Garvey -

L

a Historia. Alrededor de 6000 años de documentos escritos, miles de generaciones plasmando pensamientos y vivencias para aquellos que aún están por venir, haciendo posible el ciclo motor del progreso de todas las sociedades: intentar, fracasar, aprender y repetir. Parafraseando a Warren Buffet, si hay algo más importante que aprender de los errores de uno mismo, es aprender de los errores de los demás. Vivimos en un mundo en el que todos aparentan (o incluso creen) saber qué hacer para resolver los problemas. Pero cuando dos personas tienen opiniones incompatibles, al menos uno debe estar equivocado. La realidad es que ni políticos, ni científicos, ni guías espirituales de ningún tipo saben con absoluta certeza la mejor solución a todo. Es aquí donde la historia nos proporciona la más valiosa de las fuentes de conocimiento para permitir tanto a sociedades como a individuos tomar decisiones un poco menos a ciegas. Nadie duda de la importancia de la investigación, preservación y difusión de la historia además de garantizar la no distorsión de la misma. Por todo ello, la digitalización y catalogación de la totalidad de textos escritos por la humanidad a lo largo de su paso por este mundo tendría un incuestionable gran valor. Realizar esta tarea de manera manual es evidentemente inviable. Es por ello que existen multitud de líneas de investigación a lo largo y ancho del globo con el objetivo de conseguir automatizar esta tarea.

Este trabajo describe parte del proyecto llevado a cabo por Talemtum Lab Osborne mediante la colaboración de la Fundación Osborne[[1]](#footnote-2), Telefónica Talemtum Labs[[2]](#footnote-3) y la Fundación SEPI[[3]](#footnote-4). El objetivo del proyecto es la digitalización del archivo histórico que posee el Grupo Osborne almacenado en sus primeras bodegas en El Puerto de Santa María. Dicho archivo contiene, por un lado, facturas, notas de pago y demás documentación de carácter administrativo y, por otro, correspondencia entre miembros de la familia y con otros personajes públicos de la época de la fundación de la empresa en 1772. Entre estos personajes se encuentran Washington Irving, J.R.R. Tolkien o Isaac Peral, quien ofrece sus servicios a la empresa en una de sus cartas, así como miembros de distintas casas reales o El Vaticano.

El archivo ya ha sido utilizado en una reciente investigación sobre la biografía de Tolkien, muy relacionado con la familia Osborne al ser criado y educado por uno de sus miembros, y existen múltiples indicios de que podrían existir en él más documentos de importante valor histórico. La suegra de Thomas Osborne Mann, fundador de la empresa, fue Francisca Javiera Ruiz de Larrea y Aherán, **Frasquita Larrea**. Esta señora fue una escritora gaditana impulsora de la tertulia literaria que se cree que tuvo una gran influencia en la Constitución de 1812. La cuñada de Thomas fue la escritora Cecilia Böhl de Faber y Larrea, **Fernán Caballero**.

Gran parte de la documentación que se posee tiene un contenido desconocido, de ahí nace la motivación por digitalizarlo para permitir agilizar su investigación. Para ello, se desarrolla una página web donde se muestran los documentos escaneados junto con su transcripción. Esta transcripción es necesaria por un doble motivo: por un lado, los documentos tienen en su mayoría una caligrafía de muy difícil reconocimiento lo cual ralentiza enormemente la lectura e interpretación. Por otro lado, si se quiere realizar algún tipo de búsqueda en función de alguna palabra que forme parte del texto, sería imposible sin transcripción previa.

# Descripción del problema

Somos como enanos a los hombros de gigantes. Podemos ver más, y más lejos que ellos, no porque la agudeza de nuestra vista ni por la altura de nuestro cuerpo, sino porque somos levantados por su gran altura.

- Bernard de Chartres -

E

ste trabajo forma parte de un proyecto mayor que pretende extraer información de los documentos del archivo descrito mediante técnicas de visión artificial. El problema al que se dedica esta memoria consiste en localizar la posición de posibles escudos o sellos en correspondencia con un doble objetivo:

* Localizar la posición de dicho sello con el objetivo de eliminarlo de la imagen y que no suponga un problema para otros sistemas de reconocimiento, como por ejemplo la transcripción automatizada del texto.
* Reconocer de qué sello se trata y clasificar el documento en cuestión de acuerdo con la información disponible de dicho sello.

Algunos ejemplos de documentos se muestran a continuación.

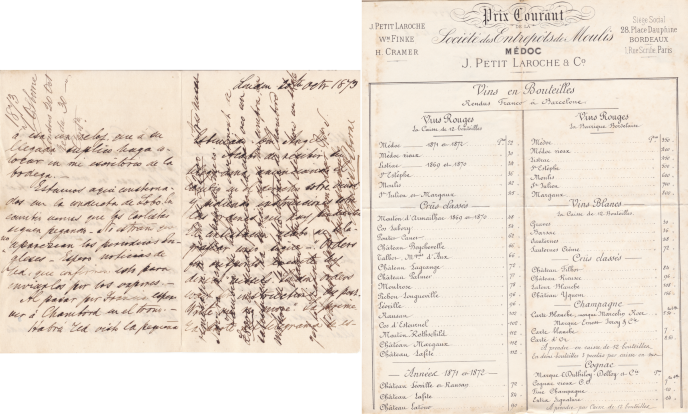


Figura X. Ejemplo de tipos de documentos que pueden presentarse

En la figura anterior pueden apreciarse dos tipos de documentos distintos. Uno presenta texto en vertical además del texto horizontal. Esto es una gran dificultad añadida como se verá más adelante, ya que el texto en diferentes direcciones que intersecta se detecta como un único objeto en la imagen.



Figura X. Dos ejemplos más de documentos

Puede verse en el documento de la izquierda como existen varias manchas además de las marcas de pliegues oscurecidas por el envejecimiento. Todos estos elementos dificultan la detección. En concreto, la mancha roja es particularmente difícil de diferenciar de un sello.

El caso de la derecha representa la situación ideal, donde el documento presenta un sello claramente separado del texto, mientras que el resto del papel apenas presenta envejecimiento o imperfección alguna.

El problema del reconocimiento automático de elementos en documentos escritos, ya sean caracteres, firmas, logos o cualquier otro elemento, es un tema de gran interés, y por lo tanto, existe una extensa bibliografía con muy diversos métodos para atajar el problema. En (Seal detection and recognition:…) se estudia cómo extraer y reconocer sellos estampados con el objetivo de clasificar documentos. En un primer paso se realiza una segmentación mediante una Transformada de Hough (término) para círculos para hallar regiones de sellos. Después se detectan los caracteres aislados hallados en las regiones se aplica un sistema de clasificación basado en características invariantes a la rotación y la escala. Después se utiliza la información de la posición relativa de cada pareja de caracteres para clasificar el sello. En nuestro caso, los sellos carecen de texto alguno, por lo tanto no pueden clasificarse así. Sin embargo, el método de segmentación (término) sí que podría resultar de interés.

En (Signature Detection and Matching for Document Image Retrieval) se estudia un método para la segmentación y clasificado de firmas. El algoritmo propuesto extrae el contorno de las regiones de las firmas y utiliza información de la posición relativa de los puntos de ese contorno para la clasificación. En nuestro caso, el deterioro del papel con el paso de los años, además de que los sellos tienen forma mucho más compleja que tan solo trazos hacen que la detección de contornos no siempre devuelva el mismo resultado para el mismo sello.

Nandedkar, Mukhopadhyay y Sural (Spectral approach blabla…) aplican un método mucho más sencillo en el cual utilizan una convolución gaussiana a la imagen como filtro de paso bajo para separar texto de logotipos. Esto se basa en el hecho de que el texto añade componentes de alta frecuencia al espectro de una imagen, mientras que un logotipo tan solo posee componentes de baja frecuencia. Como se verá más adelante, este método tan sólo funciona para sellos con grandes regiones homogéneas, mientras que falla para aquellos con trazos más delgados.

Antes de desarrollar ningún método, necesitamos alguna forma de medir cómo de efectivo resulta. Para medir la calidad de los resultados obtenidos con cada experimento se ha construido un *ground truth(termino).*

Las tecnologías que se han utilizado son:

* Python y en menor medida C++ para codificar los algoritmos.
* OpenCV, librería de visión artificial por excelencia.
* Pygame junto con Tkinter para desarrollar una aplicación que nos permita agilizar el proceso de crear un ground truth.
* MySQL para todas las necesidades de base de datos del proyecto, tanto almacenar resultados, como la información del ground truth o la ruta de las imágenes de cada uno de los sellos.

# Construcción de un ground truth

Saber qué conoces y qué no conoces. Ese es el verdadero conocimiento.

- Confucio -

En *machine learning (término)* se denomina ground truth a la colección de datos supervisados que se realiza con el objetivo de someter a prueba una hipótesis. Este ground truth consiste en una base de datos que relaciona cada documento con el sello que posee, si poseyera alguno. En tal caso, se almacenan las coordenadas de dicho sello también.

Para llevar a cabo este procedimiento se ha desarrollado un software que permite agilizar la tarea. En él puede visualizarse el documento en cuestión, acercar o alejar el zoom además de navegar por la imagen. Una vez localizado el sello, se seleccionan dos esquinas opuestas de este, eligiendo cada esquina con el click de cada uno de los dos botones del ratón. Esta interfaz se ha creado utilizando la librería Pygame, que permite dibujar gráficos simples en una ventana mediante un script de Python.

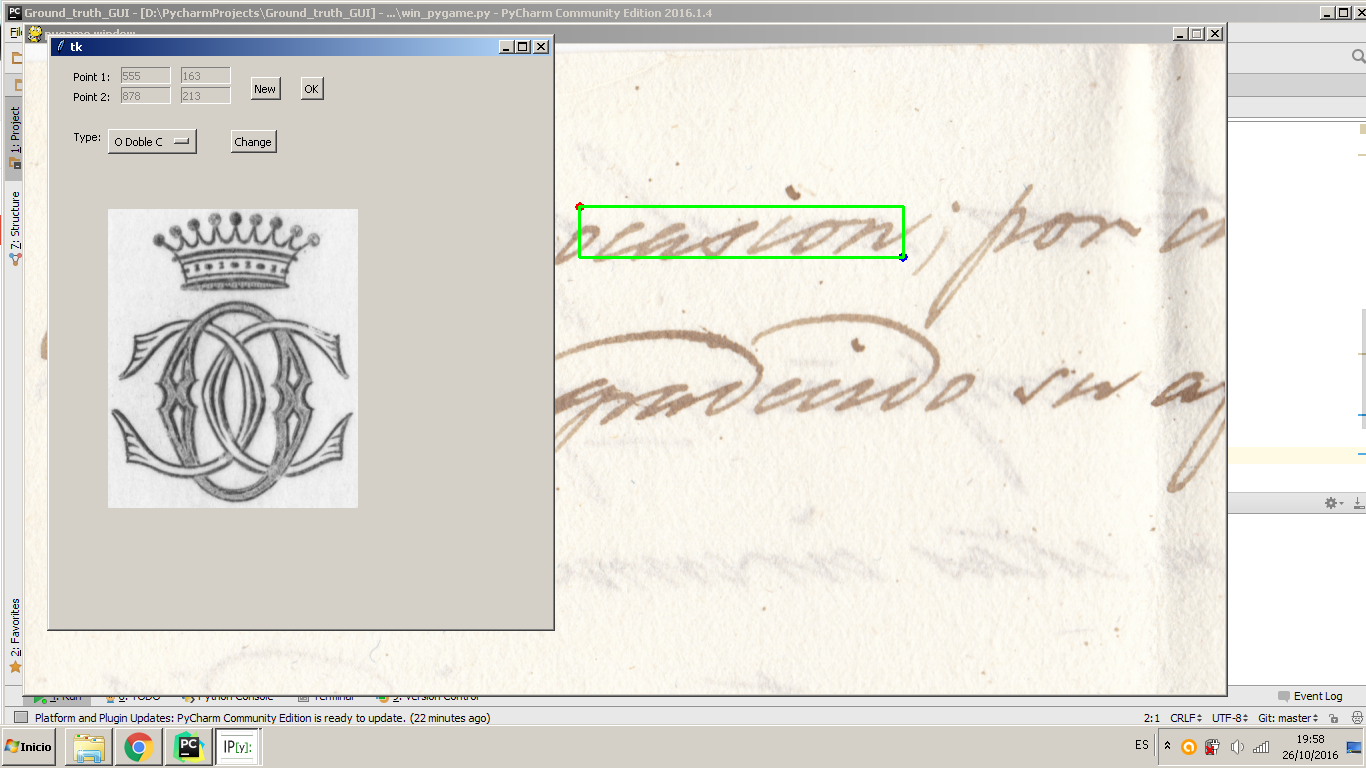


Figura X. Software para la elaboración de un ground truth

Una vez que el sello ha sido encuadrado, se utiliza el panel de control que se muestra en la imagen para seleccionar cuál es el tipo de sello que se ha encuadrado. Para esto, el panel de control se comunica con una tabla de una base de datos MySQL donde se almacenan tanto la ruta de la imagen como un nombre arbitrario que se ha escogido para describir cada sello. El usuario puede seleccionar el sello deseado de la lista desplegable y pulsar “Change” para que cambie la vista en el panel de control a una imagen del sello que se ha seleccionado y así asegurarse de que ha elegido el nombre de sello correcto. Si no se encontrase en la base de datos ningún sello del tipo que aparece en el documento, se pulsa el botón “New” y se copia en la carpeta de imágenes de sello un recorte de la imagen usando el rectángulo que se haya seleccionado. Además, se abre una ventana emergente donde se le da un nombre al sello y se le añaden de manera opcional metadatos como el autor al que corresponde el sello o una fecha aproximada a la que pertenece. Este nuevo sello se añade al menú desplegable de sellos.

Si el usuario pulsase “Ok”, se añadiría una nueva fila a otra tabla de la base de datos. Esta nueva tabla contiene la ruta a cada imagen de cada documento junto con el nombre del sello que contiene (o “no\_seal” en caso de que no tenga ninguno), además de las coordenadas del rectángulo seleccionado. Al hacer esto, el programa abre automáticamente el siguiente documento salvo que ya no exista ninguno más. Un detalle importante es que el programa almacena en un archivo de texto cuántos documentos se han clasificado ya. De este modo, si se interrumpe el proceso sin haber terminado no es necesario comenzar de nuevo.

El software no posee el diseño más atractivo, pero cumple con toda la funcionalidad que se precisa para su tarea. El resultado final es una tabla en la base de datos con los 155 documentos clasificados más otra que registra cada uno de los distintos tipos de sellos que aparecen en los documentos.

El código del programa desarrollado puede consultarse en el apéndice A.

# Esquema general de un ocr

Conoce bien las reglas, para que puedas romperlas de manera efectiva.

- Dalai Lama XIV -

E

n este capítulo se explicará cuáles son los pasos que suelen incorporarse en un algoritmo destinado a realizar un OCR. Después se explicarán varias maneras mediante las cuales se ha intentado abordar cada uno de esos pasos en este trabajo y por qué se han descartado o escogido cada uno de esos métodos.

El proceso de reconocer el contenido de un documento escrito de manera automática se denomina *reconocimiento óptico de caracteres* u *OCR* (término). De manera general todos los métodos de OCR tienden a seguir un esquema, aunque con frecuencia suelen omitirse alguno o varios de sus pasos en función del problema concreto y la solución que se ofrezca.

ESQUEMA OCR

## Binarización

En un primer paso se busca obtener a partir de una imagen en color una matriz binaria. Este paso no busca eliminar texto aún, sino simplemente una matriz de dimensiones iguales a las que tendría la imagen en escala de grises con valor falso booleano en lo que corresponda al papel y verdadero en donde hubiese algún elemento más oscuro, ya sea texto, sello o cualquier otro elemento gráfico del documento. Sí que se encarga este paso del proceso de filtrar aquellos elementos indeseables que erróneamente se cuelen como información escrita del documento, como tinta que se transparente del otro lado del papel, suciedad, marcas de envejecimiento o pliegues…etc. Primero se discutirán los métodos planteados en este trabajo para la binarización propiamente dicha. A continuación, el filtrado que eliminaría ese ruido.

### Elección manual de umbral

Consiste en elegir un umbral de intensidad de gris a partir del cual se considera si un píxel corresponde a tinta o papel. Este umbral se elegirá de manera experimental probando hasta encontrar un número que parezca acertado. Para encontrarlo, se crea un script que permite modificar este umbral además de visualizar los resultados en tiempo real. Para mejorar aún más el método, se aplica un filtrado gaussiano con el objetivo de eliminar pequeños ruidos puntuales. Los parámetros de este filtro también son /modificables en la herramienta. Tras múltiples ensayos puede verse que un umbral fijo no es un parámetro suficientemente robusto para todos los documentos.

AÑADIR REFERENCIA AL SCRIPT EN LOS ANEXOS

IMAGEN DE MISMO UMBRAL FUNCIONANDO BIEN Y MAL

### Método de Otsu

Este método recibe nombre en honor Noboyuki Otsu, quien creó el método en 1979 (citar paper de Otsu). Su objetivo es automatizar la tarea de encontrar el valor umbral para imágenes en escalas de grises de manera que todo píxel con una intensidad superior a dicho valor se considera objeto de interés, y todo aquel valor inferior, entorno (o, en nuestro caso, justo al contrario). Para hallar este valor umbral el método se vale del histograma de la imagen para hallar la varianza en las frecuencias de las intensidades que pertenecen a cada uno de los dos segmentos resultantes de cada umbral seleccionado. Aquel umbral que minimice esta varianza a la vez que maximice la varianza entre frecuencias de segmentos distintos será el umbral óptimo.

**Descripción matemática**

Partamos de dos segmentos de puntos ***K0(t)*** y ***K1(t)*** obtenidos a partir del umbral **t**. Sea ***p(g)*** la probabilidad de ocurrencia de del valor de gris 0 < g < G (con G=255 en nuestra imagen gris de 8 bits).

DIBUJO CON DEFINICIONES Y ACLARAR QUE PARA QUE SEA PROBABILIDAD HAY QUE DIVIDIR POR NÚMERO TOTAL DE PÍXELES

Entonces la probabilidad de que cada píxel pertenezca a uno u otro segmento sería:

Por tanto, si definimos como la media aritmética entre las intensidades de la imagen completa y y como las medias de intensidades dentro de cada segmento, entonces las varianzas entre intensidades dentro de cada segmento pueden calcularse como:

El objetivo es minimizar la varianza dentro de cada segmento y a su vez maximizar la varianza entre distintos segmentos. Para ello creamos una variable que englobe el cociente entre ambas medidas y se optimiza dicha variable:

Donde es la varianza entre ambos segmentos y se define:

,

y , la varianza dentro de los segmentos se define por:

La implementación más simple prueba todos los valores de ***t*** y escoge aquel para el cual ***Q(t)*** es máximo. Sin embargo esta implementación es muy lenta. Además es evidente que su tiempo de ejecución escala exponencialmente con la profundidad de bits de la imagen gris. Las implementaciones más utilizadas subdividen el rango de posibles valores de ***t*** de manera que es posible hallar el óptimo con tan solo unos cuantos intentos.

**Resultados obtenidos**

A menudo suele destacarse el tiempo de ejecución como desventaja de este método pero en la práctica no se han notado diferencias destacables entre usarlo o elegir un umbral de manera manual. En cuanto a la calidad de la imagen binarizada obtenida, los resultados son bastante buenos en la mayoría de los casos. Como cabe esperar, en los casos en los que apenas hay nada escrito en el papel y este presenta un evidente envejecimiento, el método cataloga este ruido como falso positivo. Es por esto que, aunque este método aporta generalmente resultados bastante buenos, no es suficiente en sí mismo para segmentar tinta y papel cuando el ratio cantidad de tinta frente a cantidad de ruido es demasiado bajo.

MOSTRAR CASOS DONDE FALLA-> SACARLOS DE RESULTADOS DE MÉTODO HEURÍSTICO.

### Método adaptativo

Este método pretende aplicar un umbral distinto a cada píxel de la imagen en función de la intensidad que tengan sus píxeles vecinos. De este modo, en imágenes iluminadas de manera heterogénea, el umbral es capaz de adaptarse a estos cambios de iluminación sin que las zonas muy oscuras contaminen a las claras con un umbral demasiado restrictivo y viceversa.

Para nosotros esto puede ser interesante ya que los defectos localizados en el papel son un problema frecuente. En nuestras pruebas el umbral se escoge igual a la media ponderada con un kernel gaussiano de los píxeles vecinos al píxel a umbralizar en cuestión.

IMAGEN RESULTADOS UMBRAL ADAPTATIVO

Una consecuencia evidente es que el método devuelve demasiados falsos positivos en regiones donde no hay tinta, tan sólo papel ya que el umbral adaptativo en dichas regiones es demasiado generoso al no encontrar nada suficientemente oscuro.

## Segmentación

Este paso tiene la finalidad de separar los distintos elementos gráficos del documento como pueden ser firmas, destinatarios, fechas, el contenido principal de la carta, etc. Para entender la importancia de este paso podemos imaginar qué ocurriría si se intentase reconocer caracteres en un logotipo o clasificar un sello a partir de una palabra. Es evidente que cualquier información que el sistema devolviese en estos casos no sólo sería errónea sino que podría empeorar los resultados arrojados a partir de otras porciones vecinas del documento. Por ejemplo, es frecuente en los sistemas más avanzados realimentar una detección de texto de manera que se corrija la estimación de una palabra en función de cuáles sean las palabras cercanas que se detecten. En nuestro caso, debemos separar en la mayoría de los casos tan solo sello de texto. Para resolver esta parte del proceso se han probado varios métodos.

### Segmentación mediante enfoque frecuencial

En (cita) se desarrolla un método para separar texto de elementos gráficos en documentos escaneados. La idea se basa en el hecho de que el texto introduce componentes armónicas de alta frecuencia a la imagen. Por lo tanto, un filtro de paso bajo eliminaría ese texto. En el algoritmo, se utiliza un filtro de Gauss para eliminar las componentes de alta frecuencia. El resultado es una imagen con los elementos de texto eliminados casi en su totalidad mientras que los elementos gráficos (en el artículo incluye también logos) permanecerían en la imagen, idealmente, en su totalidad. El algoritmo expuesto por Nandedkar, Mukhopadhyay y Sural es el siguiente:

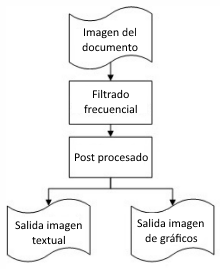


Figura X. Algoritmo espectral de seccionamiento

El algoritmo original presenta algunos pasos adicionales para separar distintos tipos de elementos gráficos, pero para nuestro problema, son pasos innecesarios.

**Filtrado frecuencial**

El filtrado frecuencial es el paso más importante del método, y es donde se elimina la mayor parte del texto. Es una evidencia fácilmente comprobable que el texto contribuye a componentes espectrales de muy alta frecuencia en imágenes de documentos (cita). Por ello, se convoluciona con un kernel gausiano como filtrado de paso bajo y se utiliza un post procesado para eliminar restos residuales de texto. El algoritmo puede verse en la figura X.

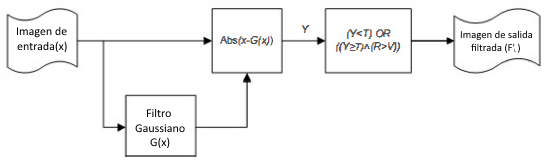


Figura X. Filtrado frecuencial

El proceso consiste en:

1. Se toma la imagen original convertida a escala de grises, X.
2. Se le aplica un filtrado gaussiano, G(X).
3. Se calcula la imagen la imagen de **paso alto** Y, restándole G(X) a la original.
4. Un filtrado selectivo en frecuencias sencillo sería entonces

Donde I es la imagen original de dimensiones NxM, 1≤ i ≤ N y 1≤ j ≤ M. T es el umbral de frecuencias. Los valores que superan un cierto valor en la imagen filtrada Y se consideran de alta frecuencia, los que no, de baja frecuencia. Por último, Bk es el valor del color considerado de fondo. En el artículo utilizan una media de los colores de fondo del papel, obtenidos sumando los píxeles de la imagen original I, que han quedado como fondo en una binarización de la imagen en escala de grises X. En este trabajo se ha considerado innecesario ya que no se pretende reconstruir los huecos dejados por los elementos de eliminados del documento, y se ha tomado Bk como blanco directamente.

Hay que destacar que un filtrado solamente en frecuencias no es apto para elementos cromáticos cuyos tonos cambien de manera abrupta, ya que también introducen componentes de alta frecuencia y dicho filtro los eliminaría. Para solventar este problema, se conservan también aquellos elementos que tengan una cantidad de color suficientemente alta, esto se consigue cuantificar mediante la *cromaticidad*.

**Definición** cromaticidad: Se entiende por cromaticidad de un tono de color a la medida de cómo de alejado se encuentre ese color de tonos grises, independientemente de su luminancia, es decir, de cómo de claro u oscuro sea.

En un espacio de colores YCrCb, la cromaticidad R se cuantifica como

Por tanto, dada nuestra imagen original, se define la matriz de cromaticidad aquella que contiene para cada píxel el valor de su cromaticidad.

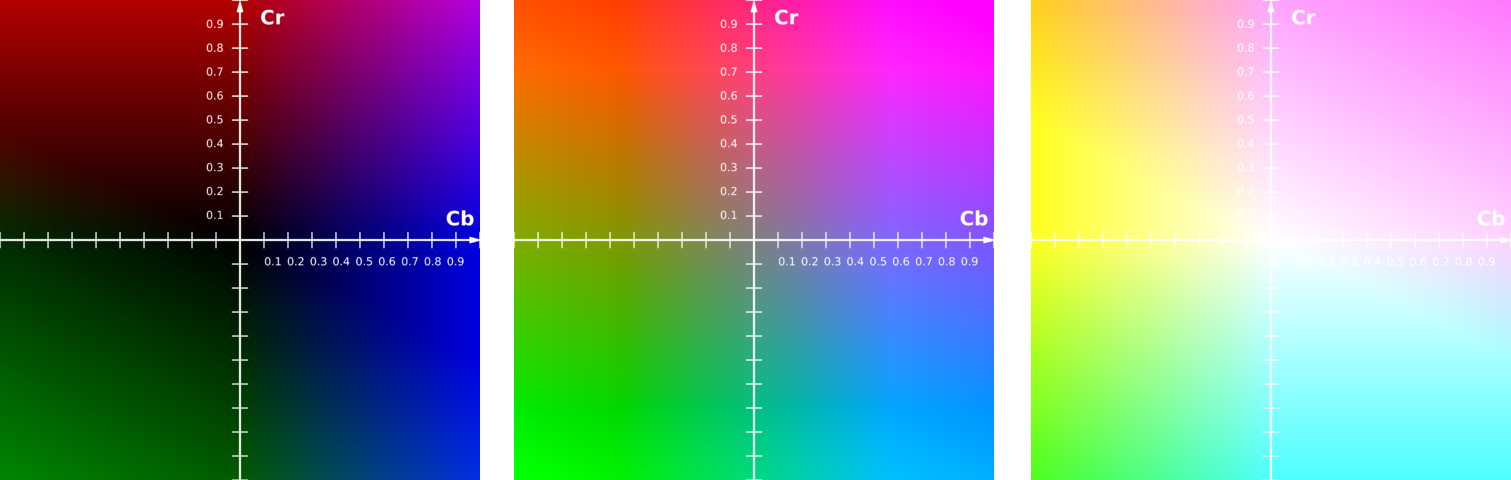


Figura x. Espacios de color YCrCb para valores de luma (Y) de 0, 0.5 y 1 respectivamente.

En la figura anterior puede verse como aquellos colores con una menor R, es decir, aquellos más cercanos al origen de coordenadas, poseen tonos más grises.

Volviendo al problema anterior, queremos evitar que al filtrar componentes de alta frecuencia eliminemos elementos gráficos que posean cambios bruscos en los colores. Para ello, imponemos un umbral V. Aquellos píxeles cuya cromaticidad sea superior a dicho umbral serán conservados en la imagen filtrada. Esto no supone ningún problema para la eliminación de texto ya que los caracteres tienen generalmente una cromaticidad muy baja.

Los resultados obtenidos mediante este procedimiento son los siguientes.



Figura X. Ejemplo de correcto funcionamiento del filtrado frecuencial.



Figura X. Ejemplo en el cual el filtrado frecuencial falla.

Como puede verse, los resultados en el primer caso son verdaderamente alentadores. Sin embargo, en el segundo caso el filtrado frecuencial elimina buena parte del sello además del texto. Esto se debe a que el sello está compuesto de trazos finos y una zona más gruesa pero que en realidad son franjas estrechas con diferencias tonales entre colores de muy baja cromaticidad. Este método es muy bueno cuando los gráficos a separar del texto poseen regiones amplias de colores uniformes, como logotipos, y que, si poseen algún trazo estrecho, éste posee una alta cromaticidad o no supone una porción importante del elemento en cuestión. Para nuestro caso por tanto, no supone una solución robusta.

Por todo eso, se procederá a probar otro método.

### Extracción y descripción de características

Para describir en qué consiste este enfoque, trate el lector de localizar la posición de cada uno de los fragmentos en la imagen inferior.

IMAGEN DE EDIFICIOS DE MANHATTAN

De manera inconsciente habrá tratado de buscar los elementos que pudieran ser más fácilmente reconocibles en cada uno de los fragmentos, como bordes o esquinas, y luego intentar localizar esos elementos en la imagen completa. Esto mismo es lo que se pretende mediante la extracción y descripción de características.

Característica: Partes de la imagen que poseen alguna propiedad que las hacen muy distintas al resto de imagen que las rodea, típicamente alguna medida relacionada con el gradiente.

Realmente no existe una definición global que pueda ser concreta sobre qué es una característica en visión artificial, ya que depende del algoritmo en concreto del que se hable. Pueden ser bordes, esquinas, blobs (o regiones de interés) o lo que se conoce como *ridges,* en español crestas. Estas últimas son regiones alargadas, por ejemplo, una imagen de un avión visto desde la planta está compuesta por dos *ridges* que se cruzan entre sí.

Existen multitud de algoritmos para esta tarea, pero en esta sección nos limitaremos a emplear uno de ellos: SURF. Nuestro objetivo de momento es simplemente ver si merece la pena invertir en este enfoque o por el contrario sería mejor descartarlo y buscar una nueva solución a nuestro problema.

Para nuestro experimento vamos a tomar dos imágenes de dos sellos extraídas de la base de datos de documentos. Después se comparará un documento con ambos sellos. Uno de ellos es el mismo que aparece en el documento, el otro no.

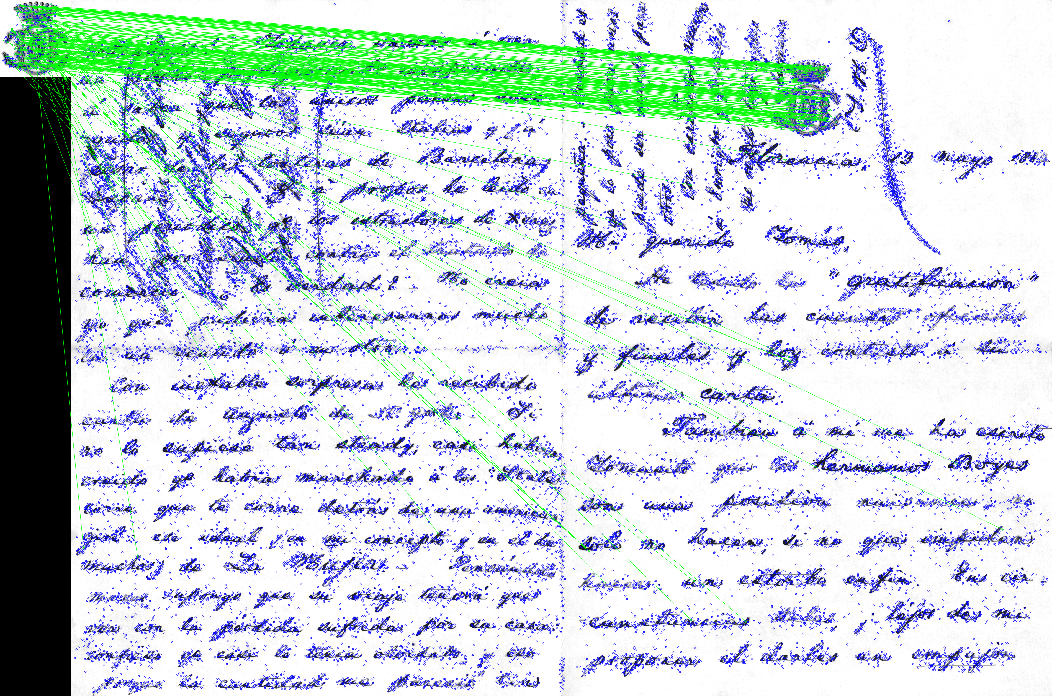


Figura X. Comparación de características entre el documento y su mismo sello.

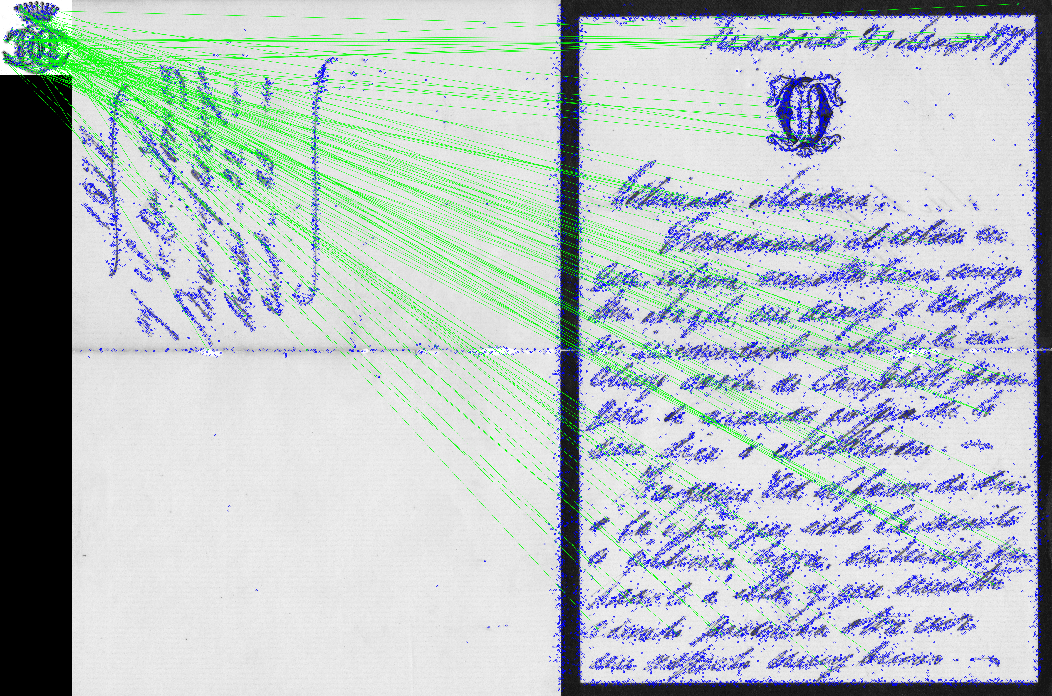


Figura X. Comparación de características entre el documento y un sello distinto.

El sistema calcula primero qué puntos de la imagen son mucho más distintos del resto (esquinas en la nomenclatura de visión artificial) y se han dibujado en azul. Después compara los puntos de la imagen del sello con los del documento y empareja aquellos que determina que más se parecen. Estos emparejamientos se han dibujado en verde. Más adelante se explicará en detalle todo el proceso.

Este primer experimento parece indicar que este sistema es bastante bueno para resolver el problema. Hay dos consideraciones importantes que hacer antes de continuar.

1. Este sistema no solo puede ser útil para segmentar estimando la posición del sello, sino que además se puede emplear en la fase de reconocimiento, para saber qué sello es.
2. Aunque los resultados parecen muy positivos no hay que olvidar que para el correcto funcionamiento de este sistema es necesario estar en posesión de una imagen de cada sello. Este no es un requisito absolutamente utópico, pero preferiría salvarse si fuera posible. Esto se debe a que en una aplicación real no es posible disponer de esta información ya que, si así fuera, significaría que la totalidad de documentos han sido clasificados y por lo tanto este proceso carecería de utilidad.

### Segmentación mediante componentes conexas

En visión artificial, la búsqueda de componentes conexas se refiere a la familia de algoritmos que pretende localizar grupos de píxeles que cumplan una determinada condición y que se encuentren en contacto directo. La idea es agrupar los píxeles oscuros que aparezcan conexos en la imagen e identificar qué propiedades poseen los que pertenecen a un sello con respecto a los que no.

OpenCV ofrece un detector de blobs que implementa componentes conexas utilizando como condición múltiples binarizados a distinto umbral y agrupando aquellos píxeles conexos en cada umbral. Este sistema es muy útil cuando existen objetos muy distintos en forma y/o color y que poseen mucho contraste con respecto del fondo. En nuestro caso, los objetos que queremos detectar son sellos y palabras. No están claramente diferenciados y, en muchos casos, el contraste con el fondo es escaso. Por todo esto el bajo control que ofrece este algoritmo no es suficiente.

Por ello se ha empleado el detector de componentes conexas sencillo sin funcionalidad adicional. Este algoritmo toma una imagen binarizada y devuelve una imagen de etiquetas. Esto consiste en que cada píxel se sustituye por una etiqueta de manera que aquellos píxeles que se encuentren conexos posean todos la misma etiqueta única para cada grupo. Una etiqueta no es más que un valor entero incremental a partir de 0, que se reserva para todos los píxeles del fondo.

Una vez que sabemos qué píxeles pertenecen al mismo objeto de la imagen binaria se pueden hacer todo tipo de operaciones. Cabe destacar que, una vez más, aún no se pretende desarrollar un sistema perfectamente funcional y capaz de resolver el problema, sino un prototipo rápido del algoritmo que nos permita juzgar si merece la pena seguir desarrollándolo o si en cambio debe ser descartado y sustituido. En esta primera prueba se ha calculado el área de cada región conexa como el número de píxeles que se incluyen dentro de ésta, y se ha descartado como posible candidato a sello aquellas regiones demasiado pequeñas. Además, es frecuente que palabras de diferentes líneas en las que aparecen caracteres con pronunciada verticalidad como podrían ser una “p” y una “t” se unifiquen en una misma región al aparecer solapados dichos caracteres.

IMAGEN RATIO DE RELLENO

Para eliminar este común error, se ha calculado el *bounding box* o cuadro delimitador de cada región y se ha calculado el área de éste como el área geométrica del rectángulo. A continuación se ha obtenido el ratio del área de la región (es decir, el conteo de puntos que forman parte de esta región) con respecto al área de su cuadro. A este ratio se le ha denominado **ratio de relleno**. Si el ratio de relleno es demasiado pequeño, la región es descartada también.

Los resultados son claros: este proceso parece efectivo para sellos claramente separados del texto y con una morfología claramente diferenciable. A continuación se muestra un ejemplo de una correcta identificación de un sello seguido de otro en el cual el sistema no era capaz de localizarlo.

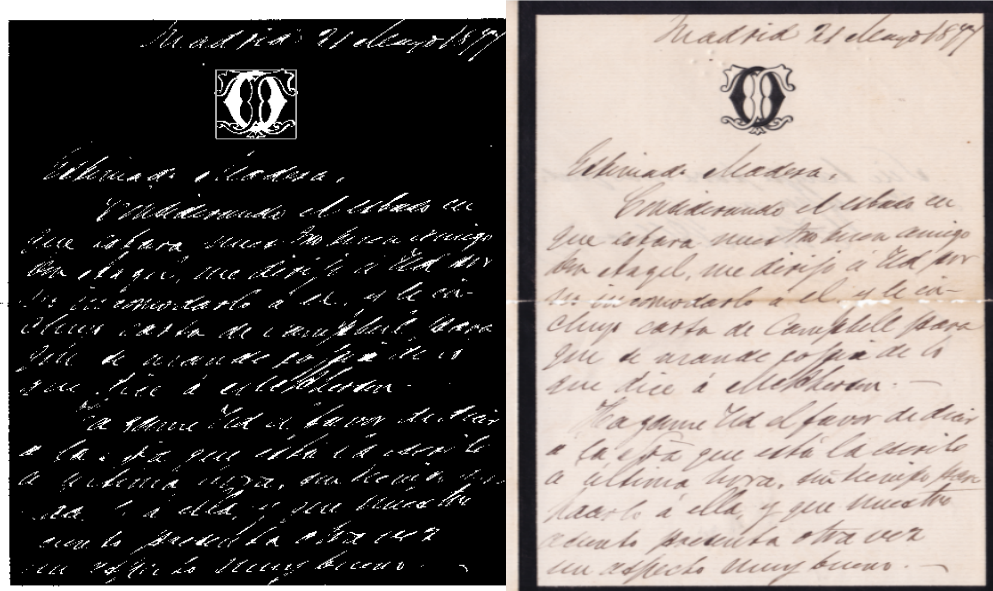


Figura X. Ejemplo del correcto funcionamiento del algoritmo.



Figura X. Ejemplo de sello incapaz de ser detectado por el algoritmo actual

### Conclusiones sobre segmentación

Por un lado disponemos de detección de características, un sistema que parece funcionar muy bien pero que requiere una imagen previa de cada sello. Por otro lado tenemos componentes conexas, que no es robusto para la totalidad de los casos. La solución propuesta final consiste en utilizar componentes conexas para extraer una muestra de cada sello y a continuación emplear detección de características a partir de la base de datos de sellos obtenida para localizar la posición del sello a la vez que clasificarlo. El algoritmo se divide de esta manera en dos fases:

1. Fase heurística, en la que se emplea componentes conexas y una serie de filtros heurísticos con el objetivo de .conseguir al menos una muestra de cada uno de los sellos
2. Fase de extracción de características, donde se emplea los prototipos de sellos hallados para localizar y reconocer los sellos, si los hubiera, de cada uno de los documentos.

### Mención de honor a segmentación mediante transformada de Hough

Con inspiración en la literatura especializada (citar SEAL DETECTION AND RECOGNITION) se experimentó con un método basado en la transformada de Hough. Los sellos no presentan, en un caso general ninguna curva de parametrización sencilla. Aun así, suelen estar presentes elementos similares a circunferencias o elipses. Se prueba a buscar estas formas geométricas en los documentos y se obtienen los resultados siguientes.

IMAGEN RESULTADOS TRANSFORMADA HOUGH

Como cabía esperar, la transformada de Hough no supone un método viable para abordar el problema que se presenta.

## Pasos restantes de un OCR estándar. Separación de líneas y palabras y sistema de reconocimiento

En el caso del problema expuesto, no estamos tratando de transcribir un texto. Es por ello que, supuesta resuelta la segmentación, carece de sentido separar líneas y menos palabras, ya que estos elementos solo existen en fragmentos de texto y no en un sello. En cuanto al sistema de reconocimiento, en el caso dado debe clasificar de qué sello en cuestión se trata. Como ya se ha mencionado el sistema de extracción de características realizará esta función a demás de segmentar.

Como se aclaró al exponer el esquema estándar de un sistema de OCR, rara vez se implementa el modelo tal cual. Es bastante frecuente que se omita alguno de los pasos o incluso que se agreguen pasos adicionales. En general en el campo de la visión artificial es poco frecuente encontrar soluciones multipropósito para un problema y por lo tanto, tampoco es común que todas las soluciones se adapten a un esquema rígido.

# Fase heurística

Saber qué saber y qué no sabes, eso es el conocimiento verdadero.

- Confucio -

S

e recomienda consultar la norma UNE 50136:1997: Documentación, presentación de tesis y documentos similares. Por otro lado, existen recursos en línea que ayudan a organizar todo el proceso de elaboración del Trabajo. El **personal de la Biblioteca** imparte formación y asesoramiento sobre su uso.

Para establecer la estructura y orden de los datos y partes del Trabajo Fin de Grado nos basamos en la ***Normativa de los Trabajos Fin de Grado*** de la Escuela Técnica Superior de Ingeniería de Sevilla (Aprobada en la sesión de la Junta de Escuela de 12 de julio de 2013, modificada en Junta de Escuela de 05 de febrero de 2014) y en la norma *UNE 50136:1997: Documentación, presentación de tesis y documentos similares.*

## Preliminares

### Cubierta

Deberá utilizarse **la plantilla** aprobada por la Junta de Escuela del 25/4/2014 que se describe en este texto y cuyo modelo puede tomarse de este documento. Debe contener todos los datos que se especifican y en el orden y modelo que se establece: nombre completo del autor, título del Trabajo, nombre completo del tutor, nombre del Departamento, fecha, Grado al que se opta, etc.

### Portada

Igulamente deberá utilizarse **la plantilla**… y contener todos los datos que figuran en ésta, en el orden especificado: el nombre de la Escuela, la titulación y, en su caso, la intensificación, el título del TFG, los nombres del autor, del tutor(es) y, en su caso, del ponente, el Área de Conocimiento, el Departamento y el año de ejecución del proyecto.

### Resumen

Texto que muestra de forma abreviada y precisa el contenido del trabajo. **Se podrá también incorporar una versión en inglés que se colocará a continuación precedida del término Abstract.**

### Prefacio o Introducción

Debe incluir una breve explicación de las razones que han llevado a la realización del Trabajo, el propósito y los objetivos que se pretenden, el ámbito, alcance y límites de la investigación, así como la metodología empleada y, si se considera oportuno, un avance de las conclusiones alcanzadas.

### Índice y/o Índice general

Tabla de contenidos donde se reflejan todas las partes del Trabajo y sus anexos, si los hubiera. Deben aparecer los títulos, en su orden y con indicación de la página en la que se pueden encontrar.

Si el Trabajo consta de varios volúmenes, cada uno deberá llevar su propio índice, pero se debe incluir también un índice general.

### Lista de ilustraciones y tablas

Si el Trabajo incluye ilustraciones y tablas se puede añadir un listado que incluya el número identificativo que figura dentro del texto, la leyenda y el número de la página en la que se encuentra.

También es conveniente mencionar los datos sobre las fuentes de donde se han obtenido dichas ilustraciones si no se han incluido en el propio texto de la memoria.

### Lista de abreviaturas y símbolos

El Trabajo debe contener las abreviaturas y símbolos internacionalmente reconocidos. Si se incorporan unidades, abreviaturas o acrónimos que puedan ser poco conocidos se deberán explicar brevemente en estas listas.

### Glosario

Los términos que requieran definición o explicación se deberán incorporar en un glosario.

## Texto principal

Para su elaboración es importante tener en cuenta su distribución en capítulos y secciones numeradas, siguiendo la plantilla elaborada por….

El texto debe comenzar con una introducción que muestre las investigaciones previas existentes sobre el tema y destacar los objetivos y métodos seguidos para llevar a cabo la investigación o análisis del tema tratado.

Para finalizar se deben escribir las conclusiones que deben estar en relación con los objetivos marcados previamente.

## Bibliografía

Debe contener las referencias bibliográficas de los documentos consultados para demostrar las bases del trabajo realizado y avalar los datos incorporados y citados en el texto.

Se elaborará de forma normalizada, para lo que se aconseja utilizar la norma UNE vigente (actualmente la “UNE ISO 690:2013. Información y documentación. Directrices para la redacción de referencias bibliográficas y de citas de recursos de información”).

Para la elaboración de esta parte del Trabajo se recomienda consultar la Web de la Biblioteca de Ingeniería que contiene recursos, guías y ayudas para la elaboración de las referencias bibliográficas.

## Anexos

Se puede incluir de esta forma material extenso utilizado en el trabajo, importante para justificar los resultados y las conclusiones obtenidas, pero que no es esencial para la comprensión del texto principal. Pueden ser datos estadísticos, legislación, etc.

**La paginación debe ser correlativa y continuar la del texto principal. Cada uno de los anexos debe identificarse con una letra mayúscula del alfabeto, comenzando por la letra A, precedida de la palabra Anexo.**

# Uso de Estilos

Esto es una cita al principio de un capítulo.

- El autor de la cita -

L

os capítulos nuevos pueden comenzarse de una forma cómoda copiando y pegando esta página (desde el principio hasta el final) en una página en blanco. Hay cuatro “profundidades” de Títulos que representan los capítulos, las secciones, las subsecciones y los apartados. Para asegurarse de que el capítulo comienza en página impar puede introducir un salto de sección impar.

## Secciones

Texto de sección. Esto está escrito con el estilo Normal *(Inicio > Estilos)*.

### Subsección

Texto de subsección.

#### Apartado

Texto del apartado dentro de la subsección.

Una cita a otra obra se inserta desde el menu *Referencias > Insertar Cita > Agregar Nueva Fuente*. Hay que rellenar los campos que nos interesen. Existen muchos Estilos para poner la cita. Por ejemplo, con el estilo IEEE 2006 quedan así (Autor, 2012), pero se pueden probar otros hasta ver el que más nos convence (Autor O. , 2001).

Una vez tengamos insertada esa cita, ya podemos encontrarla en el menú *Referencias > Insertar Cita* sin tener que agregar una nueva fuente.

### Otra subsección

Texto de la otra subsección. Aprovecharemos para insertar una nota[[4]](#footnote-5) al pie.

## Otra sección

Cada vez que escribamos uno de estos títulos y pulsemos Enter, volverá al estilo Normal.

Gracias a los Elementos Rápidos, podemos insertar *Teoremas, Ejemplos, Demostraciones, Definiciones, Imágenes y Tablas.*

|  |
| --- |
|  |
| Figura 3‑1. Esto es el pie de la figura. |

Para insertar ecuaciones, deberemos tener instalado MathType y que se integre correctamente en Word (Aparecerá una pestaña en el menú superior). Se puede usar el editor de ecuaciones integrado en Word, pero aunque tiene alguna ventaja no se recomienda. Aquí, como puede ser que el lector no tenga MathType, se ha optado por usar el editor de ecuaciones de Word.

Dentro de la pestaña MathType, se puede insertar una ecuación inline como esta, que no modifica la separación entre líneas gracias a una propiedad del estilo Normal. Hay que tener cuidado con esto, pues ecuaciones inline con mucha altura pueden superponerse con el texto. En el raro caso de que ocurra, lo mejor es reducir el tamaño de la ecuación o poner una ecuación a parte.

Las ecuaciones a parte se ponen pulsando sobre *MathType > grupo “Insert Equations” > Right-numbered*, y se numeran automáticamente según el número del capítulo. Otra opción, la que se adopta a continuación es crear una tabla con el número generado con un título “(“ incluyendo el número de capítulo, si no lo ve en sus títulos tendrá que crearlo,

|  |  |
| --- | --- |
|  | (3–1) |

Las ecuaciones numeradas tienen su propio estilo, MTDisplayEquation, que puede editarse para cambiar alineación y otros parámetros (aunque no es necesario).

Para referenciar una ecuación, usamos *MathType > grupo Equation Numbers > Insert Reference*, y luego hacemos doble click sobre el número de la ecuación que queremos, en este caso (3–1).

Vamos a ver algunos elementos rápidos, empezando con el más sencillo de ellos, la Definición.

**Definición** (Concepto):Texto de la definición. Esto es otro de los Elementos Rápidos, aunque no tenga numeración.

Las tablas también son Elementos Rápidos. Todos los elementos rápidos se acceden desde el menú de Insertar. No obstante, también podemos copiar de estas mismas líneas el elemento que nos interese, y pegarlo donde queramos editarlo.

Tabla 3–1. Tipos de transmisión y frecuencia central

|  |  |
| --- | --- |
| Tipo de Transmisión | Frecuencia central de transmisión |
| Modem | 100-1800 Hz |
| Radio AM | 530-1600 kHz |
| Radio FM | 88-108 MHz |
| Televisión | 178-216 MHz |
| Telefonía móvil | 850 MHz-1,8 GHz |
| Redes inalámbricas | 2*,*4 GHz |
| Fibra óptica | 2·1014 Hz |

De hecho, cualquiera puede crear un Elemento Rápido a su gusto desde el propio menu de Elementos Rápidos, pulsando en *“Guardar selección …”* tras haber seleccionado la parte del documento que queremos guardar.

Otro de los Elementos Rápidos es el Ejemplo.

**Ejemplo 3–1.** Al insertar ecuaciones en un ejemplo, puede haber problemas con las barras grises superior e inferior.

Los ejemplos que dan más problemas son los que llevan ecuaciones o los que continúan en la página siguiente. En ese caso hay que hacerle un retoque final al mismo.

**Ejemplo 3–2.** Al insertar ecuaciones en un ejemplo, puede haber problemas con las barras grises superior e inferior.

|  |  |
| --- | --- |
|  | (3–2) |

En cualquier caso, estas barras se configuran una vez está acabado el ejemplo en el menú Inicio > grupo “Párrafo” > pulsando sobre el icono inferior derecho de ese grupo. Por ejemplo, si nos aparecen barras grises delimitando la ecuación recién insertada, basta con situarse en el párrafo superior e inferior a la ecuación y eliminar esas barras en el menú que hemos dicho.

Si queremos referenciar cualquier elemento numerado a lo largo del texto (como por ejemplo “1 ” o “Tabla 3–1”, se hace desde el menú *Insertar > grupo “Vínculos” > Referencia cruzada*. Luego hay que buscarlo en la caterogría adecuada (*Tipo*) y especificar a qué queremos que haga referencia (*Referencia a*). Si quedan con un formato indeseado, siempre se puede marcar y aplicar el estilo Normal.

Debemos saber que en cualquier momento, si una numeración no nos aparece correctamente, puede ser debido a que Word todavía no la ha actualizado. Para forzar esta actualización, hacemos click sobre la última cifra de esa numeración y pulsamos *Actualizar Campos*.

|  |
| --- |
| **Teorema 3.1 (Nombre del teorema)**  Aunque algunas ecuaciones aparezcan cortadas contra el fondo gris, al imprimirlas en PDF salen bien. No obstante, al insertar una ecuación puede ser necesario incluir alguna línea en blanco adicional, si parece que van a solapar con el texto…  Y esta es la línea final del Teorema. |

**Demostración:** Las demostraciones siguen al teorema, tienen cierta sangria y acaban con un cuadrado negro antes de volver al estilo Normal. Las demostraciones demasiado engorrosas pueden ponerse en un apéndice al final del documento, pero eso ya es cuestión personal.

∎

Algunos conceptos son muy importantes para la Tesis y conviene añadirlos al glosario al menos la primera vez que aparecen. Para ello, los marcamos y pulsamos *Referencias > Marcar entrada.* Luego pulsaremos “Marcar” y “Cerrar”. Si Word nos ha pasado al modo “Mostrar Todo”, donde se muestran los símbolos de formato, podemos salir de él pulsando *Control + Shift + (.*

# Otro Capítulo

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

Claude Shannon, 1948

E

n este capítulo vamos a describir las partes de las que consta un documento tipo, cómo deben interpretarse los diferentes comandos que se han definido para su confección.

Los capítulos nuevos pueden comenzarse de una forma cómoda copiando y pegando esta página (desde el principio hasta el final) en una página en blanco. Hay cuatro “profundidades” de Títulos que representan los capítulos, las secciones, las subsecciones y los apartados. (título1 a título5)

Hay mucho de elección personal en lo que sigue y únicamente se justifica desde el gusto personal de quienes escribimos esto. No pretendemos por ello sentar precedentes, obligaciones ni restricciones a quien desee utilizar este documento. En cualquier caso, esperamos que su lectura sea provechosa para la confección y edición de libros, apuntes de clase, proyectos, etc.

## Estilos de un documento

Aquí definiremos los principales estilos que se han definido en este documento. Estos pueden accederse desde la pestaña *Inicio* y clicando en la sección *Estilos*, los verá en la parte media de la barra de herramientas. Pruebe a consultar todos los estilos que posee el documento pinchando en la flecha que apunta hacia abajo con una raya encima, debajo de los estilos que aparecen.

El dominio de los estilos es fundamental para una buena edición de un texto con Word. Se recomienda invertir un tiempo en “jugar” con las opciones en el menú Formato>Estilo… Una de las mayores ventajas del dominio de los estilos es asignar una numeración automática a cada capítulo, sección y apartados. De esta forma al introducir o eliminar estos elementos el editor renumera automáticamente todo el documento.

Los principales estilos que se han diseñado para este texto son los siguientes:

* *Autor\_cita\_celebre\_capitulo* -> Se aplica al autor de la cita con la que se suele iniciar un capítulo.
* *Cita\_celebre\_capitulo* -> El contenido de la cita con la que se inicia un capítulo.
* *Código\_cuerpo ->* El contenido (sin incluir el título) de un bloque de código.
* *Definición ->* El texto que forma parte de una definición.
* *Demostración ->* El texto que forma parte de una demostración.
* *Encabezado ->* El estilo con el que se muestran los encabezados de las distintas páginas.
* *Normal ->* El texto “estándar” del documento,
* *Otro\_titulo ->* Para aquellos títulos de primer nivel que no deben numerar, por ejemplo: Resumen, Agradecimientos, etc.
* *Otro\_titulo2 ->* Para aquellos títulos de segundo nivel que no se deben numerar, por ejemplo los títulos de los problemas.
* *Portada\_autores->* Para mostrar los nombres de los autores en la portada del documento.
* *Portada\_departamento ->* Para mostrar el nombre del departamento en la portada del documento.
* *Portada\_fecha ->* Para mostrar la fecha en la portada del documento.
* *Portada\_proyecto ->* Para mostrar el nombre del proyecto en la portada.
* *Subportada\_proyecto ->* Para mostrar los datos en la página interior que sigue a la portada.
* *Teorema ->* Para usar cuando se exprese un teorema.
* *Titulo\_previo ->* Muy similar a otro\_titulo.
* *Titulo\_codigo ->* Para los títulos en los bloques de código
* *Título 1 ->* Primer nivel de títulos (capítulos), tiene un subrayado en gris.
* *Título 2 ->* Segundo nivel de títulos (secciones).
* *Título 3 ->* Tercer nivel de títulos (subsecciones).
* *Título 4 ->* Cuarto nivel de título (apartados).
* *Título 5 ->* Usado para algunos encabezados menores.
* *Descripción ->* Se usa en los títulos de figuras, tablas, etc

## Títulos y Referencias Cruzadas

El otro elemento fundamental, además de los estilos, que el usuario debe dominar es el par Títulos/Referencia Cruzada. Si va al menú Insertar>Título verá que puede incluir un título de la lista. En el presente documento hay títulos definidos para ecuaciones del tipo (X-Y), por si no usa la numeración de Mathtype, para Figura X-Y, Tabla X-Y, Ejemplo X-Y y Teorema X-Y, donde X es el número del capítulo e Y es el número dentro del capítulo. Puede probar a crear uno, como por ejemplo Lema. Una vez que se ha introducido un título, para la leyenda de una Figura o Tabla, o para numerar una ecuación, se puede hacer referencia a él insertando Referencia Cruzada, en el mismo menú Insertar.

Este elemento es fundamental, de nuevo, para que al introducir o eliminar una nueva figura o ecuación el sistema pueda renumerar automáticamente (opción actualizar campos) todos los títulos y actualizar las referencias cruzadas. Para esto a veces es preciso seleccionar el texto y en el menú que aparece con el botón derecho del ratón pulsar Actualizar Campos. Además, esto permite insertar una lista de elementos al final o principio del documentos.

## Versiones y Sistemas Operativos

El programa Word® dispone de diversas versiones, este documento ha sido realizado usando la más reciente de ellas Office 2013, se ha comprobado que funciona correctamente, al menos, con las versiones 2010 y 2007 de la suite ofimática. Se proporciona un fichero con extensión .docx, que no es totalmente compatible con versiones anteriores. En todo caso el Centro de Cálculo y la Biblioteca de la Escuela cuentan en sus instalaciones con equipos con este software instalado y de uso público. Veamos ahora cómo trabajar con este documento en sistemas distintos de Windows.

### **Macintosh**

Para Macintosh está también disponible la suite Microsoft Office, por lo que no debe haber problemas al trabajar con este documento. Se ha probado en un Mac con la versión 2010 de Word® y funciona correctamente.

### Linux

Para Linux no existen versiones nativos de Office. Se nos ofrecen varias alternativas para poder usar este documento:

* Ejecutar un emulador para correr sobre él la suite de Office original, ejemplos son Wine y Crossover Office (específicamente diseñado para correr Office). La ventaja de esta opción es que trabajamos con el mismo sistema y el documento no sufre ninguna alteración y el consumo de recursos es moderado, la desventaja es que su configuración puede llegar a ser tediosa.
* Instalar una máquina virtual y sobre ella un sistema operativo Windows con su correspondiente copia de Office. Como desventaja más evidente comentar que no todos los equipos cuentan con potencia suficiente como para ejecutar esta opción.
* Usar una suite ofimática alternativa, como LibreOffice. El consumo de recursos será moderado y su uso es inmediato, por el contrario puede que el documento no se vea exactamente igual y que algunas operaciones deban ser realizadas de una forma distinta.

## Texto en inglés

El idioma por defecto de este documento es el Español, si usted quiere que algunas opciones como la autocorrección, el formato de moneda y alguna otras le aparezcan en Inglés o en algún otro idioma, simplemente empiece a escribir en ese idioma y el programa automáticamente lo detectará si el texto es lo suficientemente largo, es posible que le pida que instale el soporte extra para ese idioma si usted no lo tenía previamente instalado

## Elementos básicos de un libro

En este capítulo describimos los puntos que pueden incluirse con el formato propuesto. En primer lugar, la longitud de un libro, en general, justifica su separación en partes. Una posibilidad es que un libro esté dividido en Partes y esta a su vez en Capítulos. Y por último, a veces existen Apéndices que se incorporan cuando han acabado los capítulos. En nuestro caso sólo hemos considerado la posibilidad de dividir el libro en capítulos. Además, existen un conjunto de elementos como dedicatoria, prefacio, agradecimientos, cubierta, etc, que también son partes que se han tenido en cuenta.

En un nivel de descripción diferente, podríamos considerar que un libro se encuentra dividido en cubierta, página de título y trasera de la página de título, elementos antes del cuerpo del libro, tales como agradecimientos, prefacio, índices, etc, el cuerpo del libro en sí, dividido en capítulos y esto a su vez en secciones, subsecciones, subsubsecciones, subcapítulos, apéndices y, por último, la parte del libro después del cuerpo, que agruparía elementos tales como la lista de figuras del libro, la bibliografía, el índice, etc.

## Símbolos y fórmulas

Aunque Word® no es un editor científico ni la herramienta más utilizada para estos menesteres, hay que reconocer que el editor de fórmulas que trae integrado ha mejorado notablemente con las sucesivas versiones. Así mismo, actualmente se pueden insertar casi cualquier símbolo de un uso medianamente cotidiano. Para insertar un símbolo o una fórmula simple se debe ir a la pestaña *Insertar* y después pulsar en *Símbolo,* en la parte derecha de la barra de herramientas. Justo al lado nos encontramos con la opción de introducir un *Cuadro de texto*, el cual puede servirnos para resaltar ciertos tipos de contenido. En MacOS es muy común tener problemas con los símbolos, al estar duplicada la fuente “símbolo” y tener que desactivar una de ellas.

## Ecuaciones y MathType®

Como hemos comentados en el apartado previo el editor de ecuaciones integrado de Word® ha mejorado mucho versión tras versión, aún así si nuestro trabajo tiene un número importante de expresiones matemáticas no la mejor opción. Nos encontramos en el mercado con la herramienta Mathtype® que, tras instalarla, se integra perfectamente con Word®, apareciéndonos una nueva pestaña en la parte derecha donde tendremos acceso a todas las funciones de Mathtype®. Esta aplicación no es gratuita, si bien cuenta con una versión de evaluación plenamente funcional operativa durante 30 días ya que puede descargarse desde la url: [http://www.dessci.com/en/products/Mathtype®/](http://www.dessci.com/en/products/mathtype/).

Para insertar ecuaciones, deberemos tener instalado MathType y que se integre correctamente en Word (Aparecerá una pestaña en el menú superior[[5]](#footnote-6)). Se puede usar el editor de ecuaciones integrado en Word, pero aunque tiene alguna ventaja no se recomienda. Aquí, como puede ser que el lector no tenga MathType, se ha optado por usar el editor de ecuaciones de Word. Las ecuaciones s incluyen pulsando sobre *Mathtype® > grupo “Insert Equations” > Right-numbered*, y se numeran automáticamente según el número del capítulo. Las ecuaciones numeradas tienen su propio estilo, MTDisplayEquation, que puede editarse para cambiar alineación y otros parámetros (aunque no es necesario). Para referenciar una ecuación, usamos *Mathtype® > grupo Equation Numbers > Insert Reference*, y luego hacemos doble click sobre el número de la ecuación que queremos, en este caso **¡Error! No se encuentra el origen de la referencia.**.

Dentro de la pestaña MathType, se puede insertar una ecuación inline como esta, que no modifica la separación entre líneas gracias a una propiedad del estilo Normal. Hay que tener cuidado con esto, pues ecuaciones inline con mucha altura pueden superponerse con el texto. En el raro caso de que ocurra, lo mejor es reducir el tamaño de la ecuación o poner una ecuación a parte. También puede jugar con el interlineado, en el menú Formato>Párrafo.

Las ecuaciones a parte se ponen pulsando sobre *MathType > grupo “Insert Equations” > Right-numbered*, y se numeran automáticamente según el número del capítulo. Otra opción, la que se adopta a continuación es crear una tabla con el número generado con un título “(“ incluyendo el número de capítulo, si no lo ve en sus títulos tendrá que crearlo,

|  |  |
| --- | --- |
|  | (4–1) |

Las ecuaciones numeradas tienen su propio estilo, MTDisplayEquation, que puede editarse para cambiar alineación y otros parámetros (aunque no es necesario).

Para referenciar una ecuación, usamos *MathType > grupo Equation Numbers > Insert Reference*, y luego hacemos doble click sobre el número de la ecuación que queremos, en este caso (3–1).

Por otro lado, en el menú de MathType se permite definir un estilo de ecuación, con tipos de letras y tamaños de las mismas, para nuevas ecuaciones y para aplicarlo a todas las ecuaciones del texto. Esto último es muy útil a veces.

### Fuentes

La selección de las fuentes para la edición de cualquier texto no es una tarea trivial. En realidad, el diseño tipográfico es todo un arte. Un convenio bastante aceptado es utilizar fuentes con serif para el texto y sin serif para titulares y cabeceras de páginas. Sin embargo, la elección de cualquiera de estas familias de fuentes es prácticamente cuestión de gusto personal y, por qué no decirlo, de la moda del momento.

En los primeros tiempos de Word® las posibilidades de elección estaban bastante limitadas. Sin embargo, con el advenimiento de nuevos métodos y programas, es posible elegir prácticamente cualquier fuente existente para su uso, y asociarla a un estilo.

### Epígrafes o citas célebres

En muchos libros, después del título de un capítulo o antes del resumen, o en el lugar que apetezca, se coloca una frase con diversos significados. Para ello nosotros simplemente debemos escribir el texto que deseemos y después aplicarle el estilo *Cita\_celebre\_capitulo*, para indicar el autor de la cita debemos escribirlo y después aplicarle el estilo *Autor\_cita\_celebre.*

### Figuras y tablas

Una parte importante de cualquier texto son las figuras y tablas que lo acompañan. En Word® estos elementos se consideran elementos flotantes.

La inclusión de las figuras se realiza simplemente yendo a la pestaña *Insertar -> Imágenes* y seleccionando la imagen de la ubicación de nuestro equipo donde se encuentre. Si bien Word® incluye algunas herramientas para la generación de figuras y diagramas, nuestra recomendación es usar alguna herramienta externa, guardar el resultado como una imagen y después insertarla. Por ejemplo para la creación de diagramas podemos recomendar Visio (propietario), Dia (libre y gratuito) y Lucidchart (una opción online muy potente). Para la edición de imágenes cualquier programa de edición de las mismas nos servirá.

|  |
| --- |
|  |
| Figura 4‑1. Pie de figura |

Para insertar una tabla actuaremos de una forma similar en la pestaña *Insertar -> Tabla,* una vez hallamos hecho esto nos aparecerán a la derecha dos nuevas pestañas con la etiqueta “Herramientas de tabla” por encima. Estas dos nuevas pestañas son *Diseño* y *Presentación,*  donde podremos designar varias características visuales y de formato de las tablas. Nuestra elección se muestra en la tabla inferior, nuevamente solo hay que copiar y pegar para poder reusar este formato. Si quiere mantener una tabla sin que se rompa en dos hojas contínuas, tendrá que seleccionar todas las filas menos la última y en el menú Formato>Párrafo seleccionar la pestaña Líneas y Saltos de Páginas y allí pulsar la opción Conservar con el siguiente.

Tabla 4–1 Tipos de transmisión y frecuencia central

|  |  |
| --- | --- |
| Tipo de Transmisión | Frecuencia central de transmisión |
| Modem | 100-1800 Hz |
| Radio AM | 530-1600 kHz |
| Radio FM | 88-108 MHz |
| Televisión | 178-216 MHz |
| Telefonía móvil | 850 MHz-1,8 GHz |
| Redes inalámbricas | 2*,*4 GHz |
| Fibra óptica | 2·1014 Hz |

Observemos que en la parte inferior de las figuras y en la superior de las tablas (esta ha sido nuestra elección), se colocan textos explicativos sobre las mismas. Para introducir este título pulsamos con el botón derecho sobre el elemento (figura o tabla) y pulsamos insertar título, no se nos debe olvidar indicar la categoría correcta en cada caso, figura o tabla para que la autonumeración funcione correctamente.

### Hiperenlaces

Un primer paso a la hora de crear un documento es generar una versión en formato electrónico del mismo. Hemos decidido que ese formato sea pdf. En un formato pdf existe la posibilidad de crear hiperenlaces que facilitan la navegación a lo largo del mismo. Por ejemplo, el índice en un libro en formato pdf se generará, con la propuesta que hemos realizado, creando enlaces a las diversas partes del mismo. O bien, cuando nos referimos a una figura o tabla, es muy útil la existencia de esos enlaces al lugar exacto en el que se encuentra la figura o tabla. Esto lo realizaremos y veremos con más detalle en la siguiente sección. Si queremos incluir un enlace a un elemento externo (una página web, por ejemplo) deberemos ir a la pestaña *Insertar -> Hipervínculo*, nos aparecen dos campos para rellenar: en “Texto” pondremos el texto que apecerá en nuestro documento, por ejemplo Google, y en “Dirección” la url a la que apuntará, por ejemplo www.google.es, quedando el resultado final así: [Google](http://www.google.es).

### Tabla de contenido

La generación de la tabla (o tablas) de contenido de un texto suficientemente largo suele ser una tarea sumamente laboriosa. Word® facilita enormemente este trabajo. En otra sección de este capítulo explicaremos cómo y dónde se incorporará esta tabla de contenidos. En este apartado nos centramos en explicar algunos aspectos de cómo se construye la principal tabla de contenidos, que denominamos *Índice*.

Nuestra primera decisión fue establecer que en el índice deben aparecer hasta los apartados que hemos denominados subsecciones, lo que se logra mediante la secuencia que vamos a ver a continuación. También hemos propuesto que no aparezcan los habituales puntos que existen entre el texto y el número de página correspondiente de muchos índices.

Para usar el que índice incluido simplemente deberemos escribir nuestro texto usando los correspondientes estilos de títulos, a la hora de actualizarlo deberemos pulsar el botón derecho sobre el mismo y pulsar en *Actualizar Campos* y elegir *Actualizar todo.* También podemos ir al menú Insertas>Indice y Tablas y allí tabla de contenido, usando el estilo sencillo.

### Índice de figuras, tablas y otros elementos

Es habitual, y aquí lo hemos hecho también, incluir un índice de figuras y/o tablas al final del documento cuando éste posee muchos de estos elementos. También se pueden incluir índices de Código, Teoremas o cualquier otro elemento del estilo. Para crear un índice de este tipo debemos hacer lo siguiente: Ir al la pestaña *Referencias-> Insertar tabla de ilustraciones;* ahí debemos pinchar en el desplegable *etiqueta de título* y seleccionaremos el elemento para el cual queramos crear el índice. Otra posible opción es pulsar en el botón *Opciones*, marcar la pestaña estilo *estilo*  bajo Generar tabla de ilustraciones a partir de: y seleccionar el estilo del elemento del que queramos generar el índice.

### Formatos de páginas

El aspecto de un libro está básicamente definido por el formato que se ha elegido para los diferentes títulos de las partes que lo constituyen, el formato de las páginas y qué queremos que aparezca en las cabeceras y pies de páginas del mismo. En las páginas pares el nombre del capítulo en el que estamos y en las impares el nombre del documento. Estos elementos se colocan encima de una raya horizontal que se ha definido previamente, tanto en su grosor como en su longitud.

En todo caso, estos parámetros no se deberían de tocar, salvo en contadas ocasiones.

### Teoremas y otros elementos similares

En la escritura de cualquier texto científico los Teoremas, propiedades y demás elementos constituyen una parte muy significativa. Existen, de nuevo, múltiples posibilidades de tratar estos elementos, pero hemos considerado que el formato propuesto se adapta perfectamente a nuestros gustos y decisiones.

A modo de ejemplo hemos definido el Título Teorema X-Y, puede probar a crear los estilos Lemas, Propiedades y Corolarios si les hiciera falta. Centrándonos en los teoremas, el resultado sería el siguiente, donde se ha definido una etiqueta Teorema:

**Teorema 4–1** (Teorema de Pitágoras) *En un triángulo rectángulo...*

Demostración. Sea el triángulo ABC...

|  |  |
| --- | --- |
|  | (4–2) |

∎

Podemos observar que al finalizar la demostración hemos incluido el símbolo ∎.

### Ejemplos

En este apartado se muestra cómo meter un ejemplo. Esta forma propuesta es el resultado de jugar con Títulos y Estilos. Para citarlo, puede recurrir a Referencia Cruzada, siendo el resultado **Ejemplo *4*–*1***.

**Ejemplo 4–1.** Al insertar ecuaciones en un ejemplo, puede haber problemas con las barras grises superior e inferior.

### Índices de palabras y glosarios

Con los paquetes index y glossaries podemos incluir índices de palabras y listas con definiciones, ya sea de acrónimos u de otro tipo. Por ejemplo, se podría usar también para definir magnitudes o la notación utilizada.

#### Índices de palabras

Algunos conceptos son muy importantes para la Tesis y conviene añadirlos al glosario al menos la primera vez que aparecen. Para ello, los marcamos y pulsamos *Referencias > Marcar entrada.* Luego pulsaremos “Marcar” y “Cerrar”. Si Word® nos ha pasado al modo “Mostrar Todo”, donde se muestran los símbolos de formato, podemos salir de él pulsando *Control + Shift + (.*

#### Glosario

Para realizar un glosario deberá realizarse a mano respetando (si así se quiere), el formato propuesto en la sección correspondiente.

Tambíen se puede generar en Word de una forma más dinámica usando tabla de autoridades. Hacemos click en la parte del documento donde se quiere añadir el glosario y, posteriormente, en el apartado *referencias* de barra superior, en la sección *Tabla de autoridades*, seleccionamos la opción *Insertar* *Tabla de autoridades,* donde podremos configurar el formato que tendrá nuestro glosario y, al aceptar, declarar dónde se situará.

Para añadir entradas de forma dinámica basta con seleccionar la entrada que queremos añadir y pulsar en *Marcar cita*, en el mismo apartado donde hemos pulsado para crear la tabla. En la ventana que aparecerá añadimos la descripción de la entrada en el cuadro *Texto seleccionado* y pulsamos *Marcar.* Se puede ir actualizando el glosario pulsando F9.

En el caso de que al general el glosario aparezca un título, pulsando click derecho sobre el mismo y eligiendo la opción *Activar o desactivar códigos de campo* veremos una línea de código similar a {TOA \h \c "1"}.   
Quitando el \h eliminaremos ese título.

Si aparecieran símbolos en los saltos de linea o se sustituyeran los espacios por puntos, en el apartado *Inicio*, sección *Párrafo* pulsamos en Button imagey volveríamos a tener el documento como antes. Este botón hace que muestra/desaparecan esas marcas de formato.

## Antes del documento

Antes de empezar la edición del documento, además de cargar el fichero de estilo anterior, hemos creído necesario realizar una serie de operaciones que faciliten nuestro trabajo o lo configuren de una determinada manera. Además, hay que incluir la portada.

## Fuente del texto

Las instrucciones % establecen que el texto estará escrito en una fuente Times New Roman. El cambio de fuente se debe hacer modificando los estilos predefinidos en el documento.

## Cubierta

La primera hoja incluye una propuesta de cubierta. Puede modificarla, introduciendo una imagen central distinta, o introduciendo o sustituyendo algún logo y cambiando autores, títulos, departamento, etc.

# Anexos

Anexo A. Código Ground Truth GUI[[6]](#footnote-7)

Para facilitar la navegación por el código se añade el siguiente árbol de dependencias. En verde, los archivos que forman parte de librerías de python.

Figura X. Árbol de dependencias de Ground Truth GUI

main.py

*"""  
main.py se encarga de proveer a las ventanas de una lista que contenga el directorio de cada una de las imágenes. Además le pasa el índice para saber por qué imagen nos quedamos la última vez. También mantiene los bucles infinitos de las ventanas  
"""***import** os  
**import** win\_pygame  
  
**from** config **import** path\_to\_documents  
**from** config **import** path\_to\_index  
  
  
path = path\_to\_documents  
index\_path = path\_to\_index  
walk = os.walk(path)  
index\_file = open(index\_path + **'/'** + **'index.txt'**)  
index = int(index\_file.read())  
index\_file.close()  
  
doc\_paths = []  
**for** root, dirs, files **in** walk:  
 there\_is\_any\_img = **False  
 for** curr\_file **in** files:  
 **if** curr\_file.endswith(**".png"**):  
 there\_is\_any\_img = **True  
  
 if** there\_is\_any\_img:  
 root = root.replace(**"\\"**, **"/"**)  
 doc\_paths.append(root)  
  
doc\_win = win\_pygame.WinPygame(doc\_paths, index)  
  
**while** 1:  
 doc\_win.main\_loop()  
 doc\_win.control\_panel.root.update()  
 doc\_win.control\_panel.root.update\_idletasks()

win\_pygame.py

**import** os  
**import** sys  
**import** pygame  
**from** pygame\_functions **import** PygameFunctions **as** pf  
**import** win\_tkinter  
  
**from** config **import** path\_to\_documents  
**from** config **import** path\_to\_seals\_db  
  
  
**class** WinPygame:  
 *"""  
 Crea una instancia de la ventana de visualización de imágenes y maneja los controles de navegación.  
 Crea una instancia del panel de control  
 Guarda la imagen del nuevo sello en caso de que uno sea encontrado  
 """* **def** \_\_init\_\_(self, path=**None**, index=0):  
 self.size = self.width, self.height = 1200, 650  
 self.speed = [1, 1]  
 self.black = 0, 0, 0  
  
 pygame.init()  
   
 self.screen = pygame.display.set\_mode(self.size)  
  
 **if** path **is not None**:  
 onlyimages = [f **for** f **in** os.listdir(path[index])  
 **if** os.path.isfile(os.path.join(path[index], f)) **and** f.endswith(**'.png'**)]  
 self.doc\_img = pygame.image.load(path[index] + **'/'** + onlyimages[0])  
 **else**:  
 *# Random test image just for debugging* self.doc\_img = pygame.image.load(path\_to\_documents + **"/1823-L119.M3/117/IMG\_0001.png"**)  
  
 self.img\_c = self.doc\_img.copy()  
  
 self.scale = 1  
 self.originx, self.originy = (0, 0)  
 self.pt1x, self.pt1y = (0, 0)  
 self.pt2x, self.pt2y = (0, 0)  
 self.movement\_speed = 10  
 self.clock = pygame.time.Clock()  
  
 self.image = pygame.transform.scale(self.img\_c, (int(self.doc\_img.get\_width() \* self.scale),  
 int(self.doc\_img.get\_height() \* self.scale)))  
  
 *# Only create control panel if we are not testing* **if** \_\_name\_\_ != **"\_\_main\_\_"**:  
 self.control\_panel =\  
 win\_tkinter.WinControlPanel(self, (self.pt1x, self.pt1y, self.pt2x, self.pt2y), path, index)  
  
 **def** main\_loop(self):  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT **or** (event.type == pygame.KEYUP **and** event.key == pygame.K\_ESCAPE):  
 **if** \_\_name\_\_ == **"\_\_main\_\_"**: *# If there is a control panel, closing will be handled there.* sys.exit()  
 **elif** event.type == pygame.MOUSEBUTTONDOWN:  
 **if** event.button == 1 **or** event.button == 3:  
 pf.clear\_old\_rect(self.img\_c, self.doc\_img, (self.pt1x, self.pt1y, self.pt2x, self.pt2y),  
 event.button)  
 mouse\_coords = pygame.mouse.get\_pos()  
 self.pt1x, self.pt1y, self.pt2x, self.pt2y = \  
 pf.draw\_new\_rect((self.pt1x, self.pt1y, self.pt2x, self.pt2y), self.img\_c,  
 (self.originx, self.originy), self.scale, mouse\_coords, event.button)  
 **if** \_\_name\_\_ != **"\_\_main\_\_"**:  
 self.control\_panel.update\_labels((self.pt1x, self.pt1y, self.pt2x, self.pt2y))  
  
 **if** event.button == 4 **or** event.button == 5:  
 self.scale = pf.scale\_img(self.scale, event.button)  
  
 *# Transform image acording to translation and scale* self.originx, self.originy = pf.move\_image(self.originx, self.originy)  
 self.image = pygame.transform.scale(self.img\_c, (int(self.img\_c.get\_width() \* self.scale),  
 int(self.img\_c.get\_height() \* self.scale)))  
  
 self.screen.fill(self.black)  
 self.screen.blit(self.image, (self.originx, self.originy))  
 pygame.display.flip()  
  
 self.clock.tick(60)  
  
 **def** update\_img(self, path):  
 self.doc\_img = pygame.image.load(path)  
 self.img\_c = self.doc\_img.copy()  
 self.scale = 1  
 self.originx, self.originy = (0, 0)  
  
 **def** save\_seal(self, name):  
 width = self.pt2x - self.pt1x  
 height = self.pt2y - self.pt1y  
 cropped = pygame.Surface((abs(width), abs(height)))  
 cropped.blit(self.doc\_img, (0, 0), (self.pt1x, self.pt1y, width, height))  
  
 pygame.image.save(cropped, path\_to\_seals\_db + name + **'.png'**)  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 win = WinPygame()  
 **while** 1:  
 win.main\_loop()

pygame\_functions.py

**import** pygame  
  
  
**class** PygameFunctions:  
 MOVEMENT\_SPEED = 10  
 SCALE\_INCREMENT = 0.05  
  
 @staticmethod  
 **def** move\_image(orx, ory):  
 **if** pygame.key.get\_pressed()[pygame.K\_LEFT] != 0:  
 orx += PygameFunctions.MOVEMENT\_SPEED  
 **elif** pygame.key.get\_pressed()[pygame.K\_RIGHT] != 0:  
 orx -= PygameFunctions.MOVEMENT\_SPEED  
 **if** pygame.key.get\_pressed()[pygame.K\_UP] != 0:  
 ory += PygameFunctions.MOVEMENT\_SPEED  
 **elif** pygame.key.get\_pressed()[pygame.K\_DOWN] != 0:  
 ory -= PygameFunctions.MOVEMENT\_SPEED  
  
 **return** orx, ory  
  
 @staticmethod  
 **def** calc\_rect(coords):  
 minx = min(coords[0], coords[2])  
 maxx = max(coords[0], coords[2])  
 miny = min(coords[1], coords[3])  
 maxy = max(coords[1], coords[3])  
  
 **return** minx, miny, maxx, maxy  
  
 @staticmethod  
 **def** clear\_old\_rect(img, model\_img, coords, button):  
 rect\_coords = PygameFunctions.calc\_rect(coords)  
 img.blit(model\_img, (rect\_coords[0] - 5, rect\_coords[1] - 5),

area=(rect\_coords[0] - 5, rect\_coords[1] - 5,

rect\_coords[2] - rect\_coords[0] + 10,

rect\_coords[3] - rect\_coords[1] + 10))  
 **if** button == 1:  
 pygame.draw.circle(img, (0, 0, 255), (coords[2], coords[3]), 5)  
 **elif** button == 3:  
 pygame.draw.circle(img, (255, 0, 0), (coords[0], coords[1]), 5)  
  
 @staticmethod  
 **def** draw\_new\_rect(coords, img, img\_origin, scale, mouse\_coords, button):  
 newx, newy = (int((mouse\_coords[0] - img\_origin[0]) / scale),

int((mouse\_coords[1] - img\_origin[1]) / scale))  
 new\_coords = 0  
 **if** button == 1:  
 pygame.draw.circle(img, (255, 0, 0), (newx, newy), 5)  
 pygame.draw.rect(img, (0, 255, 0), (newx, newy, coords[2] - newx,

coords[3] - newy), 3)  
 new\_coords = (newx, newy, coords[2], coords[3])  
 **elif** button == 3:  
 pygame.draw.circle(img, (0, 0, 255), (newx, newy), 5)  
 pygame.draw.rect(img, (0, 255, 0), (newx, newy, coords[0] - newx,

coords[1] - newy), 3)  
 new\_coords = (coords[0], coords[1], newx, newy)  
  
 **return** new\_coords  
  
 @staticmethod  
 **def** scale\_img(scale, button):  
 **if** button == 4:  
 scale += PygameFunctions.SCALE\_INCREMENT  
 **elif** button == 5 **and** scale > 0:  
 scale -= PygameFunctions.SCALE\_INCREMENT  
  
 **return** scale

win\_tkinter.py

**import** tkinter **as** tk  
**from** tkinter **import** messagebox  
**import** database  
**import** widgets  
**import** os  
  
**from** config **import** path\_to\_documents  
**from** config **import** db\_credentials  
**from** config **import** path\_to\_index  
  
  
**class** WinControlPanel:  
 **def** \_\_init\_\_(self, img\_window, coords, paths, index=0):  
 self.root = tk.Tk()  
 self.root.geometry(**"500x570+30+30"**)  
   
 self.db = database.Database(db\_credentials[**'db\_name'**], db\_credentials[**'user\_name'**],  
 db\_credentials[**'pwd'**], db\_credentials[**'tables'**])  
 self.db.load\_seals()  
  
 self.coords = coords  
 self.paths = paths  
 self.path\_index = index  
   
 *# POINT 1 DISPLAY* self.pt1\_label = tk.Label(self.root, text=**"Point 1:"**)  
 self.pt1\_label.place(x=20, y=10)  
   
 self.pt1x\_value = tk.IntVar()  
 self.pt1x\_info = tk.Entry(self.root, state=tk.DISABLED, textvariable=self.pt1x\_value)  
 self.pt1x\_info.place(x=70, y=10, width=50)  
 self.pt1x\_value.set(str(coords[0]))  
   
 self.pt1y\_value = tk.IntVar()  
 self.pt1y\_info = tk.Entry(self.root, state=tk.DISABLED, textvariable=self.pt1y\_value)  
 self.pt1y\_info.place(x=130, y=10, width=50)  
 self.pt1y\_value.set(str(coords[1]))  
  
 *# POINT 2 DISPLAY* self.pt2\_label = tk.Label(self.root, text=**"Point 2:"**)  
 self.pt2\_label.place(x=20, y=30)  
   
 self.pt2x\_value = tk.IntVar()  
 self.pt2x\_info = tk.Entry(self.root, state=tk.DISABLED, textvariable=self.pt2x\_value)  
 self.pt2x\_info.place(x=70, y=30, width=50)  
 self.pt2x\_value.set(str(coords[2]))  
   
 self.pt2y\_value = tk.IntVar()  
 self.pt2y\_info = tk.Entry(self.root, state=tk.DISABLED, textvariable=self.pt2y\_value)  
 self.pt2y\_info.place(x=130, y=30, width=50)  
 self.pt2y\_value.set(str(coords[3]))  
  
 *# NEW SEAL WINDOW* **def** on\_new\_seal():  
 self.new\_seal\_win = WinNewSeal(self.db, img\_window)  
 self.new\_seal\_butt = tk.Button(self.root, text=**'New'**, command=on\_new\_seal)  
 self.new\_seal\_butt.place(x=200, y=20)  
  
 *# SEAL SELECTION ITEMS* self.seal\_type\_list = widgets.SealsList(self.root, self.db)  
 self.seal\_type\_list.x = 20  
 self.seal\_type\_list.y = 70  
 self.seal\_type\_list.place\_items()  
  
 *# OK BUTTON* **def** on\_ok\_button():  
 self.db.insert\_document(self.paths[self.path\_index],  
 self.seal\_type\_list.curr\_seal\_type.get(),  
 (self.pt1x\_value.get(), self.pt1y\_value.get(),  
 self.pt2x\_value.get(), self.pt2y\_value.get()))  
  
 **if** self.path\_index < len(self.paths):  
 self.path\_index += 1  
 onlyimages = [f **for** f **in** os.listdir(self.paths[self.path\_index])  
 **if** os.path.isfile(os.path.join(self.paths[self.path\_index], f)) **and** f.endswith(**'.png'**)]  
  
 *# Llamar método de ventana de pygame que actualiza a la nueva imagen* **if** \_\_name\_\_ != **"\_\_main\_\_"**:  
 img\_window.update\_img(self.paths[self.path\_index] + **'/'** +  
 onlyimages[0])  
 **else**:  
 messagebox.showinfo(**"End of classification"**, **"There are no more documents to classify"**)  
  
 self.ok\_button = tk.Button(self.root, text=**'OK'**,command=on\_ok\_button)  
 self.ok\_button.place(x=250, y=20)  
  
 **def** on\_closing():  
 **if** messagebox.askokcancel(**"Quit"**, **"Do you want to quit?"**):  
 index\_file = open(path\_to\_index + **'/index.txt'**, **'w'**)  
 index\_file.write(str(self.path\_index))  
 index\_file.close()  
 self.root.destroy()  
  
 self.root.protocol(**"WM\_DELETE\_WINDOW"**, on\_closing)  
  
 **def** update\_labels(self, new\_coords):  
 self.pt1x\_value.set(str(new\_coords[0]))  
 self.pt1y\_value.set(str(new\_coords[1]))  
 self.pt2x\_value.set(str(new\_coords[2]))  
 self.pt2y\_value.set(str(new\_coords[3]))  
  
  
**class** WinNewSeal:  
 **def** \_\_init\_\_(self, db, img\_win):  
 self.db = db  
 self.img\_win = img\_win  
  
 self.root = tk.Tk()  
 self.width = 300  
 self.height = 100  
 geom\_str = **"%ix%i+30+30"** % (self.width, self.height)  
 self.root.geometry(geom\_str)  
  
 *# SEAL INFO* self.name\_label = tk.Label(self.root, text=**'Nombre'**)  
 self.name\_label.place(x=20, y=10)  
 *# self.seal\_name = tk.StringVar()* self.name\_info = tk.Entry(self.root) *# , textvariable=self.seal\_name)* self.name\_info.place(x=70, y=10)  
  
 self.author\_label = tk.Label(self.root, text=**'Autor'**)  
 self.author\_label.place(x=20, y=40)  
 *# self.seal\_author = tk.StringVar()* self.author\_info = tk.Entry(self.root) *# , textvariable=self.seal\_author)* self.author\_info.place(x=70, y=40)  
 *# OK BUTTON* self.ok\_button = tk.Button(self.root, text=**'OK'**, command=self.on\_ok\_button)  
 self.ok\_button.place(x=200, y=20)  
  
 **def** on\_ok\_button(self):  
 self.db.insert\_seal(self.name\_info.get(), self.author\_info.get())  
 self.img\_win.save\_seal(self.name\_info.get())  
 self.root.destroy()  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 path = path\_to\_documents  
 walk = os.walk(path)  
  
 doc\_paths = []  
 **for** root, dirs, files **in** walk:  
 there\_is\_any\_img = **False  
 for** curr\_file **in** files:  
 **if** curr\_file.endswith(**".png"**):  
 there\_is\_any\_img = **True  
  
 if** there\_is\_any\_img:  
 root = root.replace(**"\\"**, **"/"**)  
 doc\_paths.append(root)  
  
 win = WinControlPanel(**None**, (10, 20, 30, 40), doc\_paths)  
 **while** 1:  
 win.root.update\_idletasks()  
 win.root.update()

widgets.py

*"""  
Archivo destinado a agrupar elementos de la interfaz que deban ir todos juntos. Al final tan sólo contiene la lista de sellos incluyendo el menú desplegable, la lista de éstos, la imagen de muestra correspondiente a dicho sello y el botón que sirve para cambiar dicha imagen.  
"""***import** tkinter **as** tk

**class** SealsList:  
 **def** \_\_init\_\_(self, master, db):  
 self.db = db  
 self.x = 0  
 self.y = 0  
 self.label = tk.Label(master, text=**'Type:'**) *# <--LABEL* self.OPTIONS = [seal.name **for** seal **in** self.db.seal\_list]  
 self.curr\_seal\_type = tk.StringVar(master)  
 self.curr\_seal\_type.set(self.OPTIONS[0])  
 self.option\_menu = tk.OptionMenu(master, self.curr\_seal\_type, \*self.OPTIONS) *# <--OPTION MENU* self.canvas = tk.Canvas(master, width=600, height=400) *# <--CANVAS* self.img\_route = self.db.seal\_list[0].img\_route  
 self.img\_route = self.img\_route.replace(**"\\"**, **"/"**)  
 photo = tk.PhotoImage(file=self.img\_route)  
 self.seal\_img = self.canvas.create\_image(0, 0, anchor=tk.NW, image=photo)  
 self.canvas.image = photo  
  
 self.change\_button = tk.Button(master, text=**"Change"**, command=self.on\_change\_item)  
  
 **def** on\_change\_item(self):  
 index = self.OPTIONS.index(self.curr\_seal\_type.get())  
 *# change image* path = self.db.seal\_list[index].img\_route  
 path = path.replace(**"\\"**, **"/"**)  
 photo = tk.PhotoImage(file=path)  
 self.canvas.itemconfig(self.seal\_img, image=photo)  
 self.canvas.image = photo  
  
 **def** place\_items(self):  
 self.label.place(x=self.x, y=self.y)  
 self.option\_menu.place(x=self.x + 35, y=self.y)  
 self.canvas.place(x=self.x + 35, y=self.y + 80)  
 self.change\_button.place(x=self.x + 160, y=self.y+3)

database.py

**import** pymysql  
**import** seal  
  
**from** config **import** path\_to\_seals  
  
  
**class** Database:  
 **def** \_\_init\_\_(self, db\_name, user\_name, passwd, tables):  
 self.seal\_list = [] *# list of currently stored seals* self.db = pymysql.connect(**"localhost"**, user\_name, passwd, db\_name)  
 self.cursor = self.db.cursor()  
 self.table\_names = tables  
  
 **def** load\_seals(self):  
 sql = **"select \* from %s"** % self.table\_names[0]  
  
 **try**:  
 self.cursor.execute(sql)  
 result = self.cursor.fetchall()  
  
 **for** row **in** result:  
 name = row[0]  
 author = row[1]  
 route = row[2]  
 new\_seal = seal.Seal(route, name, author)  
 self.seal\_list.append(new\_seal)  
 **except** RuntimeError:  
 print(**"error: unable to fetch seals data\n"**)  
  
 **def** insert\_seal(self, name, author):  
 path = path\_to\_seals + name + **".png"** sql = **"""insert into %s(nombre, autor, ruta, añadido\_manual)  
 values ('%s', '%s', '%s', 1)"""** % (self.table\_names[0], name, author, path)  
  
 **try**:  
 self.cursor.execute(sql)  
 self.db.commit()  
 **except** RuntimeError:  
 self.db.rollback()  
 print(**"unable to insert data into database\n"**)  
  
 **def** insert\_document(self, route, seal\_name, coords):  
 sql = **"""insert into %s(ruta, sello, coordenadas\_x1, coordenadas\_y1, coordenadas\_x2, coordenadas\_y2)  
 values ('%s', '%s', '%s', '%s', '%s', '%s')"""** % (self.table\_names[1], route, seal\_name,  
 coords[0], coords[1], coords[2], coords[3])  
  
 **try**:  
 self.cursor.execute(sql)  
 self.db.commit()  
 **except** RuntimeError:  
 self.db.rollback()  
 print(**"unable to insert data into database\n"**)

seal.py

**class** Seal:  
 **def** \_\_init\_\_(self, img\_route=**''**, name=**''**, author=**''**):  
 self.img\_route = img\_route  
 self.name = name  
 self.author = author

Anexo B. Tests de elección de umbral de binarizado

Para calcular el mejor sistema de binarizado se ha diseñado esta aplicación que permite cargar una imagen de prueba y ver en tiempo real el resultado de cada binarizado distinto que se aplique. El sistema permite probar un umbral fijo, el método de Otsu o el método adaptativo mediante media ponderada con kernel gaussiano. Además permite realizar una distorsión gaussiana para eliminar ruidos. El resultado se puede almacenar pulsando la tecla TAB (elegida arbitrariamente) para así tener almacenados resultados y poder comparar los distintos métodos. Pulsar ESC termina el la ejecución.

bin\_test.py

**import** cv2  
**import** os  
**from** paths **import** path\_to\_imgs  
**from** Umbralizacion **import** Umbralizacion  
  
  
**def** do\_nothing(x):  
 **pass**file = path\_to\_imgs + **'1862-L119.M13/109/IMG\_0001.png'**img = cv2.imread(file)  
nombre = os.path.splitext(os.path.basename(file))[0]  
  
cv2.namedWindow(**'image\_window'**, cv2.WINDOW\_NORMAL)  
cv2.namedWindow(**'control\_window'**, cv2.WINDOW\_OPENGL)  
  
METHODLABEL = **'1:OTSU-2:FIX-3:ADAP'**cv2.createTrackbar(**'Vecinity width'**, **'control\_window'**, 0, 10, do\_nothing)  
cv2.createTrackbar(**'Vecinity height'**, **'control\_window'**, 0, 10, do\_nothing)  
cv2.createTrackbar(**'Sigma X'**, **'control\_window'**, 0, 100, do\_nothing)  
cv2.createTrackbar(**'Sigma Y'**, **'control\_window'**, 0, 100, do\_nothing)  
cv2.createTrackbar(METHODLABEL, **'control\_window'**, 0, 2, do\_nothing)  
cv2.createTrackbar(**'Bin Threshold'**, **'control\_window'**, 0, 255, do\_nothing)  
cv2.createTrackbar(**'SWITCH'**, **'control\_window'**, 0, 1, do\_nothing)  
  
gray\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
filtered\_img = gray\_img  
bin\_img = Umbralizacion.umbralizar\_imagen(filtered\_img, Umbralizacion.MetodoUmbralizado.fixed, 180)  
  
**while** 1:  
 cv2.imshow(**'image\_window'**, bin\_img)  
  
 vec\_w = cv2.getTrackbarPos(**'Vecinity width'**, **'control\_window'**)  
 vec\_h = cv2.getTrackbarPos(**'Vecinity height'**, **'control\_window'**)  
 sigX = cv2.getTrackbarPos(**'Sigma X'**, **'control\_window'**)  
 sigY = cv2.getTrackbarPos(**'Sigma Y'**, **'control\_window'**)  
 method = cv2.getTrackbarPos(METHODLABEL, **'control\_window'**)  
 threshold = cv2.getTrackbarPos(**'Bin Threshold'**, **'control\_window'**)  
 switch = cv2.getTrackbarPos(**'SWITCH'**, **'control\_window'**)  
  
 method\_enum = Umbralizacion.MetodoUmbralizado(method)  
  
 kernel = (vec\_w\*2 + 1, vec\_h\*2 + 1)  
 **if** switch == 1:  
 filtered\_img = cv2.GaussianBlur(gray\_img, kernel, sigX)  
 bin\_img = Umbralizacion.umbralizar\_imagen(filtered\_img, method\_enum, threshold)  
  
 k = cv2.waitKey(1) & 0xFF  
 **if** k == 27:  
 **break  
 elif** k == 9: *# TAB key* filestring = **'../%s\_met\_%d\_vec\_%d\_sig\_%d\_thr\_%d.png'** % (nombre, method, vec\_w, sigX, threshold)  
 *# cv2.imwrite(filestring, bin\_img)*cv2.destroyAllWindows()

Umbralizacion.py

**import** cv2  
**from** enum **import** Enum  
  
  
**class** Umbralizacion:  
 **class** MetodoUmbralizado(Enum):  
 otsu = 0  
 fixed = 1  
 adaptive = 2  
  
 @staticmethod  
 **def** umbralizar\_imagen(imagen\_grises, metodo\_umbralizar, umbral=180):  
 imagen\_umbralizada = imagen\_grises  
  
 **if** metodo\_umbralizar == Umbralizacion.MetodoUmbralizado.otsu:  
 ret, imagen\_umbralizada = cv2.threshold(imagen\_grises, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)  
 **elif** metodo\_umbralizar == Umbralizacion.MetodoUmbralizado.fixed:  
 ret, imagen\_umbralizada = cv2.threshold(imagen\_grises, umbral, 255, cv2.THRESH\_BINARY)  
 **elif** metodo\_umbralizar == Umbralizacion.MetodoUmbralizado.adaptive:  
 imagen\_umbralizada = cv2.adaptiveThreshold(imagen\_grises, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, cv2.THRESH\_BINARY, 11, 2)  
  
 **return** imagen\_umbralizada

# Referencias

|  |  |
| --- | --- |
| [1] | Autor, «Este es el ejemplo de una cita,» *Tesis Doctoral,* vol. 2, nº 13, 2012. |
| [2] | O. Autor, «Otra cita distinta,» *revista,* p. 12, 2001. |

# **Índice de Conceptos**

# **Glosario**

ISO: International Organization for Standardization 4

UNE: Una Norma Española 4

1. Fundación Osborne es el vehículo canalizador de la Responsabilidad Social Corporativa del Grupo Osborne. Posee dos fines fundamentales: la preservación y difusión del patrimonio histórico de la compañía y la promoción de la formación y el emprendimiento en las áreas de influencia del Grupo Osborne. [↑](#footnote-ref-2)
2. La Fundación Telefónica tiene por misión mejorar las oportunidades de desarrollo de las personas a través de proyectos educativos, sociales y culturales, adaptados a los retos del mundo digital. [↑](#footnote-ref-3)
3. La Fundación SEPI, entre otras actividades, promueve y gestiona un extenso Programa de Becas de Inserción Laboral que cada año facilita la formación práctica en las empresas a más de 300 jóvenes titulados. [↑](#footnote-ref-4)
4. Esto se hace desde el menu *Referencias > Insertar nota al pie*. [↑](#footnote-ref-5)
5. En MacOS no suele aparecer y hay que recurrir a algún foro para ver cómo incluirla, por ejemplo http://ask.brothersoft.com/how-to-get-the-missing-toolbar-and-menu-back-331.html [↑](#footnote-ref-6)
6. Nota aclarativa: los códigos presentes en este documento pueden formar parte de proyectos en continuo desarrollo. La versión mostrada a continuación no tiene por qué corresponderse con la versión actual en el momento de lectura. En cualquier caso, la versión más actual debería encontrarse disponible en los repositorios especificados, y todos ellos disponibles en DIRECCIÓN DE MI GITHUB NUEVO [↑](#footnote-ref-7)