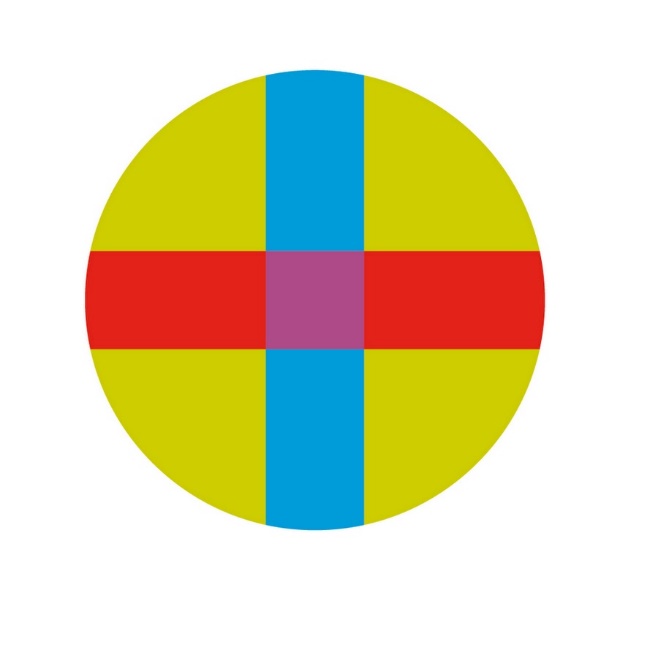
UNIVERSIDAD SAN PABLO - CEU

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA DE SISTEMAS DE INFORMACIÓN



TRABAJO FIN DE GRADO

**AUTOMATIZACIÓN DE INFORMES DE TOPOGRAFÍA DE LA LESIÓN TALÁMICA MEDIANTE ULTRASONIDO FOCAL DE ALTA INTENSIDAD**

Autor: Enrique Collada Sánchez

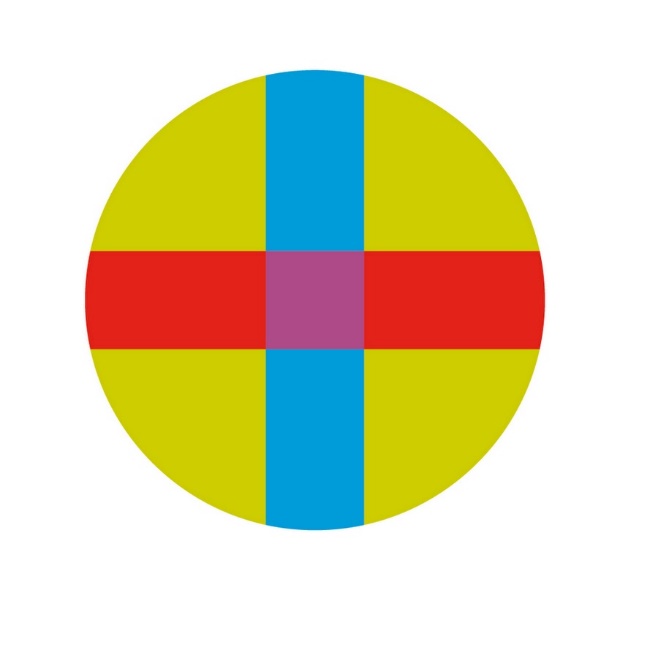
Director: José Ángel Pineda-Pardo

Junio 2021

UNIVERSIDAD SAN PABLO - CEU

ESCUELA POLITÉCNICA SUPERIOR

TÍTULO DEL GRADO



TRABAJO FIN DE GRADO

**TÍTULO DEL TFG**

Autor: Nombre y apellidos del autor

Director: Nombre y apellidos del director

mes y año de la convocatoria

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos del alumno   |  | | --- | | NOMBRE: |   Datos del Trabajo   |  | | --- | | TÍTULO DEL PROYECTO: |   Tribunal calificador   |  |  | | --- | --- | | PRESIDENTE: | FDO.: |      |  |  | | --- | --- | | SECRETARIO: | FDO.: |      |  |  | | --- | --- | | VOCAL: | FDO.: |  |  | | --- | | Reunido este tribunal el \_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_, acuerda otorgar al Trabajo Fin de Grado presentado por Don\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_la calificación de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. | |

RESUMEN

ABSTRACT

ÍNDICE

[1. Introducción 10](#_Toc70958656)

[1.1. Motivación del trabajo 10](#_Toc70958657)

[1.2. Objetivos del proyecto 11](#_Toc70958658)

[1.3. Metodología y resultados 11](#_Toc70958659)

[1.4. Organización de la memoria 11](#_Toc70958660)

[2. Estado del arte 12](#_Toc70958661)

[3. Gestión del proyecto 13](#_Toc70958662)

[4. Análisis 14](#_Toc70958663)

[4.1. Alcance 14](#_Toc70958664)

[4.2. Roles 14](#_Toc70958665)

[4.3. Especificación de requisitos 14](#_Toc70958666)

[5. Diseño 16](#_Toc70958667)

[5.1. Introducción 16](#_Toc70958668)

[5.2. Front-end 16](#_Toc70958669)

[5.3. Back-end 16](#_Toc70958670)

[6. Implementación 18](#_Toc70958671)

[6.1. Introducción 18](#_Toc70958672)

[6.2. Visualización de las imágenes NIFTI 18](#_Toc70958673)

[6.3. Muestreo de la lesión 21](#_Toc70958674)

[6.4. Ajustes de visualización de imágenes NIFTI 22](#_Toc70958675)

[6.5. Automatización de los comandos de FSLEYES 27](#_Toc70958676)

[6.6. Ajuste de las dimensiones de las imágenes 28](#_Toc70958677)

[6.7. Generación de gráficos 31](#_Toc70958678)

# Introducción

Un 11,2% de la población europea sufre enfermedades neurológicas, atendiendo a los datos de la OMS[[1]](#footnote-1). El temblor esencial es uno de los desórdenes del movimiento más frecuentes, afectando al 1% de la población aproximadamente y con un fuerte componente genético. Este desorden se caracteriza por temblores frecuentemente aislados a las extremidades superiores.

En la actualidad, la utilización de ultrasonido focal de alta intensidad (HIFU) muestra resultados exitosos a la hora de mejorar la sintomatología de pacientes con temblor esencial. La termoablación por HIFU guiada por resonancias magnéticas, ya se utiliza para el tratamiento tanto de la enfermedad de Parkingson como de los temblores esenciales.

Para el tratamiento de este tipo de enfermedades neurológicas, la barrera hemoencefálica, que protege al cerebro de la llegada de sustancias tóxicas o extrañas, viene obstaculizando la utilización de medicamentos para hacer frente a la enfermedad. Por otra parte, la cirugía en el cerebro presenta altos riesgos. El HIFU guiado por resonancia magnética es un procedimiento guiado por imagen mínimamente invasivo, que se realiza sin incisiones y con una gran precisión. Por ello, es esta una técnica muy prometedora.

Para mejorar la eficacia de esta técnica, en relación con los resultados terapéuticos, es necesario conocer con precisión la localización, el volumen y otras características de la lesión producida por HIFU en el tálamo para reducir la sintomatología del temblor esencial. Una predicción subóptima de estos parámetros puede provocar resultados terapéuticos poco deseados y efectos neurológicos adversos. Por tanto, se hace necesaria la creación de herramientas que permitan perfeccionar la localización y volumen de la lesión a partir de la experiencia y los resultados clínicos ya existentes.

# Motivación del trabajo

La simbiosis entre las nuevas tecnologías informáticas y las propias de la medicina son, cada vez de forma más evidente, una tendencia que multiplica las posibilidades de los investigadores y del personal sanitario para encontrar nuevas soluciones para curar enfermedades que afectan a gran parte de la población, mejorar la calidad de vida de los pacientes o hacer diagnósticos más certeros.

La técnica HIFU aplicada como tratamiento para los temblores esenciales es una tecnología disruptiva que, produciendo una lesión en el tálamo por termoablación, consigue prometedores resultados a la hora de reducir la sintomatología causada por esta enfermedad. Sin embargo, los resultados muestran una gran variabilidad dependiendo de las características de la lesión. Por ello, es de vital importancia conocer las características de la lesión para tomar decisiones clínicas y mejorar los resultados derivados de la aplicación de esta técnica.

Este es el motivo principal por el que este trabajo, que consiste en la automatización de informes clínicos de topografía de la lesión talámica, supone un avance que contribuye a una mayor eficacia en la utilización de esta técnica tan prometedora y que dota de una nueva herramienta informática al personal clínico, que le permita predecir los resultados post-tratamiento a partir de los datos de otros pacientes, para llevar a cabo su trabajo con mayor facilidad e información comprensible en la toma de decisiones.

# Objetivos del proyecto

Este trabajo tiene como objetivo proporcionar una herramienta informática al personal sanitario que permita integrar la información referente a la aplicación de la tecnología HIFU guiada por resonancia magnética, en un informe clínico que sea útil para conocer la realidad del paciente tras el tratamiento.

El informe producido garantiza al personal sanitario la correcta visualización de diferentes imágenes de resonancia magnética, donde reconocer con facilidad la lesión talámica producida por HIFU.

Además, se proporciona una caracterización comprensiva de los aspectos topográficos y morfológicos de la lesión talámica, con el fin de identificar con precisión aspectos relacionados con la localización y el volumen de esta.

Por último, con el objetivo de analizar la relación entre las características de la lesión y los resultados clínicos, se presenta una comparación estadística con una población de pacientes a los que ya ha sido realizada esta cirugía y de los que conocemos el porcentaje de mejora sobre la sintomatología producida por los temblores esenciales.

De esta forma, conseguimos dotar al personal sanitario con una herramienta que le permita predecir el resultado del tratamiento y conocer con detalle las características con que debe cumplir la lesión en posteriores tratamientos para conseguir una elevada eficacia ante esta frecuente enfermedad.

# Metodología y resultados

# Organización de la memoria

# Estado del arte

# Gestión del proyecto

# Análisis

# Alcance

Este proyecto proporciona una herramienta capaz de integrar en un informe clínico información post-tratamiento de una talamotomía termoablativa mediante ultrasonido focal de alta intensidad (HiFu). Esta información debe contener, en primer lugar, las imágenes de resonancias magnéticas donde pueda distinguirse la lesión. En segundo lugar, las características topográficas de la lesión y, en tercer lugar, una evaluación comparativa sobre una población de pacientes que ya fueron tratados.

# Roles

En este caso, habrá dos tipos de usuarios de la herramienta:

1. **El gestor o administrador**, que será el encargado de recopilar las diferentes resonancias magnéticas y de realizar la segmentación manual de la lesión, para después ejecutar el script que genera el informe clínico.
2. **El facultativo**, que utilizará el informe generado para evaluar la situación del paciente.

# Especificación de requisitos

**GENERACIÓN DE UN INFORME**

RF01. Generar de forma automática un informe clínico en formato PDF.

RF02. Generar de forma automática un informe clínico en formato HTML.

RF03. Incluir en el informe imágenes de la lesión talámica en a) T1, b) T2, c) FLAIR y d) SWAN en ortovisor, es decir, incluyendo en cada una de ellas la imagen axial, sagital y coronal.

RF04. Incluir en el informe 4 imágenes de tipo mosaico de la resonancia magnética en T1, T2, FLAIR y SWAN.

RF05. Las imágenes deben aprovechar el máximo espacio del folio, eliminando en lo posible los márgenes.

RF06. Las imágenes mostrarán en color rojo el contorno de la lesión.

RF07. En las imágenes en ortovisor, se eliminará el máximo ruido posible, aumentando el tamaño de las imágenes axial, sagital y coronal sin llegar a solapar ninguna de ellas.

*RF08. El informe deberá comenzar con una portada que incluya: nombre y apellidos del paciente, …*

*RF09. Las imágenes del informe deben ser modificadas para aumentar su contraste. (poco concreto)*

RF10. Tanto el fondo de las imágenes en ortovisor, como el de la imagen en mosaico debe ser de color negro.

*RF12. Incluir un apartado de características topográficas y morfológicas de la lesión que contenga: a) coordenadas del centroide, b) volumen de la lesión …*

# Diseño

# Introducción

Tras realizar el análisis de los requisitos, en este epígrafe se procede a explicar la arquitectura y estructura que hacen posible la integración de la información en un informe único final. Para llegar a ese punto, se han utilizado diferentes herramientas que permiten manipular y proporcionas distintas fuentes de información.

Por un lado, se puede hablar de un *front-end* haciendo referencia al formato de presentación del informe y un *back-end* que cuenta con tres componentes que podríamos clasificar en a) visualización de imágenes, b) extracción de características de la lesión y c) estadísticas y comparativas (sobre la población de pacientes).

# Front-end

La capa de presentación o front-end ha sido implentada utilizando HTML, CSS y Python, utilizando las librerías pdfkit, jinja2 y Pillow. El diseño del informe debe ser tal que proporcione una integración ordenada de distintos tipos de información en forma de a) imágenes, b) gráficos, c) datos numéricos, d) textos y e) tablas. El objetivo principal del informe es el de proporcionar al facultativo una herramienta que le permita visualizar y comprender con facilidad la situación de un paciente que se ha sometido a una termoablación talámica a través de HIFU, de tal forma que pueda identificar con facilidad la lesión a partir de distintas imágenes de resonancias magnéticas, conocer las características de esta lesión y evaluar la situación del paciente con respecto a una población estadística que permita, además, predecir los resultados del tratamiento.

El informe estará dividido en N bloques:

1. **Datos personales del paciente**.
2. **Visualización de las imágenes de resonancia magnética**. En este bloque se encontrarán las imágenes en ortovisor en T1, T2, FLAIR y SWAN, además de una imagen mosaico.
3. **Características de la lesión.** En este bloque se aportará información sobre las coordenadas del centroide, el volumen de la lesión
4. **Estadísticas sobre una población de pacientes.** Evaluación comparativa de la lesión y el porcentaje de mejora de los temblores esenciales de los pacientes mostrados en gráficos.

AQUÍ PONER UNA IMAGEN-BOCETO DE CÓMO SERÍA EL INFORME

# Back-end

* + 1. **Visualización de las imágenes de resonancia magnética**

Para la visualización de las imágenes de resonacia magnética con el formato requerido se utilizará la herramienta fsleyes. Esta herramienta permite manipular las imágenes NIFTI de las resonancias magnéticas de tal forma que pueda destacarse el contorno de la lesión en las coordenadas en las que se encuentra, ajustar el zoom y el contraste y convertirla a formato png, entre otras funciones. Todas las modificaciones sobre las imágenes NIFTI originales se harán a través de la línea de comandos, en una sentencia que posteriormente será ejecutada por un script de Python.

Para encontrar las coordenadas de la lesión se utilizará otra aplicación de visualización de imágenes NIFTI llamada ITK-SNAP.

* + 1. **Extracción de características de la lesión**

Utilizando Matlab…

* + 1. **Estadísticas**

# Implementación

# Introducción

Como se viene explicando en los epígrafes anteriores, el objetivo de este proyecto es la creación de una herramienta que permita integrar información de distinto tipo en un único informe clínico. Para ello, antes de poder generar este informe, ha sido necesario trabajar en distintas fases que proporcionasen la información requerida, en el formato especificado en el análisis de requisitos, para posteriormente integrarla de forma automática al ejecutar un script.

En esta sección, se explicará con detalle cada una de las fases, las herramientas y el código utilizados hasta llegar al script final con el que generar el informe.

# Visualización de las imágenes NIFTI

Las imágenes de resonancia magnética que utilizaremos están en formato NIFTI, que es uno de los diferentes formatos en que estas imágenes pueden ser almacenadas. Consisten en un único fichero de unos 20-50 MB que contiene todas las *slices* de la prueba. Lógicamente, estas imágenes deben visualizarse en un visor específico y, en este caso, utilizaremos la herramienta ITK-SNAP. Además, las imágenes se mostrarán en ortovisor, una vista compuesta por una imagen axial, una sagital y una coronal.

Para este informe, en la especificación de requisitos, se concreta que deben incluirse imágenes en T1, T2, FLAIR y SWAN. Las imágenes en T1 y en T2 se diferencian en las propiedades del tejido que se potencian. En ambas, se obtiene la respuesta de todo el tejido cerebral a la temperatura (no solo la respuesta de la lesión). Suponemos que la zona que segmentamos es la más probable de ser lesión por su respuesta a la temperatura. En el caso de T1, se obtiene una imagen hipertensa con más resolución. En el caso de T2, se puede observar de forma más intensa el edema de la lesión.

La imagen FLAIR tiene más contraste que las anteriores porque elimina la señal de los líquidos y esto hace que se muestre de forma más clara el edema.

A continuación, se muestra una imagen en T2 utilizando ITK\_SNAP:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamenteOtro de los requisitos del informe que queremos generar, es la visualización del contorno de la lesión talámica. Para este caso, se deben superponer una imagen NIFTI con otra que contenga la segmentación (manual) de la lesión[[2]](#footnote-2). Esto se consigue con la opción *Open Segmentation* que encontramos en el menú superior de ITK-SNAP. A continuación, se muestra la visualización de ambas imágenes superpuestas:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Imagen que contiene Diagrama

Descripción generada automáticamenteEn la figura X, se puede observar en la imagen axial un punto de color rojo. Esta es la lesión. Sin embargo, no se encuentra en las coordenadas que nos permitan ver la lesión en las imágenes sagital y coronal. Para ello, hacemos doble click sobre la lesión marcada en rojo y automáticamente nos situará en las coordenadas adecuadas (que más adelante utilizaremos):

Ahora ya podemos visualizar la lesión en las tres imágenes y, en el recuadro amarillo, se muestran las coordenadas de la lesión.

Este mismo proceso se puede realizar para visualizar las imágenes en T1, FLAIR y SWAN, pero solo tenemos la muestra de la lesión en T2, con lo que debemos “remuestrear” (resample) para poder visualizar la lesión en el resto de imágenes. Esto se conoce como una transformación de identidad.

# Muestreo de la lesión

Como queda explicado en el apartado anterior, necesitamos un archivo con la lesión de cada una de las distintas resonancias para superponerlas y visualizar con claridad la lesión producida por la termoablación. La lesión es en realidad una máscara, con *voxels* que tienen asignados valores de 0 (donde no hay lesión) y 1 (para el espacio donde se encuentra la lesión), puesto que no existe interpolación.

La única máscara que se nos proporciona es la de la lesión en T2. Para transformar el espacio de coordenadas de una imagen a otro, se tendría que realizar un registro de imagen. Sin embargo, en este caso asumimos que todas las imágenes (en T1, T2, FLAIR y SWAN) se encuentran en el mismo espacio de coordenadas, pero tienen distintas dimensiones. Cada *voxel* tiene coordenadas en milímetros, pero estas no coinciden entre las distintas imágenes. Por eso necesitamos *resamplear* o remuestrar la lesión para las distintas imágenes. Esto se conoce como transformación de identidad.

Queremos que solo se *rellenen con 1* los voxels más cercanos en la transformación. Esto se llama *interpolación de vecinos cercanos.*

Para obtener la segmentación de la lesión en T1, FLAIR y SWAN mediante una transformación de identidad, se utiliza la herramienta ANTs (*Advanced Normalization Tools*). Tal y como puede encontrarse en su web[[3]](#footnote-3): “*ANTs extrae información de conjuntos de datos complejos que incluyen imágenes (Word Cloud). Junto con ANTsR (respuesta), ANTs es útil para gestionar, interpretar y visualizar datos multidimensionales. ANTs se considera popularmente un conjunto de herramientas de registro y segmentación de imágenes médicas de última generación. ANTsR es una herramienta emergente que soporta el análisis estandarizado de imágenes multimodales. ANTs depende del Insight ToolKit (ITK), una biblioteca de procesamiento de imágenes médicas ampliamente utilizada a la que contribuyen los desarrolladores de ANTs*”.

Antes de instalar ANTs, es necesario configurar la línea de comandos, instalar un compilador específico e instalar CMake[[4]](#footnote-4). Todo esto puede encontrarse en la guía de instalación de ANTs (en el caso de este proyecto, para el sistema operativo macOS)[[5]](#footnote-5).

Una vez realizado el proceso de instalación de ANTs con éxito, la herramienta a utilizar será *WarpImageMultiTransform*. La estructura de la sentencia a ejecutar en la línea de comandos es la siguiente:

./WarpImageMultiTransform ImageDimension moving\_image output\_image **-R** reference\_image **--use-NN** --reslice-by-header

1. ImageDimension: las dimensiones de las imágenes (3, en este caso).
2. moving\_image: la segmentación de la lesión que tenemos.
3. Output\_image: la segmentación de la lesión que queremos producir.
4. Reference\_image: el archivo NIFTI sobre el que queremos superponer la nueva máscara de la lesión.
5. --use-NN: indica que el muestreo se realice con interpolación de vecinos cercanos.
6. --reslice-by-header: indico que, en vez de transformar geométricamente la imagen, la deje como está, pero que la muestree al espacio de la imagen de referencia.

Por tanto, para obtener la muestra de la lesión en T1 a partir de la de T2, se ha ejecutado el siguiente comando:

./WarpImageMultiTransform 3 /Users/enrique/GitHubProjects/tfg/Img\_lesion/Lesion\_T2\_PO\_1D\_W.nii.gz /Users/enrique/GitHubProjects/tfg/Img\_lesion/Lesion\_T1\_PO\_1D\_W.nii.gz -R /Users/enrique/GitHubProjects/tfg/Img\_lesion/T1\_PO.nii.gz --use-NN --reslice-by-header

Para realizar el resto de transformaciones, la sentencia será análoga. De esta forma, ya podremos utilizar también ITK-SNAP y superponer las respectivas muestras de cada lesión sobre las imágenes en T1, FLAIR y SWAN y obtener las coordenadas de la lesión en cada una de ellas.

El siguiente paso consistirá en hacer todas las modificaciones sobre las imágenes originales para cumplir con la especificación de requisitos y facilitar su visualización y comprensión en el informe clínico.

# Ajustes de visualización de imágenes NIFTI

Una vez tenemos todas las imágenes NIFTI de las resonancias magnéticas y sus correspondientes máscaras de la lesión, se procede a exportarlas a formato PNG cumpliendo con los requisitos RFX. Para ello, se utilizará la herramienta FSLEYES.

* + 1. **Instalación de FSL**

Para utilizar FSLEYES, es necesario instalar la biblioteca completa de FSL. Tal y como se describe en la Wiki de FSL[[6]](#footnote-6), “*FSL es una biblioteca de herramientas de análisis para datos de imágenes cerebrales FMRI, MRI y DTI. Funciona en Apple y en PC (tanto en Linux como en Windows a través de una máquina virtual) y es muy fácil de instalar. La mayoría de las herramientas pueden ejecutarse tanto desde la línea de comandos como en forma de interfaz gráfica de usuario*”. En este caso, se utilizará el sistema operativo macOS para la instalación y utilización de la herramienta.

Antes de instalar FSL, es necesario tener instalado xQuarz[[7]](#footnote-7) y configurar el terminal de comandos. Si se utiliza la última versión de macOS (versión 11.X), se deberá tener en cuenta la shell no utiliza ni *bash* ni *crsh*, sino que utiliza *zsh*. De modo que, para ejecutar fsl (y fsleyes) es necesario configurar, en este caso, los ficheros *.zprofile* con la configuración que tendría *.bash\_profile* según se indica en la wiki de FSL[[8]](#footnote-8).

Una vez instalado ya FSL, podemos utilizar FSLEYES. FSLEYES *“es el nuevo visor de imágenes FSL para datos 3D y 4D (sustituye a FSLView). No realiza ningún tipo de procesamiento o análisis de las imágenes - eso se hace con herramientas separadas. FSLeyes tiene muchas funciones para visualizar los datos y los resultados de diversas formas útiles*”[[9]](#footnote-9).

* + 1. **Visualización de imágenes con FSLEYES**

Del mismo modo que FSL, FSLEYES puede utilizarse bien a través de su interfaz gráfica o mediante línea de comandos. En este caso, se utilizará la línea de comandos puesto que, posteriormente, estos se ejecutarán de forma automática en un script de Python.

El objetivo de utilizar esta herramienta es cumplir con los requisitos de visualización de las distintas imágenes en el informe clínico. A continuación, se muestra el comando utilizado para ajustar y exportar a PNG la imagen de resonancia magnética en formato NIFTI en T2, junto con una explicación de los parámetros más relevantes utilizados:

fsleyes render --scene ortho --xcentre 0.00000 0.00000 --ycentre 0.00000 0.00000 --zcentre 0.00000 0.00000 --xzoom 1100.0 --yzoom 1100.0 --zzoom 1100.0 --layout horizontal -hl --performance 3 --movieSync --hideCursor --outfile /Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenT2.png --voxelLoc 288 219 42 /Users/enrique/GitHubProjects/tfg/Img\_lesion/T2\_PO.nii.gz --name "T2\_PO.nii.gz" --overlayType volume --cmap greyscale --displayRange 1.0 1000.0 --cmapResolution 256 --interpolation linear /Users/enrique/GitHubProjects/tfg/Img\_lesion/Lesion\_T2\_PO\_1D\_W.nii.gz --overlayType mask --maskColour 1.0 0.0 0.0 --outline --interpolation linear

* fsleyes render --scene ortho

Se inicia FSLEYES y se indica que se cargará una imagen en ortovisor.

* --xzoom 1100.0 --yzoom 1100.0 --zzoom 1100.0

El requisito RFX especifica las imágenes deben optimizarse al espacio disponible, eliminando el ruido y ampliando en la medida de lo posible las imágenes para facilitar su visualización. Con estos parámetros se aumentan las imágenes con respecto a sus planos axial, sagital y coronal.

* --hideCursor

Este parámetro se utiliza para eliminar el cursor, dos líneas perpendiculares en forma de cruz que señalizan un punto concreto de la imagen en su intersección, de cada uno de los planos del ortovisor.

* --outfile /Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenT2.png

Para generar el informe PDF y HTML, necesitamos imágenes en un formato habitual como el png, jpg, etc. Con este parámetro indicamos el archivo .png al que deseamos exportar el resultado final con los ajustes realizados sobre la forma de visualización.

* --voxelLoc 288 219 42

Los archivos NIFTI representan imágenes en 3D de una resonancia magnética. Para localizar dónde se ha realizado la lesión en el tálamo, es necesario conocer sus coordenadas. Este parámetro sitúa la imagen en las coordenadas especificadas.

* --displayRange 1.0 1000.0

El requisito RFX especifica que debe aumentarse el contraste de las imágenes para distinguir con más facilidad la lesión y el edema. Este parámetro permite ajustar el brillo y el contraste de la imagen.

* --overlayType mask --maskColour 1.0 0.0 0.0 –outline

El requisito RFX especifica que la lesión debe mostrarse en la imagen con el contorno destacado en color rojo. Por ello, se especifica que la capa superpuesta será de tipo máscara (*overlayType mask*), el color de la máscara será rojo (*maskColour* R G B -*1.0 0.0 0.0*-) y solo destacando el contorno (*outline*).

El resultado a la hora de visualizar la imagen es el siguiente:

* Si ejecuto el comando básico: *fsleyes render --scene ortho --outfile ImagenT2.png T2\_PO.nii.gz*

Imagen que contiene oscuro, parado, cuarto, frente

Descripción generada automáticamente

* Si ejecuto el comando con los distintos parámetros explicados anteriormente:

Un reloj de manecillas

Descripción generada automáticamente con confianza baja

Para la imagen en T1, FLAIR y SWAN el proceso será análogo, siendo los comandos utilizados los siguientes:

* Imagen T1:

*fsleyes render --scene ortho --xcentre 0.00000 0.00000 --ycentre 0.00000 0.00000 --zcentre 0.00000 0.00000 --xzoom 820.0 --yzoom 820.0 --zzoom 820.0 --layout horizontal -hl --performance 3 --movieSync --hideCursor --outfile /Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenT1.png --voxelLoc 110 112 135 /Users/enrique/GitHubProjects/tfg/Img\_lesion/T1\_PO.nii.gz --name "T2\_PO.nii.gz" --overlayType volume --cmap greyscale --displayRange 30.0 450.0 --cmapResolution 256 --interpolation linear /Users/enrique/GitHubProjects/tfg/Img\_lesion/Lesion\_T1\_PO\_1D\_W.nii.gz --overlayType mask --maskColour 1.0 0.0 0.0 --outline --interpolation linear*

* Imagen FLAIR:

*fsleyes render --scene ortho --xcentre 0.00000 0.00000 --ycentre 0.00000 0.00000 --zcentre 0.00000 0.00000 --xzoom 820.0 --yzoom 820.0 --zzoom 820.0 --layout horizontal -hl --performance 3 --movieSync --hideCursor --outfile /Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenFLAIR.png --voxelLoc 289 222 14 /Users/enrique/GitHubProjects/tfg/Img\_lesion/SWI\_PO.nii.gz --name "T2\_PO.nii.gz" --overlayType volume --cmap greyscale --displayRange 20.0 1900.0 --cmapResolution 256 --interpolation linear /Users/enrique/GitHubProjects/tfg/Img\_lesion/Lesion\_SWAN\_PO\_1D\_W.nii.gz --overlayType mask --maskColour 1.0 0.0 0.0 --outline --interpolation linear*

* Imagen SWAN:

*fsleyes render --scene ortho --xcentre 0.00000 0.00000 --ycentre 0.00000 0.00000 --zcentre 0.00000 0.00000 --xzoom 820.0 --yzoom 820.0 --zzoom 820.0 --layout horizontal -hl --performance 3 --movieSync --hideCursor --outfile /Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenSWAN.png --voxelLoc 289 235 61 /Users/enrique/GitHubProjects/tfg/Img\_lesion/SWI\_PO.nii.gz --name "T2\_PO.nii.gz" --overlayType volume --cmap greyscale --displayRange 2.0 2000.0 --cmapResolution 256 --interpolation linear /Users/enrique/GitHubProjects/tfg/Img\_lesion/Lesion\_SWAN\_PO\_1D\_W.nii.gz --overlayType mask --maskColour 1.0 0.0 0.0 --outline --interpolation linear*

Una vez obtenidas todas las imágenes en formato PNG, ya podrán incluirse en el informe clínico. Sin embargo, tal como especifica el requisito RFX, las imágenes todavía necesitan algunos ajustes para eliminar el ruido y aprovechar al máximo el folio. Estos ajustes se realizarán posteriormente utilizando algunas librerías de Python.

* + 1. **Generación de imagen mosaico**

Una imagen mosaico consiste en…

Para ello, utilizaremos en FSLEYES el parámetro *lightbox* en lugar de *ortho*, junto con otros ajustes para cumplir los requisitos similares a los explicados con anterioridad. El comando utilizado para producir una imagen mosaico a partir de la imagen en T2 es:

*fsleyes render --scene lightbox --zaxis 2 --sliceSpacing 6.219999999324791 --zrange 32.619983627262215 163.6599188131839 --ncols 4 --nrows 5 --bgColour 0.0 0.0 0.0 --fgColour 0.0 0.0 0.0 --movieSync --hideCursor --outfile /Users/enrique/GitHubProjects/tfg/Img\_lesion/mosaicoT2.png /Users/enrique/GitHubProjects/tfg/Img\_lesion/T2\_PO.nii.gz --name "T2\_PO.nii.gz" --overlayType volume --cmap greyscale --displayRange 1.0 1000.0 --cmapResolution 256 --interpolation linear /Users/enrique/GitHubProjects/tfg/Img\_lesion/Lesion\_T2\_PO\_1D\_W.nii.gz --overlayType mask --maskColour 1.0 0.0 0.0 --outline --interpolation linear*

Cabe destacar los parámetros *bgColour* y *fgColour* que indican el color del fondo. En este caso todos los valores (RGB) se encuentra en “0.0” puesto que el requisito RFX indica que el fondo de las imágenes debe ser negro.

Los parámetros *ncols* y *nrows*, indican el número de miniaturas que hay en cada columna y en cada fila respectivamente. En este caso, se ha buscado optimizar el número de elementos en cada fila y columna de forma que se aporte información detallada y se ajuste al tamaño de una cara de un folio para facilitar su visualización en el informe.

El resultado de ejecutar este comando es el siguiente:

Patrón de fondo

Descripción generada automáticamente

# Automatización de los comandos de FSLEYES

Con el apartado anterior, se entiende el procedimiento para ajustar la visualización de las imágenes nifti y para generar imágenes en formato png. En este epígrafe, se explica cómo automatizar en un script de Python los comandos obtenidos anteriormente.

El script es muy sencillo y, en este caso, se llama “*nifti\_to\_png.py*”. Tan solo hay que importar la biblioteca “*os*” y ejecutar los comandos con la función “*os.system(f*’**comando\_fsleyes***’)”*. El código utilizado se muestra a continuación:

**import os**

*# Convertimos con fsleyes las imágenes NIFTI en T1, T2, SWAN y FLAIR a PNG aplicando la máscara de la lesión.*

**os.system(f**'fsleyes render --scene ortho --xcentre 0.00000 0.00000 --ycentre 0.00000 0.00000 --zcentre 0.00000 0.00000 --xzoom 1100.0 --yzoom 1100.0 --zzoom 1100.0 --layout horizontal -hl --performance 3 --movieSync --hideCursor --outfile /Users/enrique/GitHubProjects/tfg/4Enrique/ImagenT2.png --voxelLoc 283 260 35 /Users/enrique/GitHubProjects/tfg/4Enrique/T2\_PO.nii.gz --name "T2\_PO.nii.gz" --overlayType volume --cmap greyscale --displayRange 1.0 1000.0 --cmapResolution 256 --interpolation linear /Users/enrique/GitHubProjects/tfg/4Enrique/Lesion\_T2\_PO\_1D\_W.nii.gz --overlayType mask --maskColour 1.0 0.0 0.0 --outline --interpolation linear')

*(…) Se hace el procedimiento análogo con el resto (…)*

*#Creamos una imagen en mosaico en T1, T2, FLAIR y SWAN.*

**os.system(f**'fsleyes render --scene lightbox --zaxis 2 --sliceSpacing 6.219999999324791 --zrange 32.619983627262215 163.6599188131839 --ncols 4 --nrows 5 --bgColour 0.0 0.0 0.0 --fgColour 0.0 0.0 0.0 --movieSync --hideCursor --outfile /Users/enrique/GitHubProjects/tfg/Img\_lesion/mosaicoT2.png /Users/enrique/GitHubProjects/tfg/Img\_lesion/T2\_PO.nii.gz --name "T2\_PO.nii.gz" --overlayType volume --cmap greyscale --displayRange 1.0 1000.0 --cmapResolution 256 --interpolation linear /Users/enrique/GitHubProjects/tfg/Img\_lesion/Lesion\_T2\_PO\_1D\_W.nii.gz --overlayType mask --maskColour 1.0 0.0 0.0 --outline --interpolation linear')

Este script se utilizará únicamente para transformar las imágenes NIFTI al formato PNG. Para ejecutar este script se necesita tener instalado en el ordenador toda la biblioteca de FSL. Es por ello que se ha decidido separar del script de generación del informe clínico, para disminuir las dependencias de una biblioteca tan específica y de instalación compleja. Así podrá ejecutarse el script de generación de informe de forma independiente y desde cualquier máquina, partiendo de las imágenes en formato PNG.

Sin embargo, las imágenes todavía requieren algunos ajustes como, por ejemplo, los necesarios para disminuir el ruido y hacer que se aproximen lo máximo posible a las dimensiones de un DIN-A4. Esto se explicará en el siguiente epígrafe.

# Ajuste de las dimensiones de las imágenes

Una primera transformación que haremos de las imágenes PNG antes de insertarlas en el informe clínico, será recortarlas para eliminar el ruido y para que las dimensiones de la imagen aprovechen al máximo un DIN-A4. A continuación, se muestra una imagen en T2 generada a partir de FSLEYES:

Un reloj de manecillas

Descripción generada automáticamente con confianza baja

El objetivo será eliminar las bandas superior e inferior, marcadas en la imagen anterior por rectángulos con contorno discontinuo en color rojo. Para esto, importaremos la librería PIL (*Python Imaging Library*) y el método *crop()[[10]](#footnote-10)*:

> from PIL import Image

Primero, se cargarán en distintas variables las imágenes PNG que hemos producido a partir de las imágenes NIFTI con FSLEYES:

# Opens a image in RGB mode

imT1 = Image.open("/Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenT1.png")

imT2 = Image.open("/Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenT2.png")

imSWAN = Image.open("/Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenSWAN.png")

imFLAIR = Image.open("/Users/enrique/GitHubProjects/tfg/Img\_lesion/ImagenFLAIR.png")

En segundo lugar, se deben marcar las coordenadas del área de la imagen que se quiere mantener, es decir, el área que contiene el rectángulo discontinuo verde de la Figura X. Para recortar la imagen utilizando el método *crop()*, hay que especificar dos coordenadas que se corresponden con los puntos A y B de la figura que se muestra a continuación:

Tras determinar los valores que deben tomar x1, x2, y1 e y2, se procede a utilizar el método *crop()*, indicando el nombre de la nueva imagen:

# Cropped image of above dimension

# (It will not change original image)

imT2cr = imT2.crop((x1, y1, x2, y2))

imT2cr.save("ImagenT2\_crop.png")

imT1cr = imT1.crop((x1, y1, x2, y2))

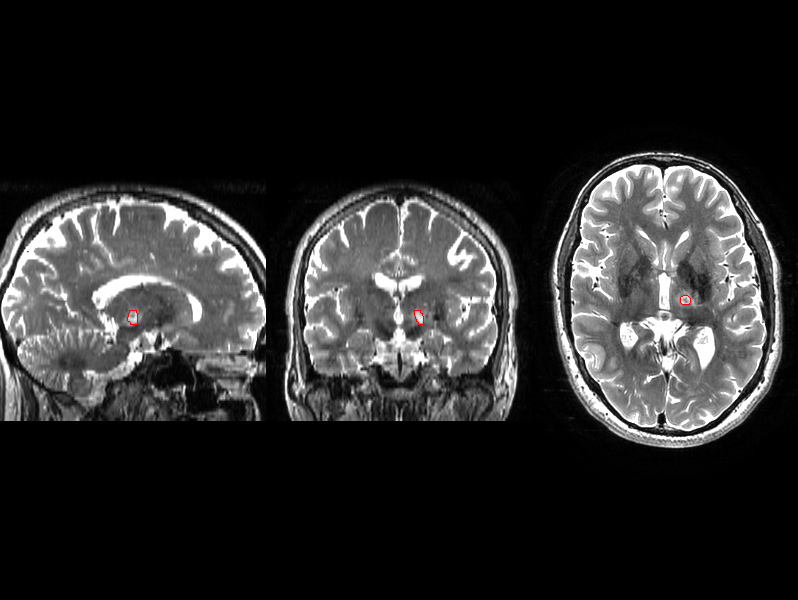
imT1cr.save("ImagenT1\_crop.png")

imSWANcr = imSWAN.crop((x1, y1, x2, y2))

imSWANcr.save("ImagenSWAN\_crop.png")

imFLAIRcr = imFLAIR.crop((x1, y1, x2, y2))

imFLAIRcr.save("ImagenFLAIR\_crop.png")



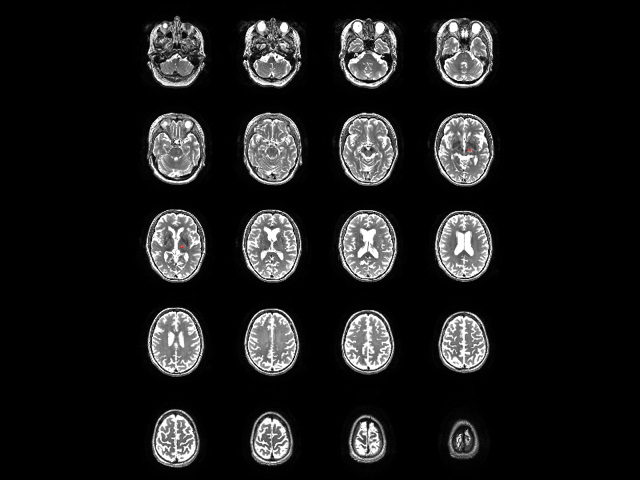
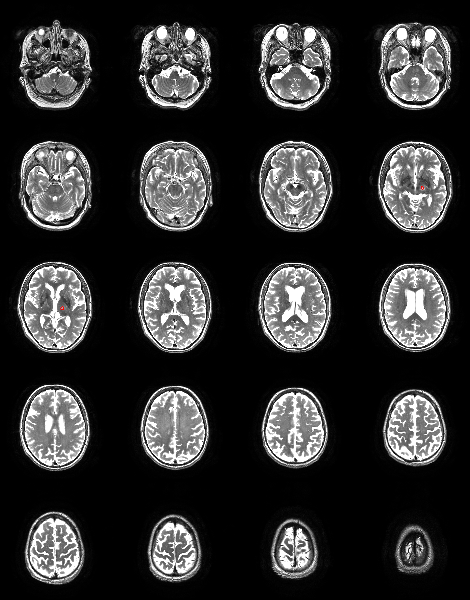
A(x1, y1)

B(x2, y2)

Imagen que contiene foto, diferente, cubierto, grupo

Descripción generada automáticamenteCon esto obtendríamos nuevas imágenes PNG de la lesión en T1, T2, FLAIR y SWAN recortadas, como se muestra en la siguiente figura:

Por último, solo quedaría hacer lo mismo con las imágenes mosaico, recalculando las coordenadas de los puntos A y B y tratando de que se ajuste a una cara de DIN-A4:



# Generación de gráficos

Además de mostrar las imágenes de resonancia magnética donde se resalta la lesión, una parte importante de este trabajo recae sobre el objetivo de comparar la lesión del paciente con una población de pacientes que han sido sometidos al mismo tratamiento. Lo interesante de esta tercera sección del informe será la comparación entre las características de la lesión (volumen, coordenadas del centroide, coordenadas del *bounding box* e intersección con el VIM) y la mejoría que muestra cada paciente. Esto permitirá al facultativo predecir la posible mejora en la sintomatología del paciente teniendo en cuenta las características de la lesión que se le ha producido en el tálamo.

Para este apartado, se ha tratado de representar la información tomando distintos tipos de gráficos como …

La información sobre la población de pacientes con la que se generarán posteriormente los gráficos se extraerá de un archivo en formato *.csv*, que contendrá las siguientes columnas:

1. Volumen: expresado en …
2. Coordenada x del centroide.
3. Coordenada y del centroide.
4. Coordenada z del centroide.
5. Diámetro del *bounding box* en el eje x.
6. Diámetro del *bounding box* en el eje y.
7. Diámetro del *bounding box* en el eje z.
8. Intersección con el VIM: expresado en porcentaje.
9. Mejoría clínica: expresada en porcentaje.



A la hora de generar los gráficos se utilizarán dos librerías de Python: pandas y matplotlib. Pandas (*Python Data Analysis Library*) es una “*herramienta de análisis y manipulación de datos de código abierto, rápida, potente, flexible y fácil de usar, construida sobre el lenguaje de programación Python*”[[11]](#footnote-11). En este caso, será la herramienta utilizada para leer los datos del fichero .csv y guardarlos en un *dataframe*. El procedimiento es el siguiente:

#*Importamos la librería pandas*import pandas as pd

#*Cargamos en un dataframe la información contenida en el csv*  
df = pd.read\_csv('caracteristicas\_mejoria.csv')

El procedimiento es así de sencillo, sin embargo, debemos tener en cuenta el formato de nuestros datos en el csv si queremos que se generen correctamente los gráficos. Por ejemplo, en el caso de los valores numéricos que contenga el fichero, no los interpretará pandas de la misma manera si se utiliza el formato anglosajón para expresar los decimales (“150.33”), que si se utiliza el formato español (“150,33”). En el primer caso, la función *read\_csv* lo tomará como un dato de tipo *float,* pero en el segundo caso lo tomará como uno de tipo *string*. Por tanto, en caso de utilizar la numeración española, utilizará valores de tipo *string* que impedirá generar los gráficos que se desean. Por ello, en este segundo caso, habrá que aplicar dos transformaciones a los valores: sustituir la “,” por un “.” y transformarlo a un dato numérico tipo *float*.

*#Transformo los datos de la columna “volumen” leídos como string en tipo float*  
df['volumen'] = df['volumen'].str.replace(',', '.').astype(float)

En el caso de este proyecto, por sencillez, se utilizarán los valores numéricos en el formato anglosajón.

1. *Neurological desorders: public health challenges*, OMS, Febrero 2007, <https://www.who.int/mental_health/neurology/neurological_disorders_report_web.pdf?ua=1> [↑](#footnote-ref-1)
2. La segmentación de la lesión, en el caso de este proyecto, se hace de forma manual con herramientas como ITK-SNAP. Para este proyecto, se han proporcionado las imágenes NIFTI en T1, T2, FLAIR y SWAN y la segmentación de la lesión en T2. El resto de lesiones se “remuestrearán” a partir de la lesión en T2 (explicacdo en otro epígrafe más adelante). [↑](#footnote-ref-2)
3. <http://stnava.github.io/ANTs/> [↑](#footnote-ref-3)
4. [*https://stackoverflow.com/questions/52531492/cmake-command-not-found-on-macos*](https://stackoverflow.com/questions/52531492/cmake-command-not-found-on-macos) [↑](#footnote-ref-4)
5. *Compiling ANTs on Linux and Mac OS*, <https://github.com/ANTsX/ANTs/wiki/Compiling-ANTs-on-Linux-and-Mac-OS> [↑](#footnote-ref-5)
6. *fslWiki,* <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki> [↑](#footnote-ref-6)
7. *FslInstalation/ShellSetup*,<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslInstallation/ShellSetup> [↑](#footnote-ref-7)
8. *Moving to ZSH, part 2: Configuration Files*, <https://scriptingosx.com/2019/06/moving-to-zsh-part-2-configuration-files/> [↑](#footnote-ref-8)
9. *Introduction -FSLeyes*, <https://fsl.fmrib.ox.ac.uk/fslcourse/lectures/practicals/intro1/index.html> [↑](#footnote-ref-9)
10. *Python PIL | Image.crop() method,* <https://www.geeksforgeeks.org/python-pil-image-crop-method/> [↑](#footnote-ref-10)
11. Definición obtenida de <https://pandas.pydata.org> [↑](#footnote-ref-11)