



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

**ALGORITMOS PARA EL
DIAGNÓSTICO DE LAS
DEFORMIDADES DE LA COLUMNA
VERTEBRAL.**

Alumno: Davinia María Garrido López

Tutor: Prof. D. Raúl Mata Campos

Depto.: Tecnologías de Telecomunicaciones

Septiembre, 2017

Índice

1. INTRODUCCIÓN	10
1.1 INTRODUCCIÓN	10
1.2 OBJETIVOS DEL PROYECTO	10
2. ANTECEDENTES	12
2.1 APLICACIONES DE VISUALIZACIÓN DE IMÁGENES MÉDICAS	12
2.1.1 RadiAnt DICOM Viewer	12
2.1.2 Navegatum	15
2.1.3 OsiriX	16
2.1.4 Dicompyler	17
2.1.5 MicroDicom	18
2.1.6 Open DICOM Viewer	20
2.1.7 Ginkgo CADx	20
2.1.8 Alma Medical Imaging (Alma MPR)	21
2.2 LA COLUMNA VERTEBRAL	23
2.3 LESIONES Y ENFERMEDADES DE LA COLUMNA VERTEBRAL	25
2.3.1 Escoliosis	25
3. ANÁLISIS	28
3.1 LA IMAGEN DIGITAL	28
3.1.1 CARACTERÍSTICAS DE LA IMAGEN	28
3.2 IMÁGENES EN MATLAB	29
3.2.1 TIPOS DE IMÁGENES SEGÚN EL MODO DE ALMACENAMIENTO	29
3.2.1.1 Imágenes indexadas	30
3.2.1.2 Imágenes de intensidad	31

3.2.1.3	Imágenes binarias	31
3.2.1.4	Imágenes RGB	32
3.2.2	FUNCIONES BÁSICAS DE TRATAMIENTO DE IMÁGENES	32
3.2.2.1	Lectura de la imagen	32
3.2.2.2	Escritura de la imagen	33
3.2.2.3	Representación de la imagen	34
3.2.2.4	Determinar el tamaño de la imagen	35
3.2.2.5	Selección de una zona de la imagen	35
3.2.2.6	Cambio de tamaño de la imagen	35
3.2.2.7	Conversión de tipo de imágenes	35
3.3	FORMATOS DE IMAGEN	36
3.3.1	BMP (Windows Bitmap).....	36
3.3.2	GIF (Graphics Interchange Format).....	36
3.3.3	JPEG (Joint Photographic Experts Group)	37
3.3.4	TIFF (Tagged Image File Format)	37
3.3.5	PNG (Portable Network Graphics).....	37
3.3.6	PSD	37
3.3.7	RAW.....	38
3.4	FORMATO DICOM	38
3.4.1	MODALIDADES DE IMÁGENES MÉDICAS.....	38
3.4.1.1	Radiografías obtenidas con rayos X	38
3.4.1.2	Resonancia magnética	39
3.4.1.3	Ultrasonidos	39
3.4.1.4	Medicina nuclear	40
3.4.2	IMAGEN DICOM	41
3.4.2.1	¿Qué es DICOM?	41

3.4.2.2 ¿Qué contiene DICOM?	41
3.4.2.3 Historia	42
3.4.2.4 Aplicaciones	42
4. DISEÑO	46
4.1 OPCIONES GENERALES	46
4.1.1 CARGA Y VISUALIZACIÓN DE IMÁGENES.....	46
4.1.1.1 Utilidad	46
4.1.1.2 Implementación	46
4.1.1.3 Funcionamiento.....	47
4.1.2 CAMBIO DE FORMATO Y ALMACENAMIENTO	51
4.1.2.1 Utilidad	51
4.1.2.2 Implementación	51
4.1.2.3 Funcionamiento.....	51
4.1.3 SALIDA DE LA INTERFAZ.....	53
4.1.3.1 Utilidad	53
4.1.3.2 Implementación	53
4.2 OPCIONES DE EDICIÓN BÁSICAS	54
4.2.1 RECORTE	54
4.2.1.1 Utilidad	54
4.2.1.2 Implementación	54
4.2.1.3 Funcionamiento.....	54
4.2.2 ZOOM	57
4.2.2.1 Utilidad	57
4.2.2.2 Implementación	57
4.2.2.3 Funcionamiento.....	57
4.2.3 PAN	58

4.2.3.1 Utilidad	58
4.2.3.2 Implementación	58
4.2.3.3 Funcionamiento	58
4.2.4 TRANSFORMACIONES GEOMÉTRICAS DE LA IMAGEN	59
4.2.4.1 Utilidad	59
4.2.4.2 Fundamento teórico.....	59
4.2.4.3 Implementación	60
4.2.4.4 Funcionamiento	62
4.3 OPCIONES DE PRE PROCESADO	63
4.3.1 RUIDO	63
4.3.1.1 Utilidad	64
4.3.1.2 Fundamentos teóricos	64
4.3.1.3 Implementación	66
4.3.1.4 Funcionamiento	66
4.3.2 HISTOGRAMA	70
4.3.2.1 Utilidad	70
4.3.2.2 Implementación	71
4.3.2.3 Funcionamiento	71
4.3.2 AJUSTES DE CONTRASTE DE LA IMAGEN	72
4.3.3.1 Utilidad	72
4.3.3.1 Implementación	72
4.3.3.1 Funcionamiento	73
4.4 OPCIONES DE MEJORA DE LA IMAGEN	74
4.4.1 ROI (Region Of Interest).....	76
4.4.1.1 Utilidad	77
4.4.1.2 Implementación	77

4.4.1.3 Funcionamiento.....	77
4.4.2 BINARIZACIÓN.....	79
4.4.2.1 Utilidad	79
4.4.2.2 Implementación	79
4.4.2.3 Funcionamiento.....	80
4.4.3 BORDES.....	80
4.4.3.1 Utilidad	80
4.4.3.2 Implementación	81
4.4.3.3 Funcionamiento.....	83
4.5 OPCIONES DE MEDICIÓN	85
4.5.1 CURSOR	85
4.5.1.1 Utilidad	85
4.5.1.3 Funcionamiento.....	85
4.5.2 ESCOLIOSIS	86
4.5.2.1 Utilidad	86
4.5.2.2 Implementación	88
4.5.2.3 Funcionamiento.....	90
4.5.4 DATOS DE PACIENTES.....	92
4.5.4.1 Utilidad	92
4.5.4.2 Implementación	92
4.5.4.3 Funcionamiento.....	93
5. CONCLUSIONES	96
6. LÍNEAS DE FUTURO	98
7. BIBLIOGRAFÍA.....	99

Índice de figuras

Ilustración 1. Imagen tomada de la aplicación RadiAnt DICOM Viewer	12
Ilustración 2. Diferencias entre RadiAnt DICOM Viewer y RadiAnt DICOM Viewer CD/DVD	14
Ilustración 3. Imagen tomada de la aplicación Navegantium.....	15
Ilustración 4. Imagen tomada de la aplicación OsiriX.....	17
Ilustración 5. Imagen tomada de la aplicación Dicompyler	18
Ilustración 6. Imagen tomada de la aplicación MicroDicom	19
Ilustración 7. Imagen tomada de la aplicación Open DICOM Viewer	20
Ilustración 8. Imagen tomada de la aplicación Ginkgo CADx	21
Ilustración 9. Imagen tomada de la aplicación Alma Medical Imaging	22
Ilustración 10. Imagen detallada de la columna vertebral.....	23
Ilustración 11. Ejemplo de uso del escoliómetro	26
Ilustración 12. Cálculo del ángulo de Cobb	27
Ilustración 13. Imagen con diferentes resoluciones.....	29
Ilustración 14. Ejemplo de imagen indexada.....	30
Ilustración 15. Ejemplo de imagen de intensidad	31
Ilustración 16. Ejemplo de imagen binaria.....	31
Ilustración 17. Ejemplo de imagen RGB.....	32
Ilustración 18. Ejemplo de imagen médica obtenida con ultrasonidos.....	40
Ilustración 19. Ejemplo de imagen médica obtenida con medicina nuclear	41
Ilustración 20. Imagen de apertura de archivo	47
Ilustración 21. Interfaz de selección de archivos	48
Ilustración 22. Visualización de la plataforma	49
Ilustración 23. Mensaje que aparece al llegar a la última foto	50
Ilustración 24. Mensaje que aparece si llegamos a la primera foto	50

Ilustración 25. Menu de guardado de imágenes.....	52
Ilustración 26. Ventana para poder guardar la imagen.....	52
Ilustración 27. Ruta de menú para salir de la plataforma	53
Ilustración 28. Navegación en el menú para llegar a la función de recorte.....	55
Ilustración 29. Selección de la zona a recortar.....	56
Ilustración 30. Imagen recortada.....	56
Ilustración 31. Recorrido del menú para llegar a la opción de zoom	57
Ilustración 32. Recorrido en el menú para llegar a la función PAN.....	58
Ilustración 33. Resultado de aplicar la función	59
Ilustración 34. Ejemplo de transformación de inclinación en la dirección x.....	61
Ilustración 35. Giro a la derecha de la imagen	62
Ilustración 36. Ejemplo de imagen con ruido sal y pimienta	64
Ilustración 37. Imagen, transformada periódica y un periodo de la misma centrado en bajas frecuencias.	65
Ilustración 38. Recorrido en el menú para llegar al filtro para eliminar el ruido.....	69
Ilustración 39. Resultado después de aplicar el filtro contra el ruido Gaussiano	69
Ilustración 40. Histograma original y expandido de una imagen.....	70
Ilustración 41. Resultado después de expandir el histograma.....	71
Ilustración 42. Resultado en la plataforma después de calcular el histograma.....	72
Ilustración 43. Funcionamiento del ajuste de contraste de la imagen.....	73
Ilustración 44. Imagen previa al ajuste de contraste	73
Ilustración 45. Imagen después de realizar el contraste.....	74
Ilustración 46. Tipos de bordes	75
Ilustración 47. Uso de las derivadas para detectar bordes	76
Ilustración 48. Navegación por menú para aplicar la ROI	78
Ilustración 49. Selección de la ROI	78

Ilustración 50. ROI de la imagen.....	79
Ilustración 51. Imagen binarizada	80
Ilustración 52. Resultado de aplicar el detector de bordes de Canny	82
Ilustración 53. Diferentes resultados al aplicar los algoritmos de detección de bordes.....	84
Ilustración 54. Elección del algoritmo de Canny para la detección de bordes	84
Ilustración 55. Navegación por el menú para encontrar la opción de cursor	85
Ilustración 56. Hacemos click en el punto de la imagen donde nos interese tener el dato	86
Ilustración 57. Medición del ángulo de Cobb.....	87
Ilustración 58. Menú para realizar las medidas de escoliosis	90
Ilustración 59. Dibujo de las líneas para el cálculo del ángulo de Cobb	90
Ilustración 60. Dibujo de las líneas para el cálculo del ángulo de Cobb	91
Ilustración 61. Representación del ángulo de Cobb	91
Ilustración 62. Anotaciones sobre el grado de escoliosis del paciente	92
Ilustración 63. Interfaz para la visualización de los datos del paciente	94
Ilustración 64. Introducción del nombre del paciente	94
Ilustración 65. Datos del paciente ya introducidos	95
Ilustración 66. Guardar imagen después de añadir los datos.....	95

1. INTRODUCCIÓN

1.1 INTRODUCCIÓN

Gracias a los avances tecnológicos aplicados a la medicina, surge una disciplina llamada bioingeniería [1] que, a partir de herramientas, métodos y principio de la ingeniería, analiza aspectos relacionados con la biología. La bioingeniería está centrada en el estudio de los problemas de los organismos vivos.

En este proyecto, trabajaremos con la rama de la ingeniería biomédica (orientada a la medicina). Por los rápidos avances en este campo en los últimos años, se pueden obtener imágenes de cualquier parte del cuerpo humano con radiaciones prácticamente inofensivas. Esta rama ha permitido la obtención de diagnósticos médicos de forma más rápida y precisa que con los métodos tradicionales.

Utilizaremos técnicas de procesado digital basándonos en imágenes médicas obtenidas mediante rayos X o resonancias magnéticas con el fin de efectuar diagnósticos. Estas técnicas nos permitirán, además, extraer información del paciente, resaltar algunas características de la imagen o mejorar la visualización de ésta. Esto facilita mucho el trabajo de los especialistas, ya que antes de tener estas imágenes en digital, la única solución a problemas de visualización era repetir la prueba, lo cual aumentaba los costes. [2]

Para hacer todo esto usaremos Matlab, una de las herramientas más potentes para el tratamiento de imágenes gracias a su toolbox y a la facilidad de programación de algoritmos desde cero que ofrece.

1.2 OBJETIVOS DEL PROYECTO

El objetivo principal de este proyecto es el diseño e implementación de una herramienta básica de visualización y anotación de imágenes médicas en el formato DICOM.

Dada la amplia variedad de tipos de imágenes médicas, así como sus diferentes características, este trabajo se centra en imágenes cervicales, sobre las que se desarrollará la herramienta. Esto permitirá definir y desarrollar técnicas de procesamiento digital de imágenes adaptadas a la tipología de las mismas. [3]

Se diseñarán algoritmos que permitan realizar la lectura, visualización, procesamiento básico de las mismas (para facilitar su interpretación y diagnóstico), anotaciones y escritura de dichas imágenes en el formato DICOM. Se realizará también un algoritmo que permita determinar el grado de desviación de la columna vertebral (ángulo de Cobb) y determine también el diagnóstico para evitar tener que hacer las mediciones a mano. Además de un estudio comparativo de los algoritmos que se utilicen. Las anotaciones no deberán modificar los datos DICOM.

Se desarrollará una aplicación con entorno gráfico para la integración de los algoritmos implementados de forma que proporcione un interfaz cómodo, intuitivo y sencillo de utilizar para facilitar el diagnóstico a partir de las imágenes médicas analizadas.

2. ANTECEDENTES

Ya que en este proyecto se va a realizar un sistema de visualización de imágenes en formato DICOM, es necesario empezar hablando sobre las distintas aplicaciones que hay ya en el mercado y sus funcionalidades.

2.1 APLICACIONES DE VISUALIZACIÓN DE IMÁGENES MÉDICAS

2.1.1 RadiAnt DICOM Viewer

Es el software más recomendado para usuarios sin experiencia en el procesamiento de imágenes ya que es muy intuitivo. Sin embargo, tiene un gran abanico de funcionalidades y soporta todas las modalidades de archivos: TC, RMN, US, MN, Rx y muchas otras. Está diseñado para utilizar los recursos consumiendo lo menos posible y a gran velocidad. Se puede abrir una sola imagen o varias, en la misma ventana o en otras ventanas para compararlas entre ellas. [4]



Ilustración 1. Imagen tomada de la aplicación RadiAnt DICOM Viewer

Entre sus características están:

- Desplazamiento y zoom fluidos
- Ajustes de brillo y contraste, modo negativo
- Configuración predeterminada de ventana para Tomografías Computadas (pulmones, huesos, etc.)
- Posibilidad de rotar (90, 180 grados) o girar las imágenes (horizontal y vertical)
- Longitud de segmento
- Valores de ángulo (ángulo normal y Cobb)
- Herramienta de lápiz para dibujar libremente
- Permite exportar los archivos DICOM a otros formatos de imagen o películas.

Este producto se encuentra en dos modalidades: RadiAnt DICOM Viewer y RadiAnt DICOM Viewer CD/DVD. Ambos tienen la opción de descarga de una versión de prueba por un tiempo limitado, pero con funcionalidad completa. No hay inmensas diferencias entre ambos, pero hay algunas que debemos destacar:

- RadiAnt DICOM Viewer CD/DVD no permite la apertura de imágenes médicas desde USB, desde un archivo ZIP o de ubicaciones PACS.
- RadiAnt DICOM Viewer CD/DVD tampoco permite trabajar con 3D ni a integración de sistemas de terceros por ventana de comandos.
- El precio de RadiAnt DICOM Viewer CD/DVD es de 399€ por licencia y el de RadiAnt DICOM Viewer es de 99€.

	RadiAnt DICOM Viewer	RadiAnt DICOM Viewer CD/DVD
Basic facts		
Desktop application for installation on PCs, laptops and tablets running Windows systems	✓	
Autorun package for distribution on optical discs as a patient CD/DVD viewer running without installation ¹		✓
Windows XP/Vista/7/8/8.1/10 supported	✓	✓
No additional dependencies (.NET, Java, etc.)	✓	✓
Very compact size ²	✓	✓
Excellent performance with 32 and 64-bit versions optimized for multi-core processors	✓	✓
Open DICOM studies		
Open DICOM studies from CD/DVD/Blu-ray discs	✓	✓ ³
Open DICOM studies from local and network folders	✓	
Open DICOM studies from USB drives	✓	
Open ZIP archives with DICOM files	✓	
Search and download DICOM studies from PACS locations	✓	
Advanced tools		
2D MPR (orthogonal multiplanar reconstructions)	✓	✓
PET-CT image fusion	✓	✓
Time-intensity curves	✓	✓
3D MPR (oblique multiplanar reconstruction) with MIP/MinIP/Avg modes	✓	
3D VR (volume rendering)	✓	
Interface		
Full-screen mode	✓	✓
Multi-touch support for Windows 8/8.1/10 touch-enabled devices	✓	✓
Multilingual interface - more than 30 translations available ⁴	✓	✓
Customizable keyboard shortcuts	✓	✓
Integration with third-party systems via command line parameters	✓	

Ilustración 2. Diferencias entre RadiAnt DICOM Viewer y RadiAnt DICOM Viewer CD/DVD

2.1.2 Navegantium

Navegantium es un software exclusivo de Windows y orientado a su uso con tablets o dispositivos táctiles. Es sencillo de usar y gestiona imágenes DICOM desde unidades extraíbles, la unidad local o un servidor PACS.

Es capaz de reconstruir planos múltiples (MPR), con volumen (3D) y de rayos X, y realizar un análisis profundo de imágenes médicas.

Además, se pueden hacer anotaciones y medidas directamente en las imágenes y generar informes de forma sencilla. [5]



Ilustración 3. Imagen tomada de la aplicación Navegantium

Entre sus características están:

- Compatible con las siguientes modalidades DICOM: CT, MR, PT, CR, DX, RF, MG, XA y NM.
- Importar/exportar estudios DICOM a/desde carpetas HDD/SSD locales, unidad USB, CD/DVD.
- Compatibilidad con unidades en la nube.

- Organizador DICOM (paciente/serie/imágenes).
- Una interfaz de ordenación muy sencilla para un rápido acceso.
- Potente motor de búsqueda que permite buscar imágenes PACS y locales por nombre, fecha o modalidad.
- Reconstrucción de planos múltiples (MPR) de gran calidad en los tres planos anatómicos principales (axial, sagital, coronal), reconstrucción con volumen (3D), proyección de rayos X y pantalla de imágenes de adquisición originales.
- Herramientas de manipulación: desplazar, zoom, girar en 2D/3D, ventanas dinámicas, preselecciones de ventanas, paletas de colores, POI, líneas de referencia, sincronizar.
- Paletas de colores para aplicar a los estudios.

Se puede conseguir gratis en https://www.microsoft.com/es-es/store/p/navegatum-dicom-viewer/9wzdncrdhzbzq?cid=site_home_es&rtc=1

2.1.3 OsiriX

Se trata del visualizador DICOM para MacOs y iOs por excelencia. Es capaz de trabajar con imágenes producidas por cualquier PACS o modalidad de imagen (MRI, CT, PET, PET-CT, CR, DX ultrasonidos, etc.).

Fue diseñado para la navegación y visualización de imágenes multimodalidad y multidimensionales.

Tiene tanto una versión para escritorio como una versión móvil.

El precio de esta aplicación es de 745€ por licencia (la cual sólo puede ser instalada en un ordenador y permite un año de actualizaciones gratuitas). A partir del primer año, la renovación de la licencia cuesta 106€. El manual de uso también costaría 106€. [6]



Ilustración 4. Imagen tomada de la aplicación OsiriX

Entre sus características están:

- Reconstrucción multiplanar (MPR)
- Representación de la superficie
- Reconstrucción volumétrica y proyección de intensidad máxima (MIP).
- Fusión de imágenes entre dos series diferentes (PET-CT y SPECT-CT soporte de pantalla).
- Creación y lectura de contenidos en CD/DVD DICOM, así como importación de archivos a diferentes formatos de imágenes.

2.1.4 Dicompyler

Aunque puede trabajar con todo tipo de imágenes DICOM, sus herramientas están orientadas para procesamiento de imágenes destinadas a la investigación en radioterapia. Está disponible para Linux, Mac y Windows. Es gratuito y de código abierto. [7]

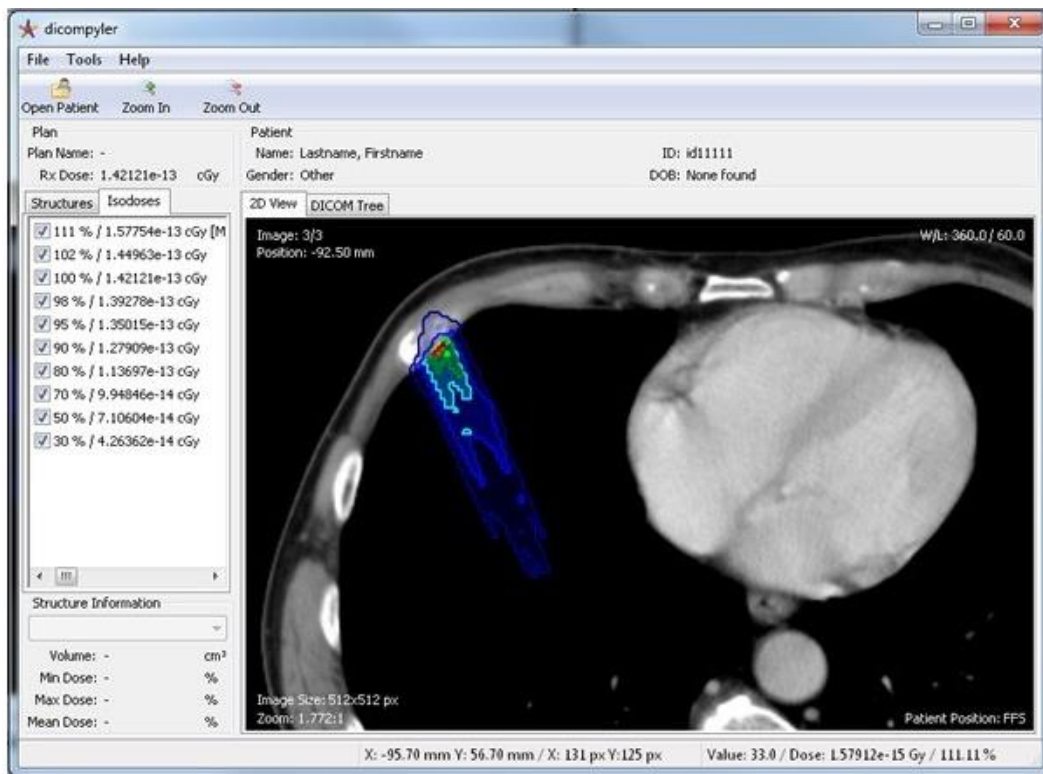


Ilustración 5. Imagen tomada de la aplicación Dicompyler

2.1.5 MicroDicom

MicroDicom es un visualizador DICOM para Windows que permite el procesado de imágenes en el campo de la medicina con una interfaz muy sencilla. Se pueden consultar listas de pacientes, realizar mediciones sobre la imagen y anotaciones.

La licencia para un solo equipo con un año de actualizaciones y soporte cuesta \$69.99 en su versión normal, y \$349.99 en su versión CD/DVD. [8]

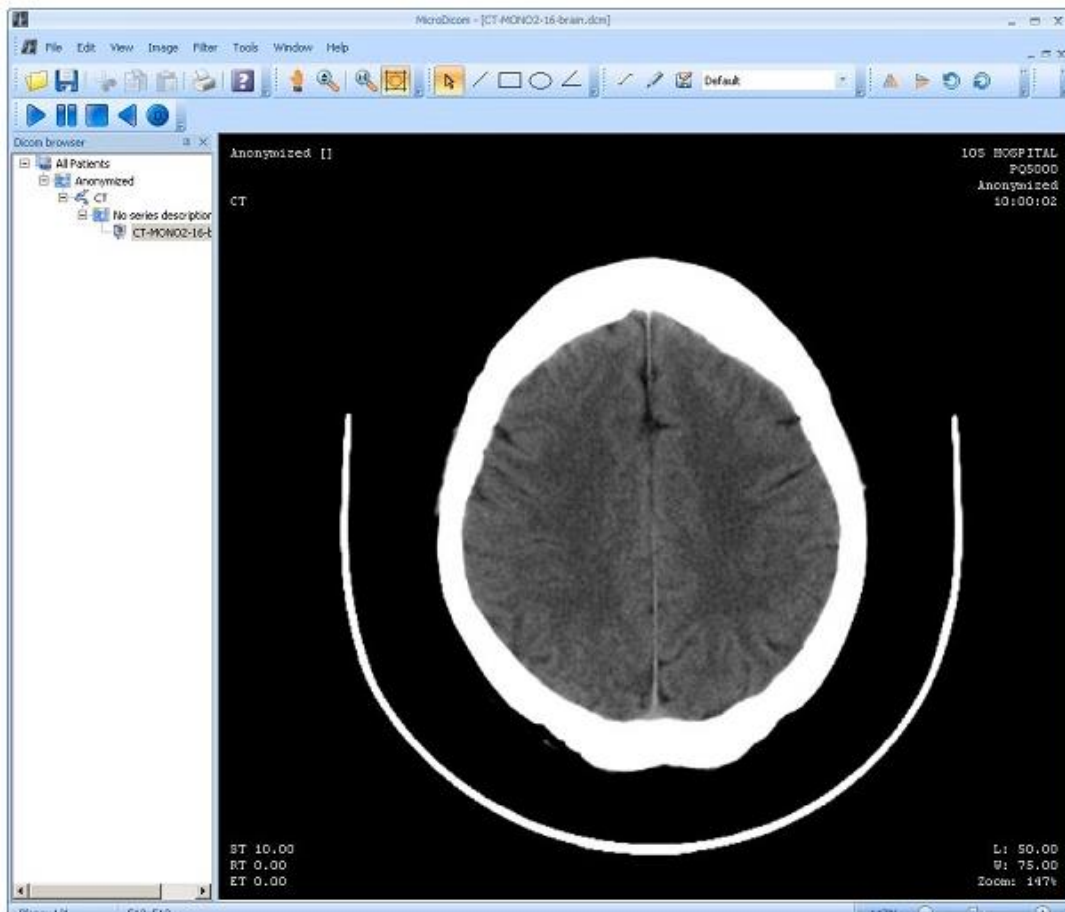


Ilustración 6. Imagen tomada de la aplicación MicroDicom

Entre sus características están:

- Permite visualizar tanto archivos DICOM como JPG, BMP, GIF, PNG o TIFF. Las imágenes se pueden cargar haciendo doble click o incluso arrastrando.
- Permite generar archivos de video en formato AVI o WMV con series de imágenes organizadas.
- Abre, crea y guarda imágenes médicas en formato DICOM.
- Permite analizar las imágenes hasta el último detalle gracias a la función de zoom.
- Tiene un editor básico para redimensionar, rotar e invertir imágenes.
- Control del brillo/contraste.

2.1.6 Open DICOM Viewer

Se trata de un visualizador de código abierto, gratuito y de alto rendimiento compatible con Windows, Linux y MacOS. Este alto rendimiento se consigue gracias a lo simple que es. [9]

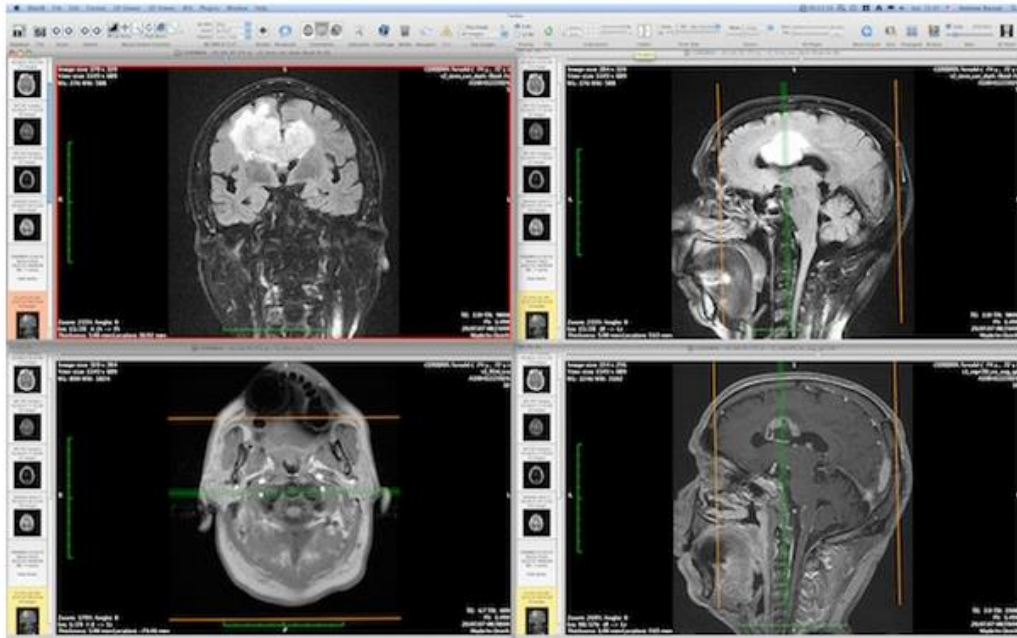


Ilustración 7. Imagen tomada de la aplicación Open DICOM Viewer

2.1.7 Ginkgo CADx

Ginkgo CADx es un visualizador multiplataforma gratuito que permite convertir otros formatos de imágenes a DICOM. Posee una interfaz intuitiva y personalizable. Soporta un gran número de modalidades de imágenes como CT, MR, ECG, PET, XC, SC, XA, MG... [10]

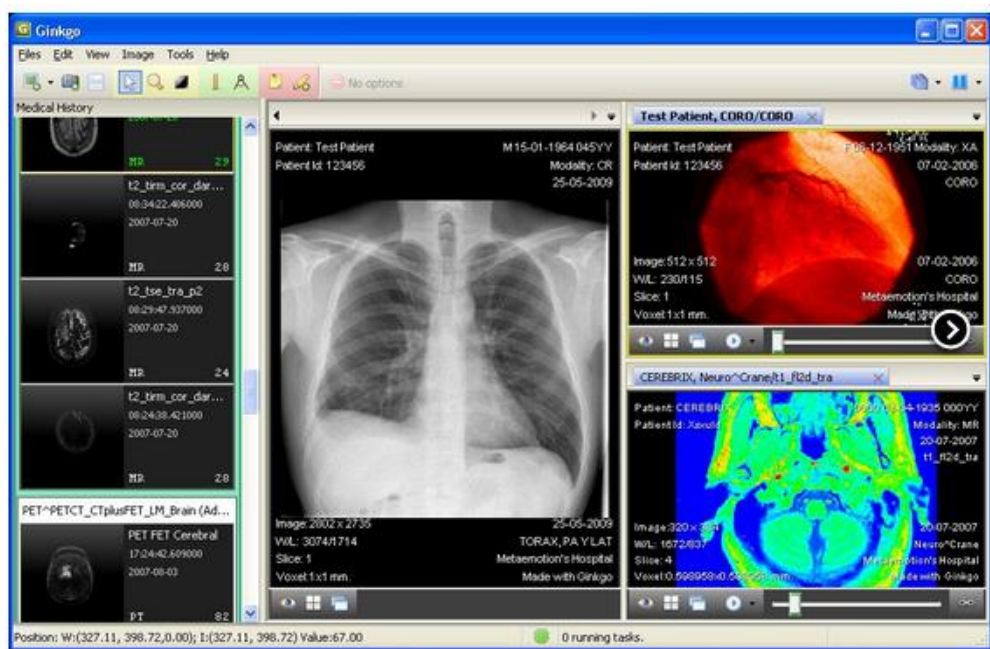


Ilustración 8. Imagen tomada de la aplicación Ginkgo CADx

2.1.8 Alma Medical Imaging (Alma MPR)

Este programa permite que, contando con una serie de imágenes de entrada, se construya un modelo tridimensional y permite visualizar de forma simultánea las proyecciones en distintas vistas (axial, coronal, sagital y oblicua) interactivamente.

El software posibilita al usuario cambiar el plano de sección y adquirir la imagen de sección correspondiente en tiempo real.

Este software se puede conseguir por un pago mensual de 40€ al mes, o mediante un pago único de 950€ más otro pago de 150€ anuales para mantenimiento. [11]

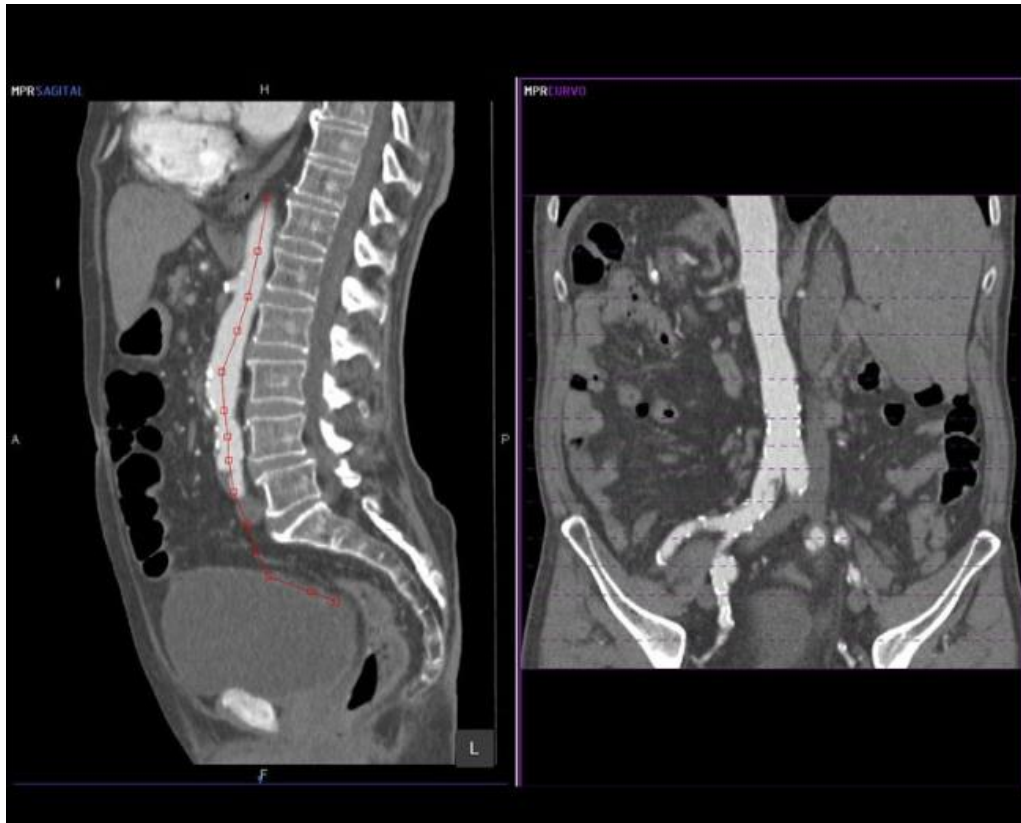


Ilustración 9. Imagen tomada de la aplicación Alma Medical Imaging

Entre sus características están:

- Manipulación de imágenes 2D: control de la ventana de visualización (W/L interactivo, ventanas predefinidas, negativo, etc.), zoom y traslación, rotación de la imagen (+90, -90) y espejo horizontal y vertical.
- Generación de nuevas series de imágenes a partir de las proyecciones de las diferentes vistas.
- Anotaciones: medidas (distancias, ángulos, diámetros, áreas, etc.), comentarios y flechas, regiones de interés (ROI).
- Librería de paletas
- Exportación de imagen 2D en formato JPG, TIF o BMP.

2.2 LA COLUMNA VERTEBRAL

Como este trabajo se centra en imágenes médicas de la columna vertebral, se va a hacer una pequeña introducción sobre la anatomía de esta parte del cuerpo.

La columna vertebral (o espina dorsal) se compone de las vértebras, los discos y la médula espinal. Es un canal de comunicación para el cerebro, de las señales que se envían y se reciben a través de la médula espinal. [12]

La columna se divide en 5 áreas funcionales:

1. Cervical: 7 vértebras (C 1-7)
2. Torácica: 12 vértebras (T 1 – 12)
3. Lumbar: 5 vértebras (L 1-5)
4. Sacra: 5 vértebras
5. Coxígea: 4-5 vértebras

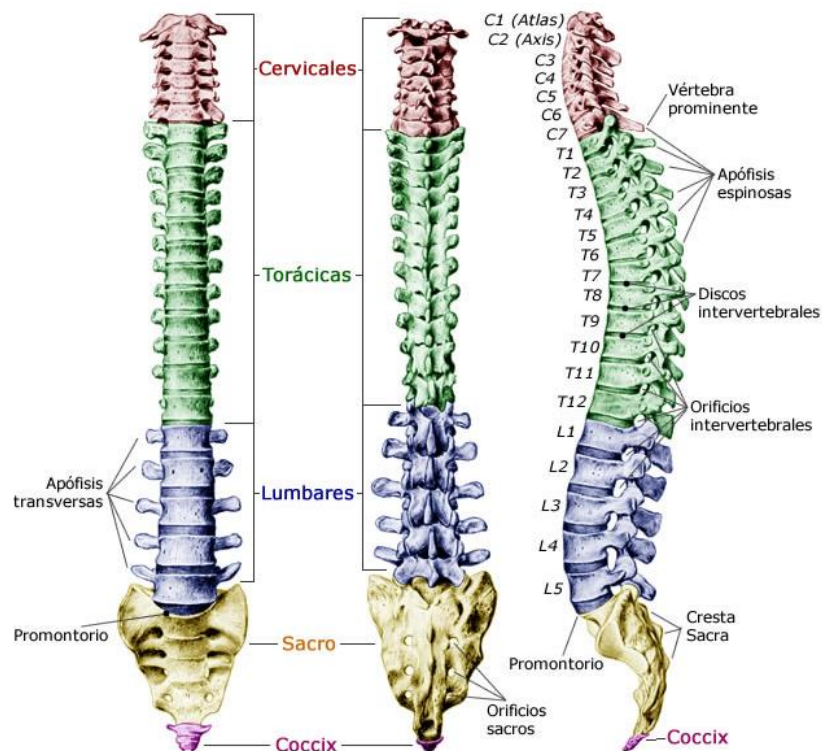


Ilustración 10. Imagen detallada de la columna vertebral

Médula espinal

La médula espinal es un conjunto de los nervios y fibras unidas que se extiende desde el cerebro hasta la parte baja de la espalda. La médula espinal está protegida por una especie de conducto de hueso formado por vértebras que están separadas por discos. Las señales eléctricas mandadas por el cerebro a través de la médula espinal, dan instrucciones a las distintas áreas del cuerpo.

Discos

Cada vértebra está separada por una sustancia llamada disco, que actúa como un cojín y un sello al mismo tiempo. En su parte exterior son resistentes, en el interior poseen un líquido más suave. Cada vez nos movemos, cambian de forma evitando así que se rocen las vértebras con el movimiento.

Anatomía ósea

- Columna cervical

Hay siete vértebras cervicales. Los huesos cervicales sirven para flexionar, extender, y girar la cabeza.

Cada vértebra cervical se compone de dos partes, un cuerpo y un arco de protección para la médula espinal (el arco neural).

- Columna torácica

La columna torácica se junta con las costillas en la zona del pecho. Hay 12 vértebras en esta región.

El canal espinal en la región es bastante más pequeño que las áreas cervicales o lumbares. Esto provoca que la médula espinal corra un riesgo mayor en la zona torácica en caso de fractura.

El movimiento que se produce en la columna torácica es principalmente rotación. Las nervaduras impiden la flexión hacia el lado.

- Columna lumbosacra

Las cinco vértebras lumbares son grandes, anchas y gruesas. La vértebra lumbar más baja, L5, se articula con el hueso sacro, y el sacro se une a la pelvis.

Permite movimientos tanto adelante y hacia atrás como hacia los lados.

2.3 LESIONES Y ENFERMEDADES DE LA COLUMNA VERTEBRAL

Dado que este trabajo está centrado en imágenes de columna, se van a trabajar los diagnósticos de las enfermedades para este tipo de imágenes.

2.3.1 Escoliosis

La escoliosis no está considerada una enfermedad, es un término que se usa para describir cualquier curvatura anómala de la columna vertebral. Si miramos la parte de atrás del cuerpo, una columna vertebral normal es recta. En el caso de que haya escoliosis, la curva de la columna vertebral puede variar de estas tres formas: [13]

- Levoescoliosis: una sola curva hacia la izquierda (en forma de letra C) de la columna.
- Dextroescoliosis: una sola curva a la derecha (con forma de letra C invertida) de la columna.
- Dos curvas (en forma de la letra S) en la columna.

La escoliosis se clasifica en:

- Congénita: causada por anomalías vertebrales al nacer. Se calcula que hay un 15% de estos casos.
- Idiopática: de causa desconocida. Se calcula que hay un 65% de estos casos.
- Neuromuscular: se desarrolla como síntoma secundario de otra enfermedad (espinas bífida, parálisis cerebral...). Se calcula que hay un 10% de estos casos.

Diagnóstico

Para el diagnóstico se empieza haciendo la prueba de flexión hacia delante de Adams. Esta prueba se realiza muy a menudo a estudiantes en edad escolar. Esta prueba consiste en que el paciente sin camisa se incline hacia el frente, si se nota una prominencia, se considera que hay riesgo de padecer escoliosis y se manda al paciente a hacerse una radiografía para confirmar el diagnóstico.

También se puede utilizar un escoliómetro para diagnosticar la escoliosis.



Ilustración 11. Ejemplo de uso del escoliómetro

El método usado para medir de forma cuantitativa la curvatura de la columna es la medida del ángulo de Cobb. Éste es el ángulo que se forma entre dos líneas perpendiculares a la parte de arriba de la vértebra superior involucrada y la parte de debajo de la vértebra inferior.

Para los pacientes con dos curvas, se calculan los ángulos para ambas.

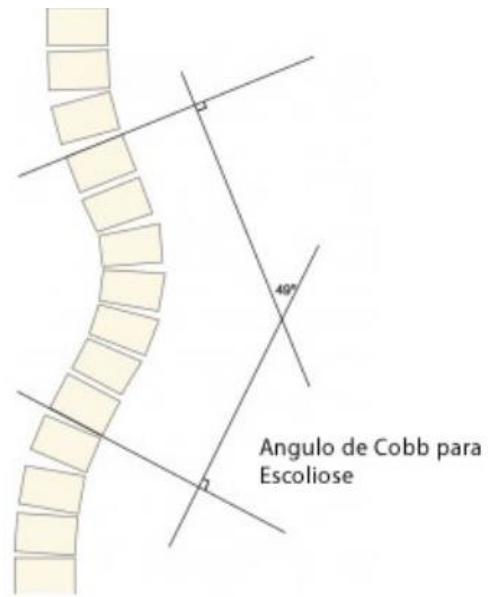


Ilustración 12. Cálculo del ángulo de Cobb

3. ANÁLISIS

No podemos hablar de procesamiento de imágenes sin antes aclarar el concepto de imagen digital y cómo se trabaja con este tipo de imágenes en Matlab (en especial con el formato DICOM).

3.1 LA IMAGEN DIGITAL

La imagen se conoce como la representación visual de un objeto. Puede representarse a través de la fotografía, el video o la pintura.

Una imagen digital es una representación en dos dimensiones de una imagen, construida a partir de una matriz numérica de tamaño $m \times n$, frecuentemente binaria (0 y 1), en la que cada uno de los puntos de intensidad se le denomina píxel. [14] [15]

El píxel puede definirse como la unidad más pequeña homogénea en color de la que está compuesta una imagen digital. Cada píxel tiene bits de información. Si un píxel tiene una profundidad de 8, puede producir 2^8 , o lo que es lo mismo, 256 tonos de grises por cada uno de los píxeles. La cantidad de píxeles que tenga una imagen y el tamaño de los mismos va a determinar su resolución espacial. [16]

El tamaño de la imagen es el producto de los píxeles de ancho por los de alto. Hay que tener en cuenta este factor, ya que, al trabajar con algunas herramientas, se tendrá en cuenta esta característica y repercutirá en la eficiencia de la misma debido a que cuanto mayor sea la imagen, más tiempo invierte el algoritmo en procesarla.

3.1.1 CARACTERÍSTICAS DE LA IMAGEN

Resolución espacial: La resolución espacial es la habilidad que tiene un sistema de imágenes de mostrar pequeños detalles de un objeto. En radiología convencional, la resolución de una imagen se limita a 10 pares de líneas por milímetro (lp/mm). En receptores digitales, la resolución oscila entre 2.5 lp/mm y 10 lp/mm. Esto indica que hay menos detalle en radiología digital que en la convencional. Lo cual no quiere decir que la imagen digital sea peor que la convencional. [18]

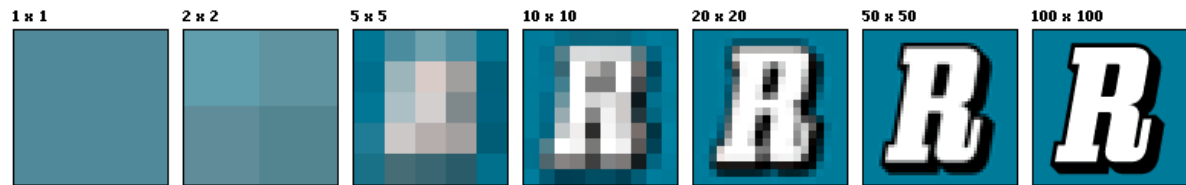


Ilustración 13. Imagen con diferentes resoluciones

Resolución de contraste: es la capacidad de un sistema digital de mostrar cambios pequeños en la escala de grises. Una resolución de contraste alta denota que las diferencias entre densidades adyacentes van a ser aumentadas, se mostrarán más niveles de grises y se podrán apreciar mejor los cambios.

Brillo: el brillo se encarga de alterar también la gama de los tonos de la imagen. Modificar el brillo de una imagen provoca una reducción del contraste y la pérdida de detalle.

3.2 IMÁGENES EN MATLAB

A continuación, vamos a ver cómo trabaja Matlab con las imágenes. Esto es necesario para poder entender cómo se implementan y se desarrollan los algoritmos que vamos a utilizar para el desarrollo de la aplicación.

3.2.1 TIPOS DE IMÁGENES SEGÚN EL MODO DE ALMACENAMIENTO.

Antes de entrar en detalle, es necesario explicar cómo se dividen las imágenes según el tipo de elementos que las componen: [19]

- Double: valores de doble precisión en el rango de [-10308, 10308]
- Uint8: enteros de 8 bits en el rango de [0, 255]
- Uint16: enteros de 16 bits en el rango de [0, 65535]
- Uint32: enteros de 32 bits en el rango de [0, 4294967295]
- Int8: enteros de 8 bits en el rango de [-128, 128]

- Int16: enteros de 16 bits en el rango de [-32768, 32767]. Es el tipo más común para el formato DICOM.
- Int32: enteros de 32 bits en el rango de [-2147483648, 2147483647]
- Logical: valores de 0 o 1

Matlab puede trabajar con cuatro tipos de imágenes: [20]

3.2.1.1 Imágenes indexadas

Una imagen indexada consiste en una matriz de datos, X (de tipo uint8, uint16 o double) y un mapa de color. El mapa de color es una matriz de $m \times 3$ de clase double que contiene valores entre 0 y 1.

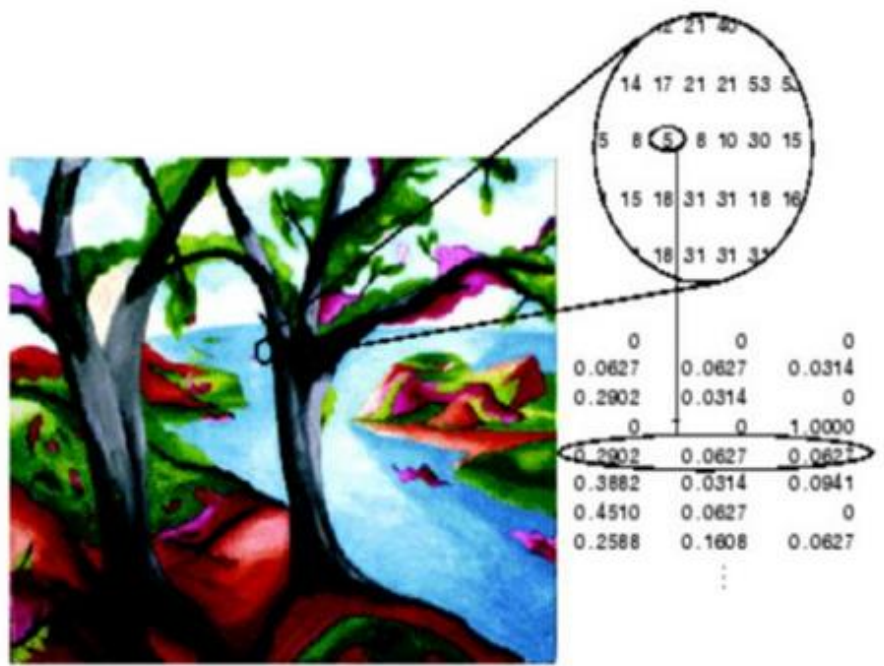


Ilustración 14. Ejemplo de imagen indexada

Cada una de las columnas de las que está compuesto el mapa de color especifica una componente de verde, rojo y azul de cada valor de índice de la matriz X.

3.2.1.2 Imágenes de intensidad

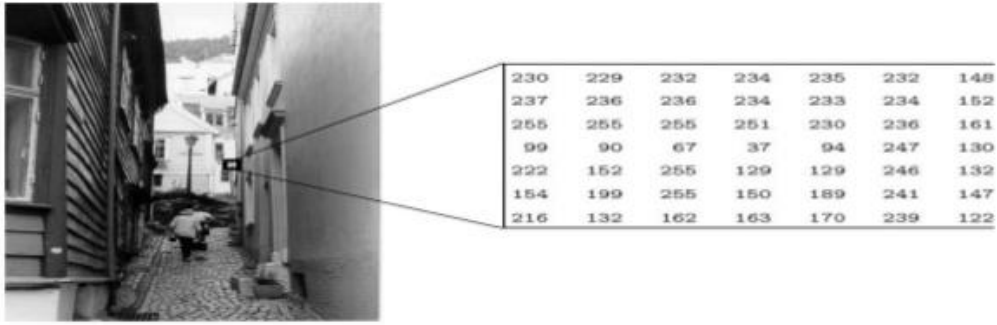


Ilustración 15. Ejemplo de imagen de intensidad

Este tipo de imágenes se componen de una sola matriz de datos I . Los valores de ésta representan las intensidades de los píxeles dentro de un rango (0 para el negro o la ausencia de color y, normalmente, 1 o 255 para el blanco).

Generalmente, las imágenes del formato DICOM con las que trabajaremos, serán de este tipo.

3.2.1.3 Imágenes binarias

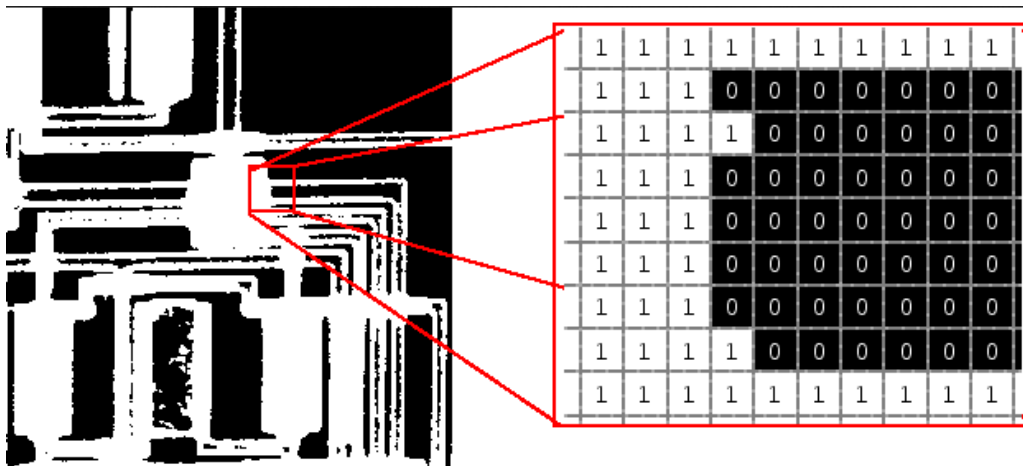


Ilustración 16. Ejemplo de imagen binaria

En este tipo de imágenes, cada píxel puede tener dos valores discretos que contengan únicamente blanco o negro. Esta imagen es guardada como una matriz bidimensional de ceros (negro) y unos (blanco).

3.2.1.4 Imágenes RGB

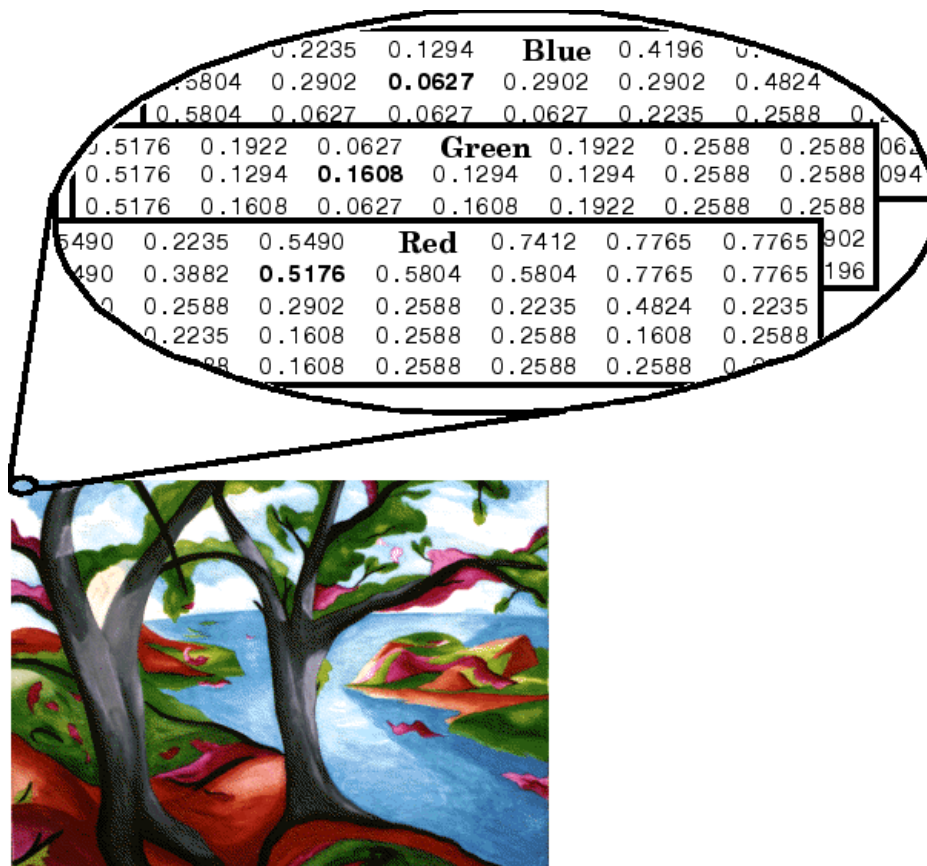


Ilustración 17. Ejemplo de imagen RGB

Una imagen RGB, también denominada imagen de color verdadero, se almacena como una matriz de $m \times n \times 3$ que define un valor de rojo, verde y azul para cada píxel. Una imagen de este tipo puede ser de clase `double`, `uint8` o `uint16`. Si la imagen RGB es de tipo `double`, cada componente de color tiene un valor entre cero y uno; teniendo en cuenta que el cero corresponde al color negro y el 1 al color blanco.

3.2.2 FUNCIONES BÁSICAS DE TRATAMIENTO DE IMÁGENES

3.2.2.1 Lectura de la imagen

Para leer imágenes en Matlab se utiliza el comando [21]

```
A = imread(filename)
```

Este comando lee la imagen del archivo especificado. Si el archivo tiene múltiples imágenes, sólo se lee la primera.

En el caso de las imágenes en formato DICOM, se utiliza [22]

```
X=dicomread(filename)
X = dicomread(info)
```

`X = dicomread (filename)` lee los datos de imagen desde el archivo de archivo DICOM (Digital Imaging and Communications in Medicine) compatible. Para las imágenes en escala de grises de un solo cuadro, `X` es una matriz $M \times N$. Para imágenes monocromáticas en color verdadero, `X` es una matriz $M \times N \times 3$. Las imágenes multitrama son siempre arrays 4-D.

3.2.2.2 Escritura de la imagen

Para guardar imágenes se utilizan los comandos [23]

```
imwrite(A, filename)
imwrite(A, map, filename)
```

`imwrite` escribe los datos de imagen `A` en el archivo especificado por nombre de archivo y los guarda en la carpeta actual. La profundidad de bits de la imagen de salida depende del tipo de datos de `A` y del formato de archivo. Para la mayoría de los formatos:

- Si `A` es del tipo de datos `uint8`, entonces `imwrite` emite valores de 8 bits.
- Si `A` es del tipo de datos `uint16` y el formato de archivo de salida admite datos de 16 bits (JPEG, PNG y TIFF), entonces `imwrite` produce valores de 16 bits. Si el formato de archivo de salida no admite datos de 16 bits, `imwrite` devuelve un error.
- Si `A` es una imagen de color en escala de grises o RGB de tipo de datos doble o simple, `imwrite` asume que el rango dinámico es $[0,1]$ y escala automáticamente los datos en 255 antes de escribirlo en el archivo como valores de 8 bits. Si los datos en `A` son individuales, se debe convertir `A` en doble antes de escribir en un archivo GIF o TIFF.
- Si `A` es del tipo de datos lógico, entonces `imwrite` asume que los datos son una imagen binaria y lo escribe en el archivo con una profundidad de bit de 1, si el formato lo permite. Los formatos BMP, PNG o TIFF aceptan imágenes binarias como matrices de entrada.

- Si A contiene datos de imagen indexados, debe especificar adicionalmente el argumento de entrada de mapa.

Para el caso de las imágenes DICOM [24]

```
dicomwrite(X, filename)
```

Escribe la imagen binaria, en escala de grises o en color verdadero en el archivo especificado.

3.2.2.3 Representación de la imagen

```
imshow(I)
```

```
image(I)
```

`imshow (I)` muestra la imagen I en una figura, donde I puede ser una imagen en escala de grises, RGB (truecolor) o imagen binaria. Para imágenes binarias, `imshow` muestra píxeles con el valor 0 (cero) como negro y 1 como blanco. `imshow` optimiza la figura, los ejes y las propiedades del objeto de imagen para la visualización de imágenes. [25]

`image(C)` muestra los datos en la matriz C como una imagen. Cada elemento de C especifica el color para un píxel de la imagen. La imagen resultante es una cuadrícula mxn de píxeles donde m es el número de columnas y n es el número de filas en C. Los índices de fila y columna de los elementos determinan los centros de los píxeles correspondientes. [26]

`Image` es un comando MATLAB que visualiza su matriz como simplemente una matriz de números. Los colores utilizados para representar cada valor pueden ser sin sentido a la representación de la matriz como una imagen.

`imshow` es un comando de la toolbox de procesamiento de imagen que trata a su matriz como una imagen. Se supone que los elementos son intensidades de píxeles y ofrece más control sobre el mapa de color, no dibuja las líneas de cuadrícula y mantiene la relación de aspecto.

3.2.2.4 Determinar el tamaño de la imagen

```
tam = size(A)
```

Devuelve un vector de fila cuyos elementos contienen la longitud de la dimensión correspondiente de A. Por ejemplo, si A es una matriz 3x4, entonces el tamaño (A) devuelve el vector [3 4].

Si A es una tabla o un calendario, el tamaño (A) devuelve un vector de fila de dos elementos que consta del número de filas y el número de variables de tablas. [27]

3.2.2.5 Selección de una zona de la imagen

```
I2 = imcrop
```

Crea una herramienta interactiva de recorte de imágenes asociada con la imagen mostrada en la figura actual, denominada imagen de destino. La herramienta Recortar imagen es un rectángulo ajustable que puede colocarse sobre la imagen y realizar la operación de recorte interactivamente con el ratón. Imcrop devuelve la imagen recortada, I2. [28]

3.2.2.6 Cambio de tamaño de la imagen

```
B = imresize (A, escala)
```

Devuelve la imagen B que es n veces el tamaño de A según la escala indicada. La imagen de entrada A puede ser una escala de grises, RGB o imagen binaria. Si A tiene más de dos dimensiones, `imresize` sólo redimensiona las dos primeras dimensiones. Si la escala es de 0 a 1, B es menor que A. Si la escala es mayor que 1, B es mayor que A. [29]

3.2.2.7 Conversión de tipo de imágenes

A veces, es necesario convertir a una imagen a un formato diferente para poder trabajar con ciertas funciones. Estas son las conversiones más comunes: [30]

- `Gray2ind`: Crea una imagen indexada a partir de una imagen de escala de grises.

- `im2bw`: Crea una imagen binaria de una de intensidad, o RGB, basado en umbralización de la luminancia.
- `Ind2gray`: Crea una imagen en escala de grises a partir de una imagen indexada.
- `Mat2gray`: Crea una imagen en escala de grises a partir de datos en una matriz.
- `Rgb2gray`: Crea una imagen en escala de grises a partir de una imagen RGB.
- `Rgb2ind`: Crea una imagen indexada a partir de una imagen RGB.

3.3 FORMATOS DE IMAGEN

El formato DICOM no deja de ser un formato más de imagen que ha sido adoptado internacionalmente para el manejo de imágenes médicas.

Antes de entrar en detalle con el formato DICOM, vamos a explicar otros de los formatos que se suelen utilizar.

3.3.1 *BMP (Windows Bitmap)*

Es un formato de imagen de mapa de bits, propio de Microsoft Windows. Los valores de los píxeles se definen de abajo arriba y de izquierda a derecha.

Las imágenes son de gran tamaño y no son utilizadas en páginas web por esa razón. Se les puede dar una compresión sin pérdidas de calidad.

3.3.2 *GIF (Graphics Interchange Format)*

Se diseñó para comprimir imágenes, utilizando el algoritmo de compresión LZW (Lempel Ziv Welch). Es muy útil a la hora de publicar imágenes en la web si no nos importa mucho la calidad de éstas, ya que, aunque el formato no tiene pérdidas, su máximo de colores es 256 (8 bits de profundidad).

3.3.3 JPEG (Joint Photographic Experts Group)

Permite una paleta de hasta 16 millones de colores. Las cámaras digitales suelen usar este formato. JPEG utiliza un algoritmo de compresión con pérdidas para que el tamaño de las imágenes no sea muy grande. Hay variantes, como JPEG 2000, que hacen la compresión de la imagen sin pérdida de información.

Dependiendo del grado de compresión que se elija la imagen tendrá más o menos tamaño, pero a su vez tendrá menos o más pérdidas de calidad.

Este formato calcula el valor del color de un píxel en función de los píxeles que lo rodean. Capta más claramente los cambios de brillo que de color ya que el algoritmo de compresión se basa en el funcionamiento del ojo humano.

3.3.4 TIFF (Tagged Image File Format)

Es un formato de archivo para guardar las imágenes de mapa de bits. Permite almacenar más de una imagen en el mismo archivo. La imagen tiene un gran tamaño. Al igual que el formato GIF, este tipo de archivos también usan el algoritmo de compresión LZW (Lempel Ziv Welch).

El formato TIFF permite almacenar imágenes en blanco y negro, en colores verdaderos (hasta 32 bits por píxel) y también indexar imágenes utilizando una paleta.

3.3.5 PNG (Portable Network Graphics)

Se creó como alternativa al GIF. Es un formato gráfico basado en un algoritmo de compresión sin pérdida para mapas de bits. Permite almacenar imágenes con una mayor profundidad de contraste.

3.3.6 PSD

Formato para el procesado de imágenes con Adobe Photoshop.

3.3.7 RAW

Contiene todos los píxeles de la imagen tal cual fueron tomados. Es el que suelen utilizar las cámaras réflex. Usa una compresión de datos sin pérdidas. Tiene una profundidad de color de 36 a 48 bits/píxel. La mayor desventaja es que cada fabricante suele usar una versión del formato, lo que provoca incompatibilidades.

3.4 FORMATO DICOM

3.4.1 MODALIDADES DE IMÁGENES MÉDICAS

En el contexto clínico, la imagen de la "luz invisible" es generalmente equiparada a radiología o "imagen clínica" y el médico responsable de interpretar (y algunas veces adquirir) las imágenes es un radiólogo.

La imagen médica "luz visible" incluye imágenes de vídeo digital o imágenes fijas que se pueden ver sin equipo especial. La dermatología y el cuidado de heridas son dos modalidades que usan imágenes de luz visible. La radiografía de diagnóstico designa los aspectos técnicos de la imagen médica y en particular la adquisición de imágenes médicas.

Como campo de investigación científica, la imagen médica constituye una subdisciplina de la ingeniería biomédica, la física médica o la medicina dependiendo del contexto: la investigación y el desarrollo en el área de instrumentación, adquisición de imágenes (por ejemplo, radiografía), modelización y cuantificación. Estos son algunos ejemplos de imágenes médicas:

3.4.1.1 Radiografías obtenidas con rayos X

Las radiografías son exámenes médicos no invasivos que utilizan una pequeña dosis de radiación ionizante para producir imágenes de cada hueso del cuerpo. Se suelen usar para diagnosticar fracturas o dislocaciones.

3.4.1.2 Resonancia magnética

La resonancia magnética (MRI) es una técnica de imagen médica utilizada en radiología para formar imágenes de la anatomía y los procesos fisiológicos del cuerpo tanto en la salud como en la enfermedad. Los escáneres de resonancia magnética utilizan fuertes campos magnéticos, ondas de radio y gradientes de campo para generar imágenes de los órganos del cuerpo.

La resonancia magnética no implica rayos X, lo que lo distingue de la tomografía computarizada (CT o CAT). Si bien los riesgos de los rayos X están ahora bien controlados en la mayoría de los contextos médicos, la resonancia magnética todavía puede considerarse superior a la TC en este sentido.

La resonancia magnética suele producir información diagnóstica diferente en comparación con la TC. Puede haber riesgos y molestias asociadas con las exploraciones de resonancia magnética. En comparación con la tomografía computarizada, los escáneres por resonancia magnética suelen tardar más tiempo, son más fuertes y, por lo general, requieren que el sujeto entre en un tubo estrecho y confinado. Además, las personas con algunos implantes médicos u otro metal no removible dentro del cuerpo pueden ser incapaces de someterse a un examen de resonancia magnética con seguridad.

3.4.1.3 Ultrasonidos

La ecografía médica (también conocida como ecografía diagnóstica o ultrasonografía) es una técnica de diagnóstico por imágenes basada en la aplicación de ultrasonido. Se utiliza para ver estructuras internas del cuerpo tales como tendones, músculos, articulaciones, vasos y órganos internos. Su objetivo es a menudo encontrar una fuente de una enfermedad o excluir cualquier patología. La práctica de examinar a las mujeres embarazadas usando ultrasonido se llama ecografía obstétrica, y es ampliamente utilizado.

Los ultrasonidos son ondas sonoras con frecuencias superiores a las audibles para los seres humanos (> 20.000 Hz). Las imágenes ultrasónicas también conocidas como sonogramas se realizan mediante el envío de pulsos de ultrasonido en el tejido utilizando una sonda. El sonido se hace eco del tejido, con diferentes tejidos que reflejan grados variables de sonido. Estos ecos se registran y se muestran como una imagen para el operador.

En comparación con otros métodos prominentes de imágenes médicas, el ultrasonido tiene varias ventajas. Proporciona imágenes en tiempo real, es portátil y se puede llevar a la cabecera, es sustancialmente menor en costo, y no utiliza radiación ionizante perjudicial.

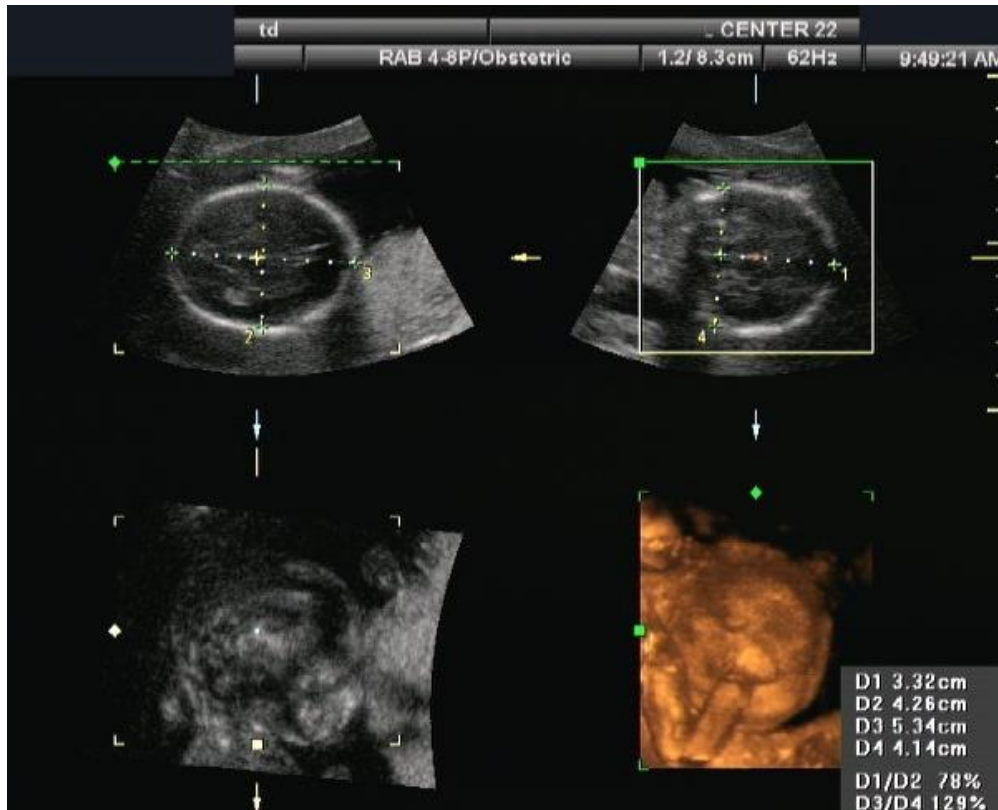


Ilustración 18. Ejemplo de imagen médica obtenida con ultrasonidos

3.4.1.4 Medicina nuclear

La medicina nuclear es una especialidad médica que implica la aplicación de sustancias radiactivas en el diagnóstico y tratamiento de enfermedades. La medicina nuclear, en cierto sentido, es "radiología hecha de adentro hacia afuera" porque registra la radiación que emite desde dentro del cuerpo en lugar de la radiación que es generada por fuentes externas como rayos X. Además, las exploraciones de medicina nuclear difieren de la radiología ya que el énfasis no está en la anatomía de la imagen, sino en la función y, por tal razón, se le llama modalidad de imagen fisiológica.

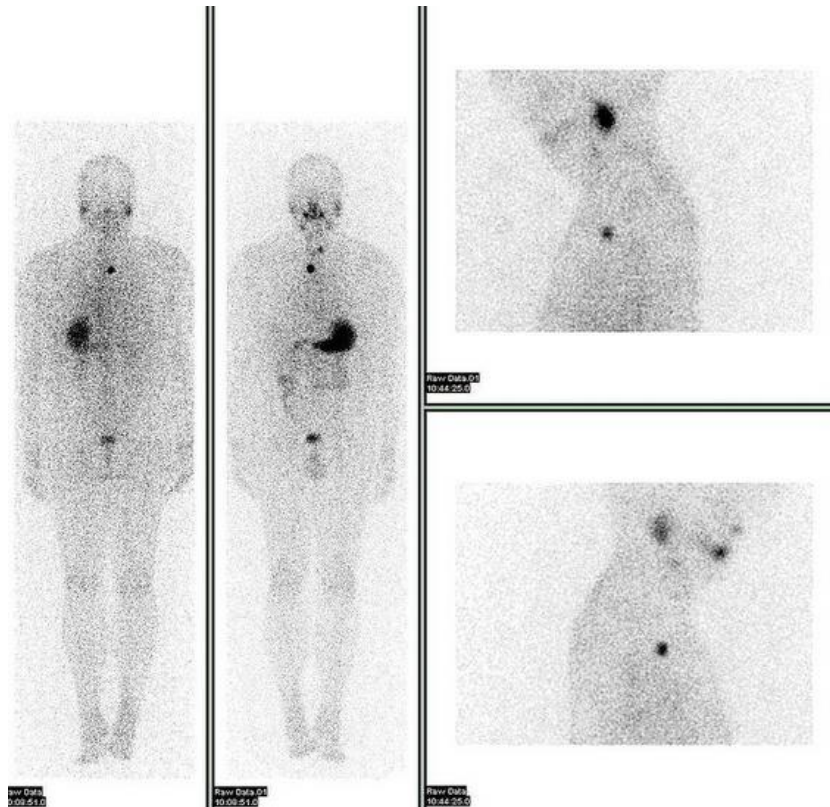


Ilustración 19. Ejemplo de imagen médica obtenida con medicina nuclear

3.4.2 IMAGEN DICOM

3.4.2.1 ¿Qué es DICOM?

DICOM (Digital Imaging and Communication in Medicine) es un estándar aceptado a nivel mundial para el intercambio, impresión, utilización, visualización, almacenamiento y transferencia de imágenes médicas.

Este tipo de ficheros se intercambian entre dos organismos que puedan recibir imágenes e información de pacientes en este formato. [31]

3.4.2.2 ¿Qué contiene DICOM?

Este tipo de archivos están formados por dos partes diferenciadas:

- La imagen: su representación digital.

- Los metadatos de la imagen: una estructura de información sobre el paciente, el centro médico o el doctor que haya tomado la imagen.

3.4.2.3 Historia

Con la introducción de la tomografía computarizada (TC), seguida de otras modalidades de diagnóstico por imágenes digitales en la década de 1970, y el uso creciente de computadoras en aplicaciones clínicas, el Colegio Americano de Radiología (ACR) y la National Electrical Manufacturers Association (NEMA) reconocieron la necesidad emergente de un método estándar para transferir imágenes e información asociada entre dispositivos fabricados por varios vendedores. El Colegio Americano de Radiología (ACR) y la Asociación Nacional de Fabricantes Eléctricos (NEMA) formaron un comité conjunto en 1983 para desarrollar un estándar para:

- Promover la comunicación de información de imágenes digitales, independientemente del fabricante del dispositivo.
- Facilitar el desarrollo y la expansión de los sistemas de archivo y comunicación de imágenes (PACS) que también pueden interactuar con otros sistemas de información hospitalaria.
- Permitir la creación de bases de datos de información diagnóstica que puedan ser consultadas por una amplia variedad de dispositivos distribuidos geográficamente.

3.4.2.4 Aplicaciones

Las definiciones de objeto de información DICOM codifican los datos producidos por una amplia variedad de tipos de dispositivos de generación de imágenes, entre ellos:

Hay multitud de dispositivos que producen imágenes DICOM en sus procesos:

- TC (tomografía computarizada)
- MRI (resonancia magnética)

- Ultrasonido
- Radiografía
- Fluoroscopia
- Angiografía
- Mamografía
- Tomosíntesis mamaria
- PET (tomografía por emisión de positrones)
- SPECT (tomografía computarizada de emisión de fotón único)
- Endoscopia
- Microscopía
- Imágenes de diapositivas enteras
- OCT (tomografía de coherencia óptica)

DICOM también es implementado por dispositivos asociados con imágenes o flujo de trabajo de imágenes incluyendo:

- PACS (sistemas de archivo y comunicación de imágenes)
- Visualizadores de imágenes y estaciones de visualización
- CAD (sistemas de detección / diagnóstico asistidos por ordenador)
- Sistemas de visualización 3D
- Aplicaciones de análisis clínico
- Impresoras de imagen
- Escáneres de películas

- Quemadores de medios (que exportan archivos DICOM a CD, DVD, etc.)
- Los importadores de medios (que importan archivos DICOM de CD, DVD, USB, etc.)
- RIS (sistemas de información radiológica)
- VNA (archivos neutrales a los proveedores)
- Sistemas de EMR (registro médico electrónico)
- Sistemas de información radiológica

Muchos campos de la medicina tienen un grupo de trabajo dedicado dentro de DICOM, y DICOM es aplicable a cualquier campo de la medicina en que la imagen sea frecuente, incluyendo:

- Radiología
- Cardiología
- Oncología
- Radioterapia
- Neurología
- Ortopedia
- Obstetricia
- Ginecología
- Oftalmología
- Odontología
- Cirugía Maxilofacial
- Dermatología

- Patología
- Ensayos clínicos
- Medicina Veterinaria

4. DISEÑO

Se aportará como anexo el código de la aplicación.

4.1 OPCIONES GENERALES

4.1.1 CARGA Y VISUALIZACIÓN DE IMÁGENES

4.1.1.1 Utilidad

Esta función nos permite abrir las imágenes con las que vamos a trabajar. Tenemos dos opciones de visualización:

- Imagen individual.
- Carpeta de imágenes.

Para esta segunda opción tenemos unos botones de navegación para recorrer el conjunto de imágenes hacia delante o hacia atrás.

4.1.1.2 Implementación

La función `[FileName, PathName, map] = uigetfile({'*.dcm'}, 'Select .dcm files', 'MultiSelect', 'on')` nos abre una ventana donde podemos elegir entre seleccionar una única imagen o un conjunto de ellas.

En el caso de querer trabajar con más de una imagen, se crea una estructura de datos que tendría la longitud del número de imágenes que la contienen. Lo que se hace es crear un array de nombres de imágenes y a cada uno de los nombres se le asigna un índice que vamos recorriendo en bucle.

Para la visualización de una imagen en formato DICOM, usamos la función

```
info = dicominfo(FileName{i});  
Y = dicomread(info);
```

Para mostrar la imagen, se usa el comando `imshow(a)`; indicando con `axes(handles.Imagen)`; que queremos que esa imagen se muestre en la figura con la etiqueta "Imagen".

4.1.1.3 Funcionamiento

Para abrir la imagen (o imágenes) con las que vamos a trabajar, nos situamos en el menú superior de la interfaz y en el submenú de “Archivo”, pulsamos “Abrir”.

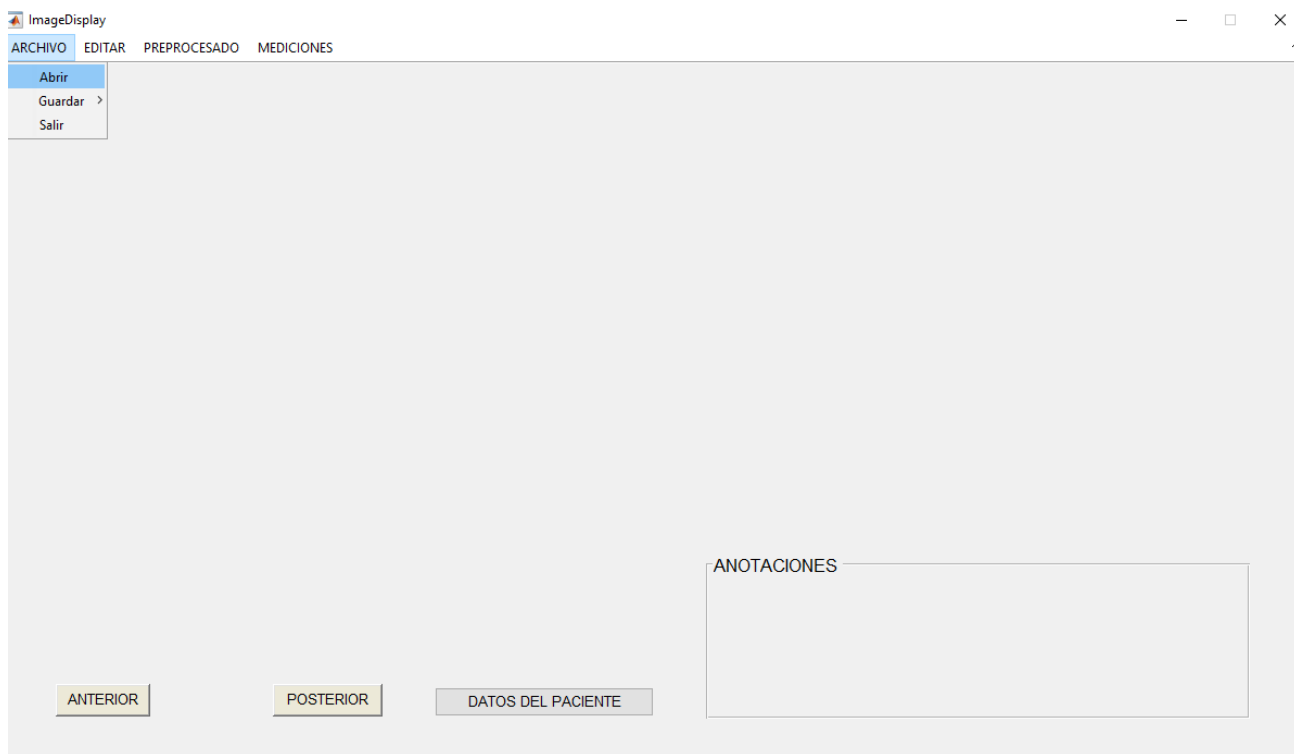


Ilustración 20. Imagen de apertura de archivo

Aquí se nos abre una ventana donde podemos seleccionar una sola imagen, o un conjunto de ellas.

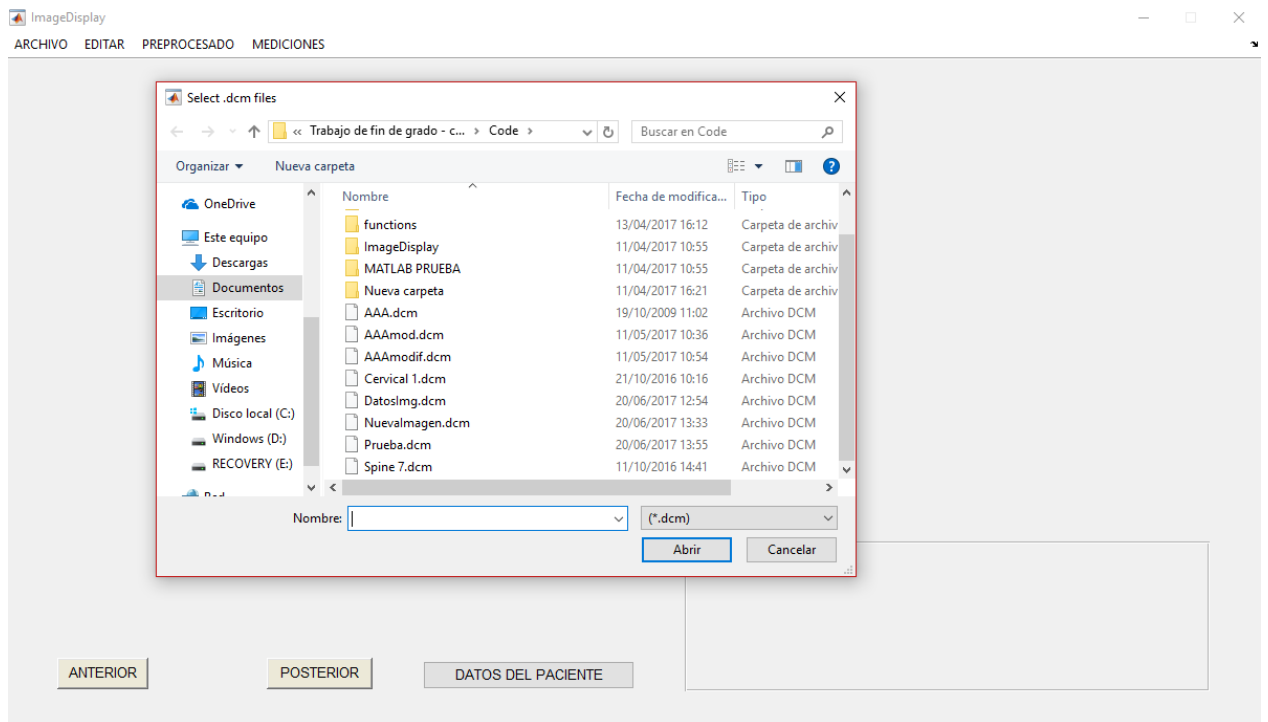


Ilustración 21. Interfaz de selección de archivos

Una vez seleccionada nuestra preferencia, nos aparece la primera imagen. Si queremos visualizar el resto de imágenes, basta con pulsar los botones de navegación (“Anterior” y “Posterior”).



Ilustración 22. Visualización de la plataforma

En el caso de recorrer el conjunto de imágenes que hemos seleccionado de una carpeta, nótese que la aplicación nos avisará cuando lleguemos a la última imagen y le sigamos dando a “Posterior”. Lo mismo ocurre si le damos a “Anterior” después de estar situados en la primera imagen.

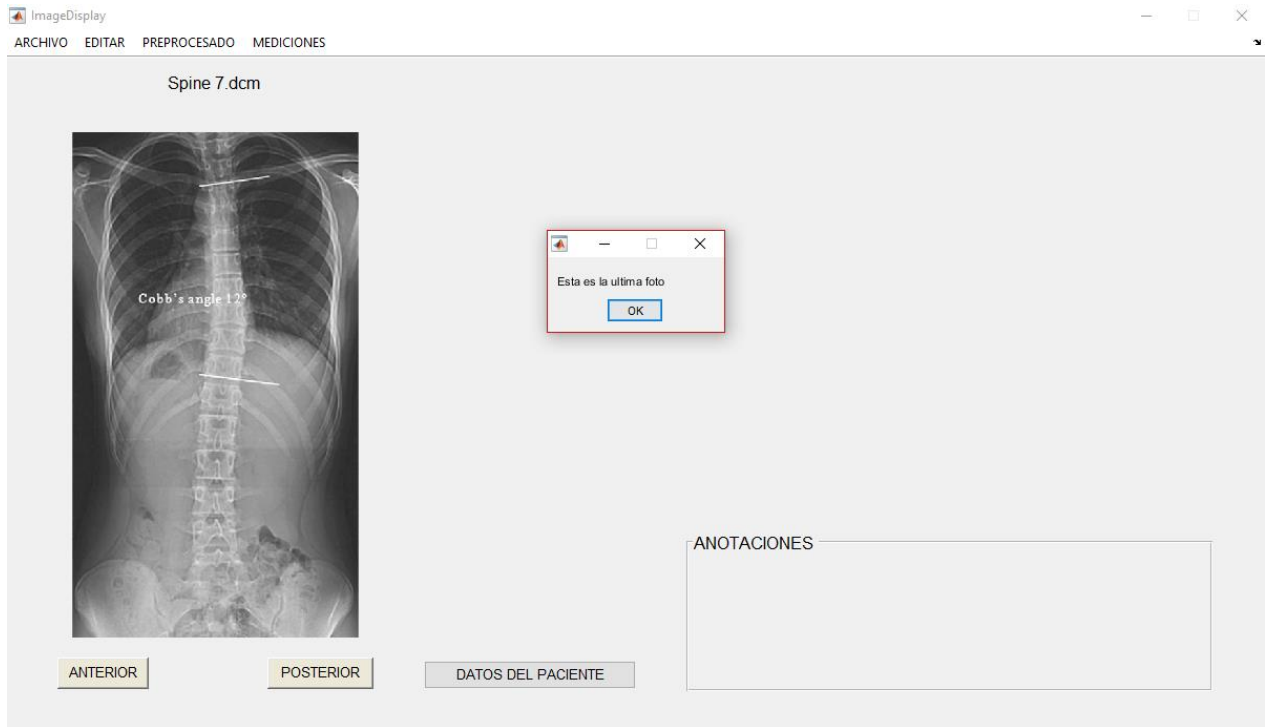


Ilustración 23. Mensaje que aparece al llegar a la última foto

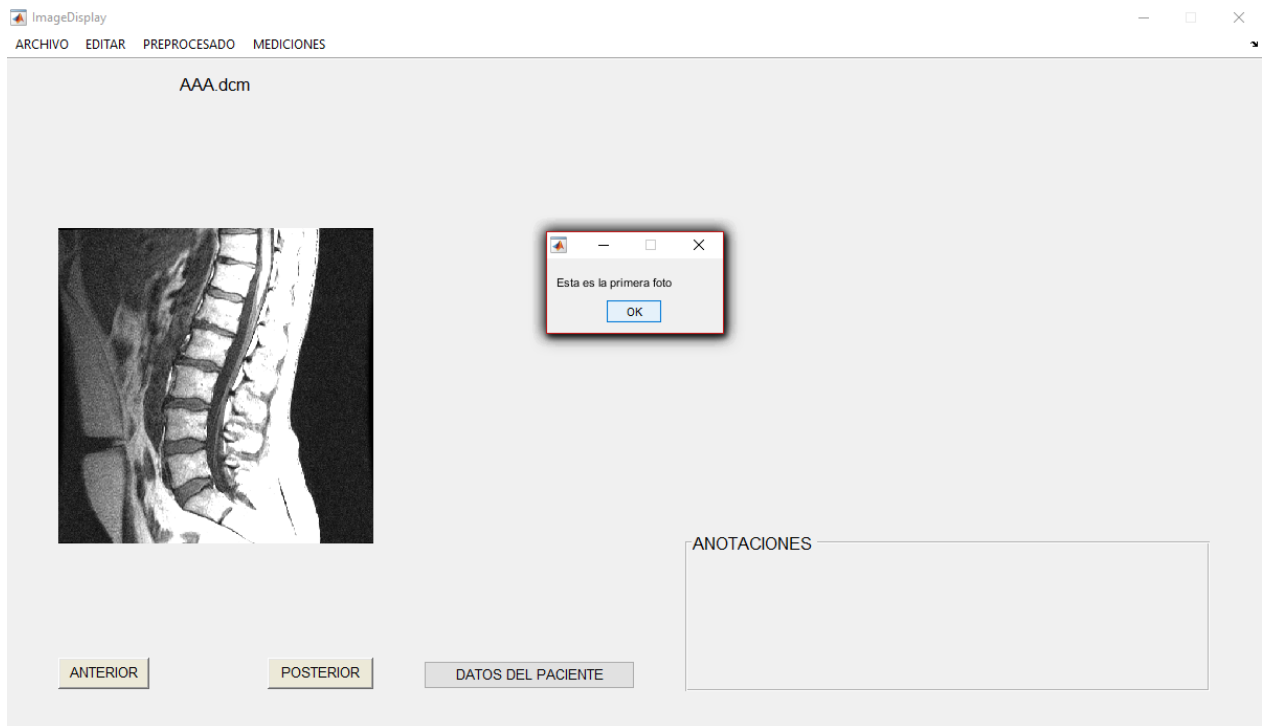


Ilustración 24. Mensaje que aparece si llegamos a la primera foto

4.1.2 CAMBIO DE FORMATO Y ALMACENAMIENTO

4.1.2.1 Utilidad

Esta función sirve para guardar, en un directorio elegido, la imagen DICOM con la que hemos estado trabajando. La imagen se puede guardar en dos formatos: DICOM o JPG. Éste último formato es útil si queremos visualizar la imagen, pero no disponemos de un visualizador DICOM.

4.1.2.2 Implementación

Para guardar la imagen en formato JPG, usando `getframe()` se extrae directamente de nuestro eje, que en este caso hemos llamado “Imagen” donde la hemos representado previamente, similar a hacer una captura de pantalla. Nos da la imagen de cambiar el nombre del archivo si así lo deseáramos. [32]

```
ImgJPG=getframe(handles.Imagen);  
figure();  
imshow(ImgJPG.cdata, []);  
axis off  
ImgJPG=frame2im(ImgJPG);  
[name, dir] = uiputfile('*.jpg', 'Guardar en');  
name = fullfile(dir, name);  
imwrite(ImgJPG, name);
```

4.1.2.3 Funcionamiento

Para guardar nuestras imágenes en la memoria, sólo tenemos que navegar por el menú “Archivo” y en “Guardar” elegir el formato en el que queremos que se almacene la imagen.

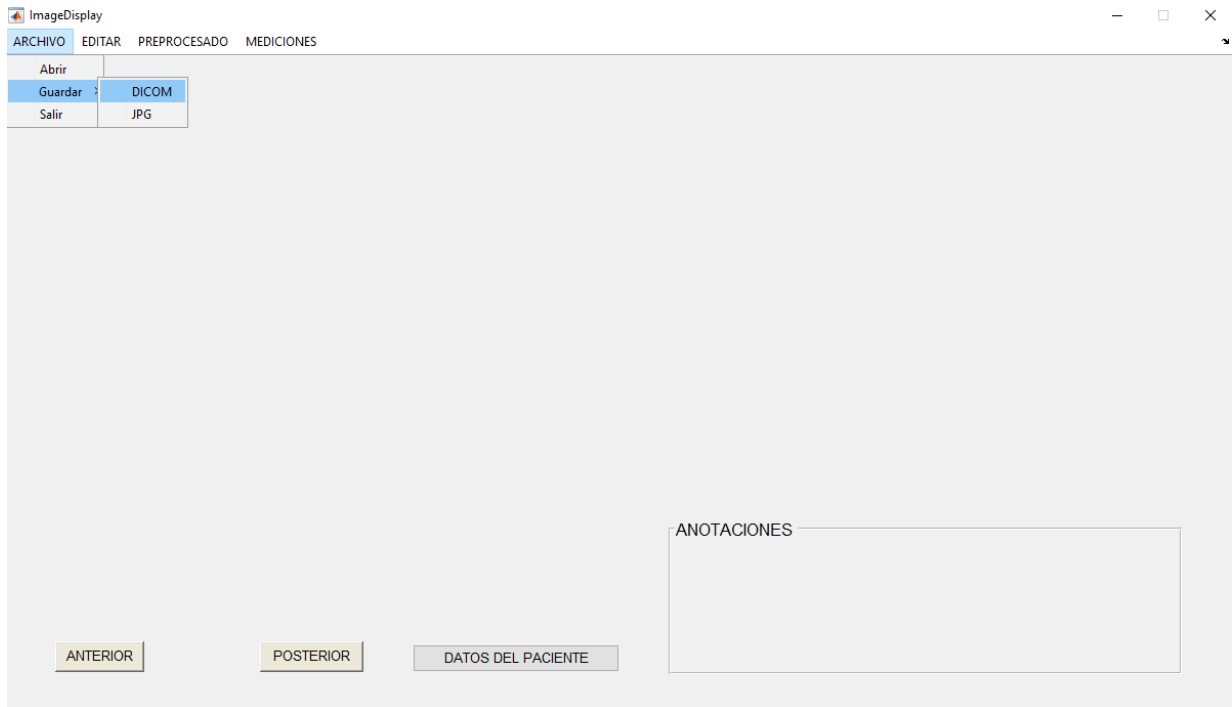


Ilustración 25. Menu de guardado de imágenes.

Si queremos guardar la imagen en formato JPG, se nos abre una ventana de previsualización y otra ventana donde le indicamos el nombre y el directorio que le vamos a asignar a la nueva imagen.

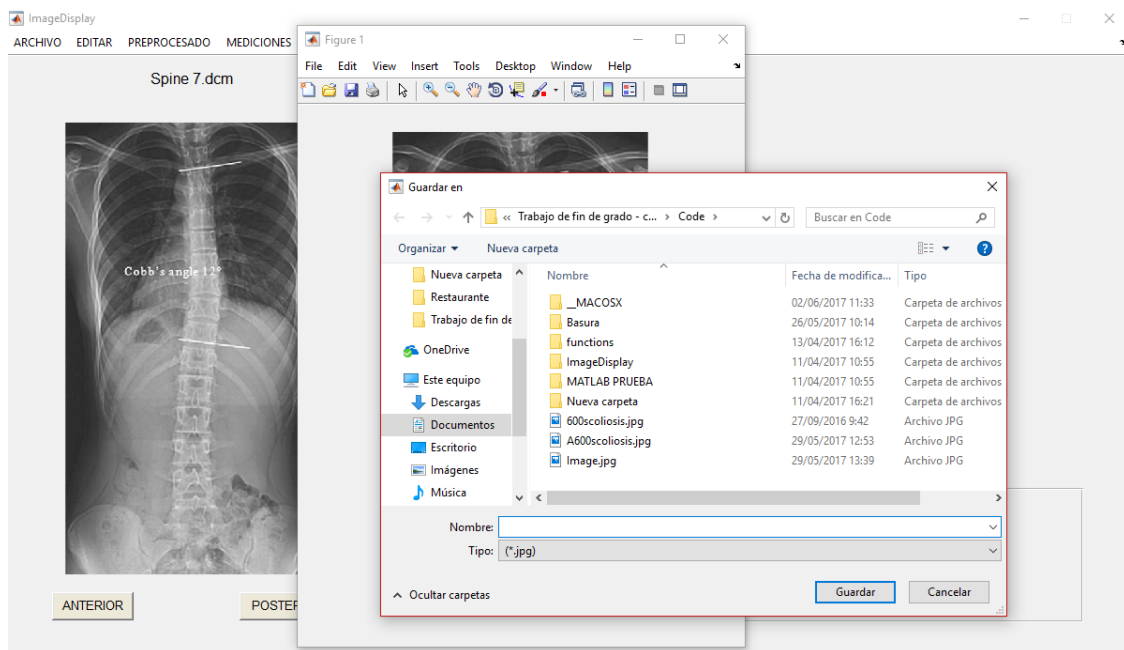


Ilustración 26. Ventana para poder guardar la imagen

4.1.3 SALIDA DE LA INTERFAZ

4.1.3.1 Utilidad

Esta función nos permite salir de la interfaz cerrando todas las ventanas que la aplicación hubiera abierto.

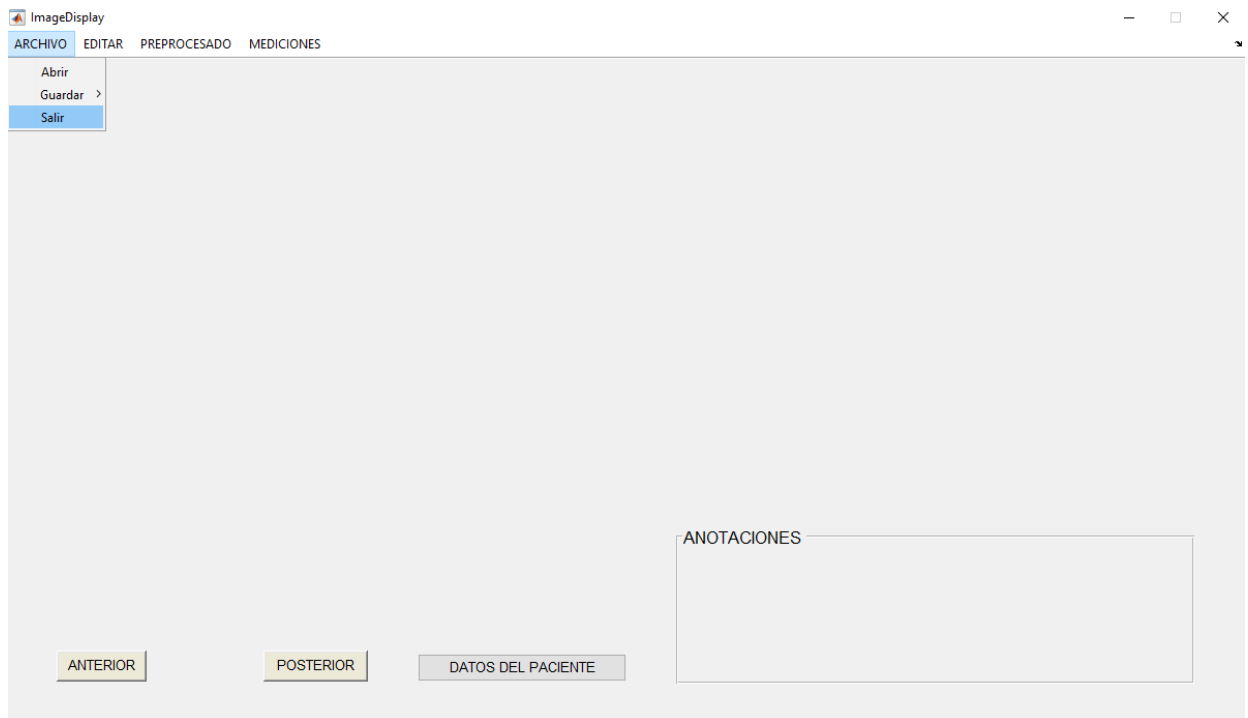


Ilustración 27. Ruta de menú para salir de la plataforma

4.1.3.2 Implementación

Se usa la función `close all;` para cerrar todas las ventanas que hubiese abiertas.

4.2 OPCIONES DE EDICIÓN BÁSICAS

4.2.1 RECORTE

4.2.1.1 Utilidad

Esta función nos permite escoger un cuadro de la imagen y desechar el resto. Esto hace que podamos ver la imagen con más detalle y trabajar sobre ella con facilidad.

4.2.1.2 Implementación

Para dibujar el rectángulo en la imagen, usamos `imrect`. Esto nos permite dibujar las líneas que delimitan a nuestra nueva imagen. [33]

Con `position` tendremos los vértices de la imagen y a partir de ahí, cogemos esos nuevos vértices para la nueva imagen. [34]

`Resize` nos transforma nuestra imagen recortada al mismo tamaño de la imagen anterior.

```
h = imrect;
position = wait(h);
X1 = position(1);
X2 = position(3)+position(1);
Y1 = position(2);
Y2 = position(4)+position(2);
newImage=a(Y1:Y2,X1:X2);
[outputImage] = Resize(newImage);
```

Este código se encarga de descartar los píxeles de la zona que queden fuera de las líneas que hayamos dibujado y sólo representar los que hayan quedado en el área ajustando el tamaño total de la imagen nueva al tamaño de la anterior.

4.2.1.3 Funcionamiento

Se accede a esta opción a través del menú “Editar”

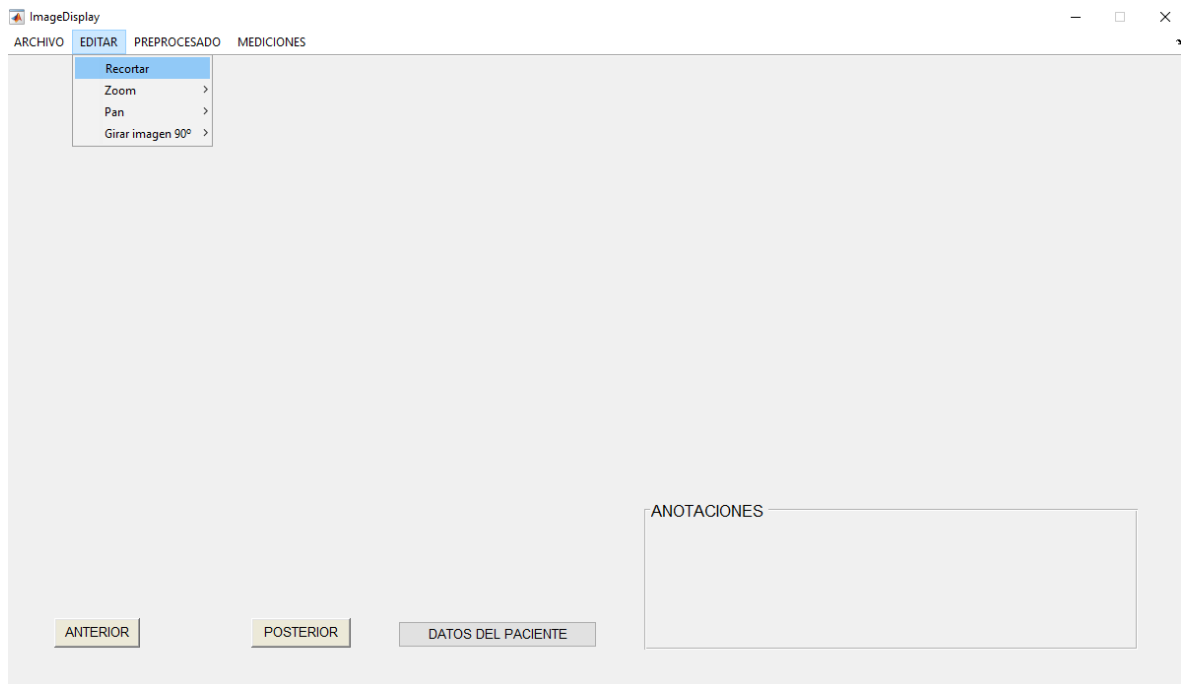


Ilustración 28. Navegación en el menú para llegar a la función de recorte

Nos saldrá el cursor con una cruz y eso indica que la función ya está lista para usarse. Pinchamos en un punto y arrastramos hacia el punto contrario para formar el rectángulo.

Una vez dibujado, podemos tanto cambiarlo de lugar pulsando en el centro y arrastrando, como ampliarlo o reducirlo tocando los pequeños cuadros que hay en las líneas de borde.



Ilustración 29. Selección de la zona a recortar

Quando ya tengamos el cuadro en la posición que queramos, hacemos doble click sobre él y ya nos aparece la imagen recortada.



Ilustración 30. Imagen recortada

4.2.2 ZOOM

4.2.2.1 Utilidad

Esta opción nos permite alejar o acercar una imagen para visualizar detalles que de otro modo sería imposible.

4.2.2.2 Implementación

En este caso no ha sido necesario el desarrollo de una función, ya que MatLab tiene esta opción implementada `zoom (figure, factor);`. La figura en nuestro caso será el axis en el cual estamos representando la imagen. [35]

4.2.2.3 Funcionamiento

Teniendo ya la imagen cargada en nuestra interfaz,

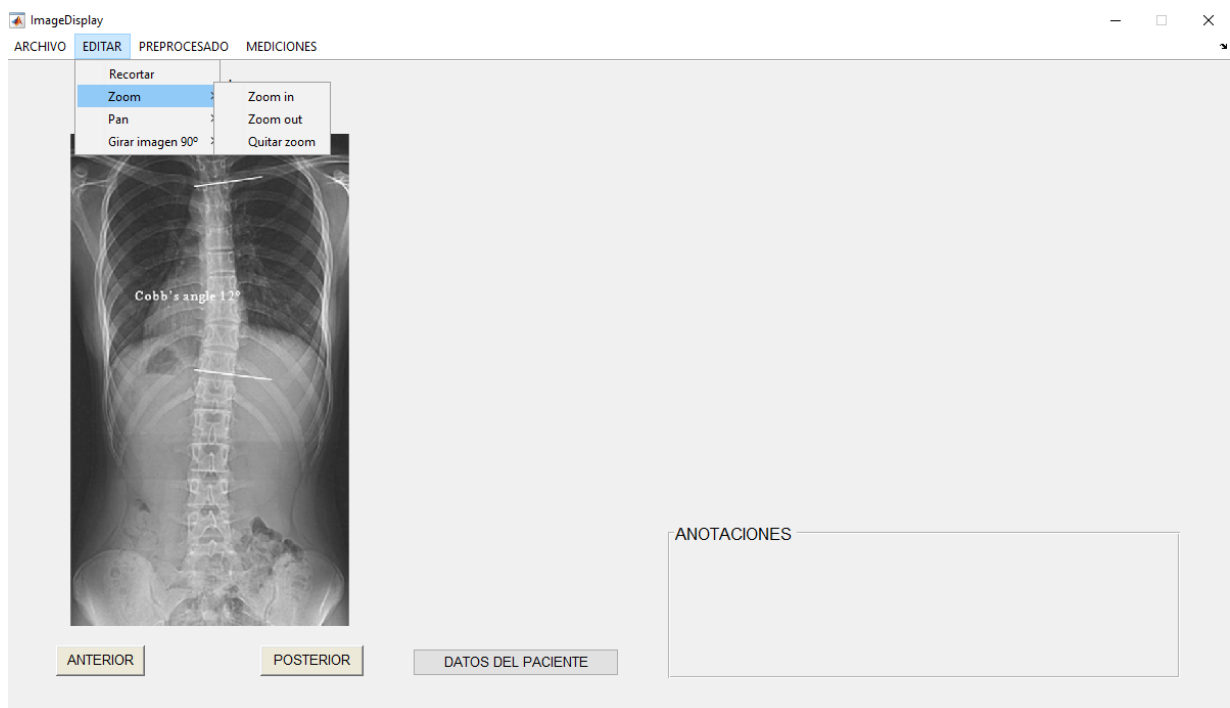


Ilustración 31. Recorrido del menú para llegar a la opción de zoom

4.2.3 PAN

4.2.3.1 Utilidad

Esta función permite que, después de hacer zoom sobre la imagen, podamos desplazarla para poder ver otros detalles. Esto puede ser útil en el caso de que queramos guardar varios trozos de la imagen por separado para procesarlos de forma distinta. [36]

4.2.3.2 Implementación

Esta función se activa y se desactiva con:

```
pan on;  
pan off;
```

4.2.3.3 Funcionamiento

Esta función se encuentra en el menú de edición.



Ilustración 32. Recorrido en el menú para llegar a la función PAN

Al arrastrar el cursor por la imagen, ésta se mueve y podemos dejarla quieta en un punto que nos interese.



Ilustración 33. Resultado de aplicar la función

4.2.4 TRANSFORMACIONES GEOMÉTRICAS DE LA IMAGEN

4.2.4.1 Utilidad

Se presentan un conjunto de funciones que permitan girar la imagen sobre un punto para facilitar su visualización.

4.2.4.2 Fundamento teórico

Estas transformaciones cambian la relación de espacio entre píxeles. Se utilizan para corregir distorsiones geométricas, para aplicar efectos geométricos y para el registro de imágenes. Una transformación geométrica consiste en: [37]

- Una transformación que define la nueva disposición de los píxeles en el plano de la imagen.
- Interpolación de los niveles de grises.

En las transformaciones espaciales cada punto de una imagen (x,y) se mapea en un nuevo punto del sistema de coordenadas.

Una transformación afín es un conjunto de transformaciones lineales en las que las coordenadas finales del punto se expresan linealmente en términos de las del punto original. Los puntos que se encuentran en una línea, permanecen en la misma después de la transformación. Estas transformaciones se componen de rotaciones, traslaciones, magnificación e inclinación.

$$\begin{aligned} x' &= ax + by + m \\ y' &= cx + dy + n \end{aligned} \quad (1)$$

Si $m=n=0$

$$\begin{cases} x' = ax + by \\ y' = cx + dy \end{cases} \longrightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2)$$

Las transformaciones afines también permiten la rotación de la imagen un ángulo determinado alrededor de un punto. Éstas conservan el paralelismo entre líneas, pero no sus ángulos y longitudes.

$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3)$$

4.2.4.3 Implementación

Giro hacia la derecha:

```
Der = imrotate(a,-90);
a = Der;
```

Giro hacia la izquierda:

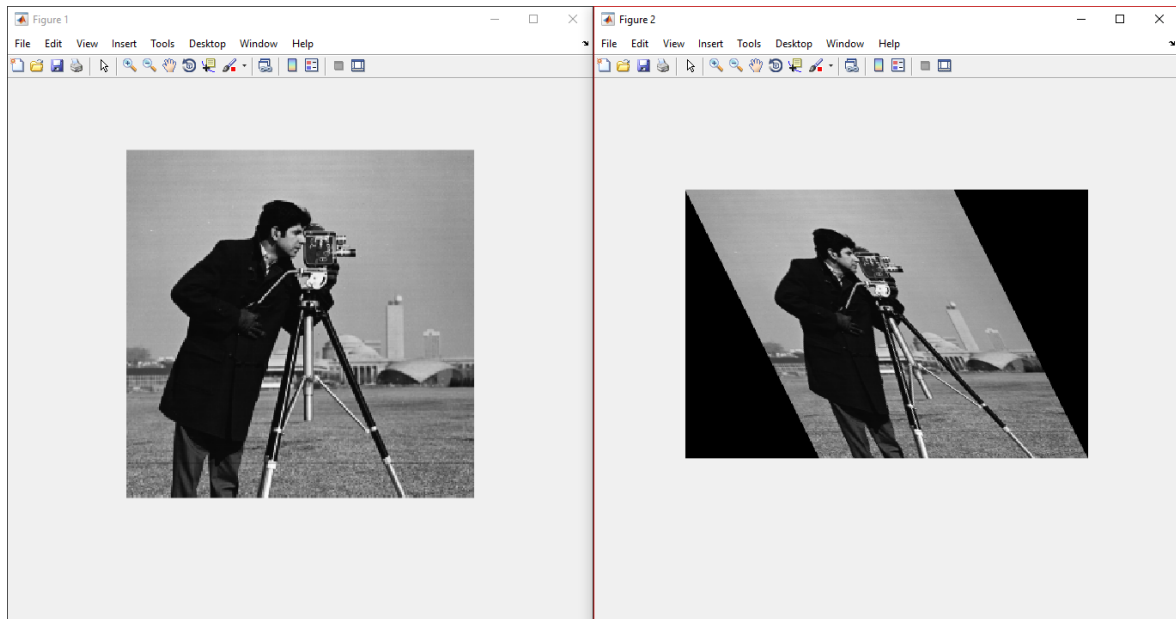
```
Izq = imrotate(a,90);
a = Izq;
```

Función `maketform`

Crema una estructura de transformación espacial.

```
Tform = Maketform ('tipo de transformacion', [matriz de transformaci3n])
```

En tipo de transformaci3n se puede seleccionar entre una afn, proyectiva, de caja o t3pica. En nuestro caso optaremos por una afn porque el resto no resultan tan 3tiles para este tipo de im3genes.



Ilustraci3n 34. Ejemplo de transformaci3n de inclinaci3n en la direcci3n x

Funci3n `imtransform`

Crema una transformaci3n en relaci3n a la estructura creada con `maketform` y la imagen que tengamos.

```
Imtransform ('Imagen', Tform).
```

4.2.4.4 Funcionamiento

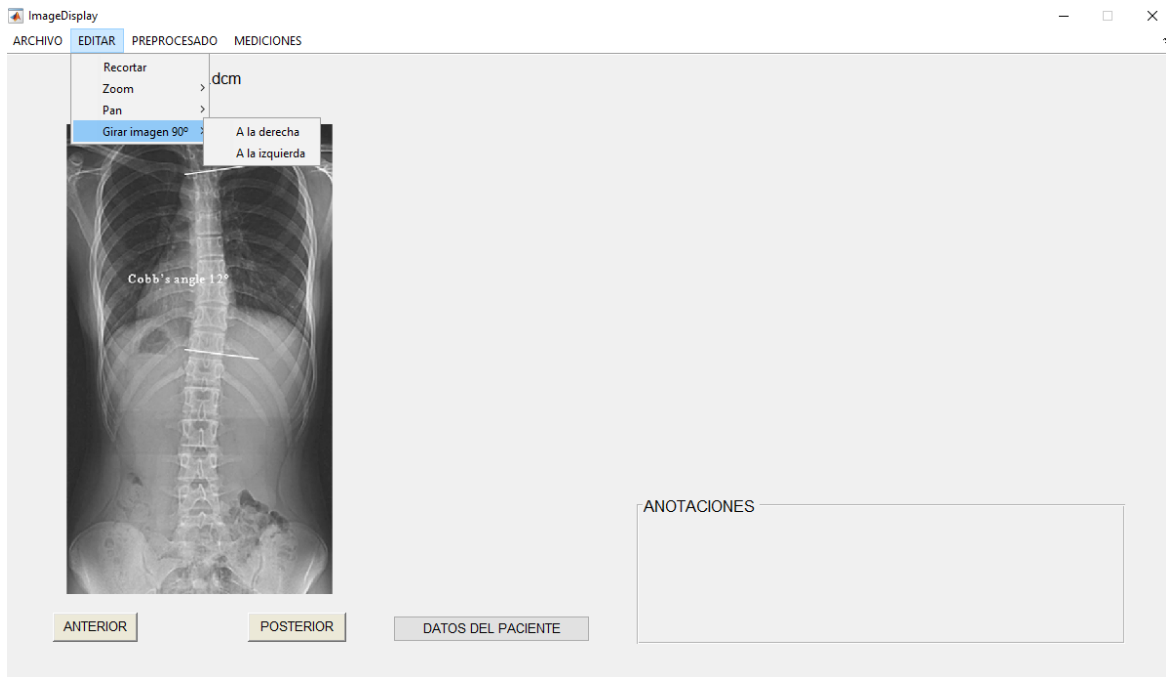


Ilustración 31. Recorrido en el menú para llegar a la función de giro

Si giramos la imagen a la derecha



Ilustración 352. Giro a la derecha de la imagen

Si giramos a la izquierda



Ilustración 33. Giro a la izquierda de la imagen

4.3 OPCIONES DE PRE PROCESADO

4.3.1 RUIDO

El ruido digital es la variación aleatoria en la imagen producido por una reproducción alteración del color o la luminosidad en el dispositivo de entrada.

Tipos de ruido:

- Ruido impulsional o “Sal y pimienta”

En este tipo de ruido, los píxeles de la imagen son muy diferentes en color o intensidad a los de su alrededor. En general, este tipo de ruido afectará a una pequeña cantidad de píxeles de la imagen.



Ilustración 36. Ejemplo de imagen con ruido sal y pimienta

- Ruido Gaussiano

En este tipo de ruido, todos los píxeles de la imagen cambian su valor de acuerdo con una distribución normal o gaussiana.

4.3.1.1 Utilidad

El objetivo de esta herramienta es la eliminación del ruido que pueda existir en la imagen debido al proceso de captura o en procesos posteriores.

4.3.1.2 Fundamentos teóricos

Antes de explicar las funciones que usaremos, vamos a hacer una introducción sobre los distintos filtros aplicables a imágenes. Para ellos hablaremos de los conceptos de DFT y máscara.

La DFT es la Transformada Discreta de Fourier y, en este caso, usaremos su versión para dos dimensiones.

El filtrado en el dominio de la frecuencia consiste en modificar la Transformada de Fourier de una imagen y luego calcular la transformada inversa para obtener la imagen procesada.

Gracias a la propiedad de separabilidad, puede realizarse la Transformada de Fourier por filas, y al resultado por columnas.

$$F(u, v) = \frac{1}{N} \sum_{j=1}^{N_1} \sum_{k=1}^{N_2} f(j, k) e^{-\left(\frac{2\pi i}{N}(uj+vk)\right)} \quad (4)$$

$$f(j, k) = \frac{1}{N} \sum_{j=1}^{N_1} \sum_{k=1}^{N_2} F(u, v) e^{-\left(\frac{2\pi i}{N}(uj+vk)\right)} \quad (5)$$

El resultado de la DFT está separado en módulo y fase. Se suele representar el módulo. También se intenta que las componentes de baja frecuencia queden centradas y las de alta frecuencia queden en los extremos.

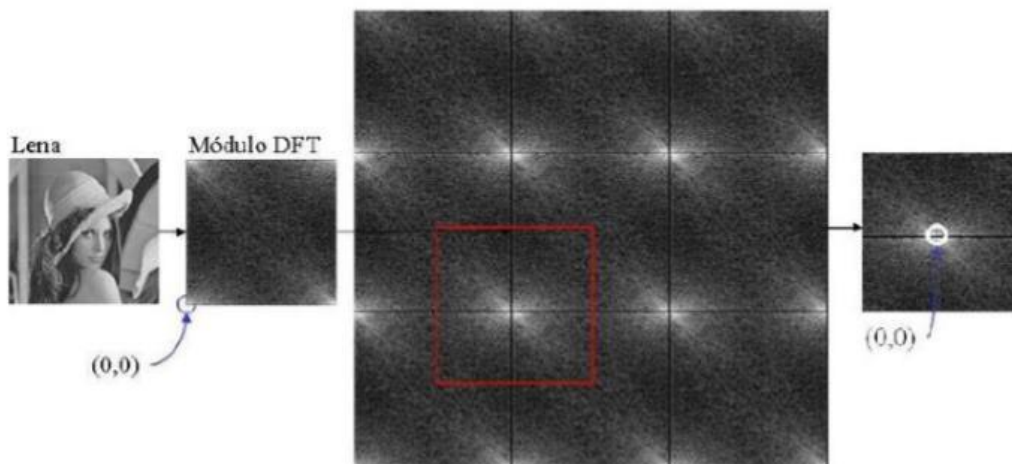


Ilustración 37. Imagen, transformada periódica y un periodo de la misma centrado en bajas frecuencias.

La máscara es una matriz más pequeña que la imagen que vayamos a filtrar y que nos sirve para recorrer la imagen completa poco a poco realizando una operación sobre las secciones por los que va pasando. Con este método se evita tener que trabajar con la matriz completa de la imagen.

4.3.1.3 Implementación

- Ruido Sal y Pimienta [38]

```
K = medfilt2(imagen);
```

Esta función realiza el filtrado medio de la imagen en las dos dimensiones. Cada píxel de salida contiene el valor mediano en una vecindad 3x3 alrededor del píxel correspondiente en la imagen de entrada.

Se imprime la imagen de salida con ceros en los bordes, por lo que los valores medianos para puntos dentro de la mitad del ancho del vecindario ($(m\ n)/2$) de los bordes pueden aparecer distorsionados.

- Ruido Gaussiano

```
K = wiener2(imagen, [2 2]);
```

Esta función hace un filtrado paso bajo a una imagen de escala de grises que ha sido degradada por ruido aditivo de potencia constante.

4.3.1.4 Funcionamiento

Ruido Sal y Pimienta.

Abrimos una imagen y descubrimos que tiene ruido de tipo Sal y Pimienta.



Ilustración 34. Imagen con ruido sal y pimienta

Abrimos el menú de preprocesado y aplicamos el filtro correspondiente.

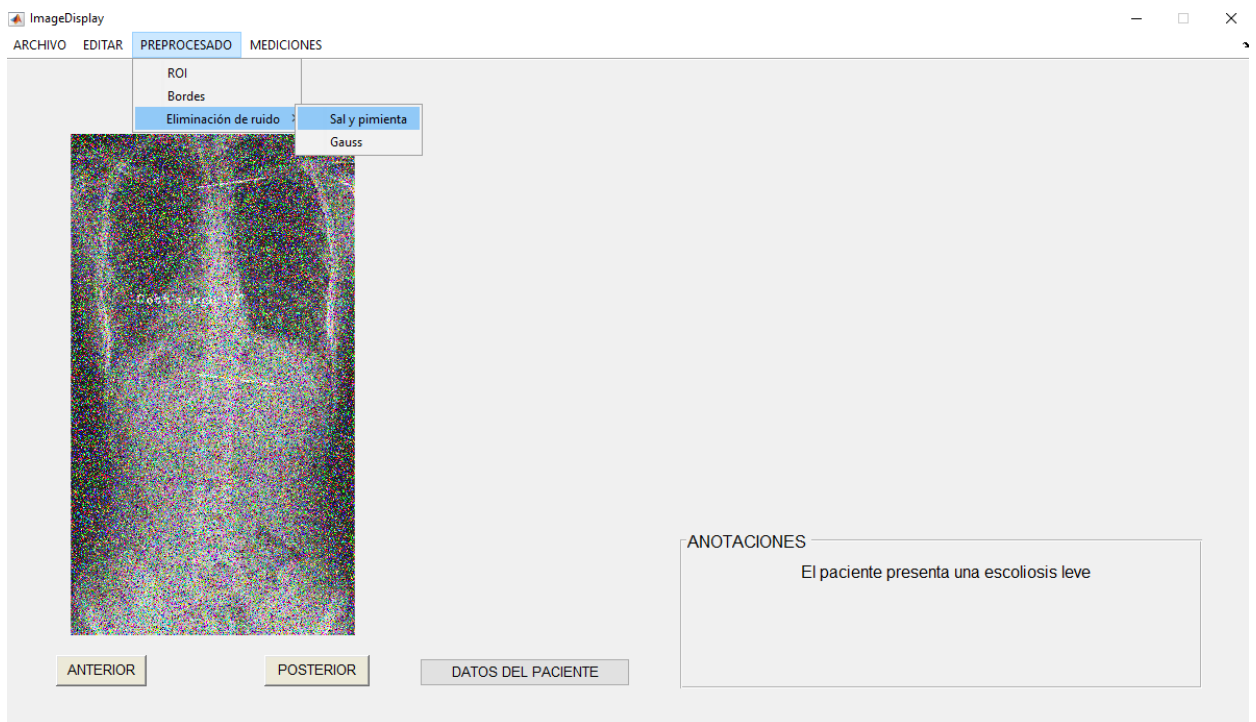


Ilustración 35. Recorrido en el menú para eliminar el ruido sal y pimienta

Vemos una clara mejoría en la imagen.



Ilustración 36. Resultado después de aplicar el filtro para quitar el ruido

Ruido Gaussiano.

Abrimos una imagen y descubrimos que tiene ruido de tipo Gaussiano.



Ilustración 37. Imagen con ruido Gaussiano

Abrimos el menú de preprocesado y aplicamos el filtro correspondiente.

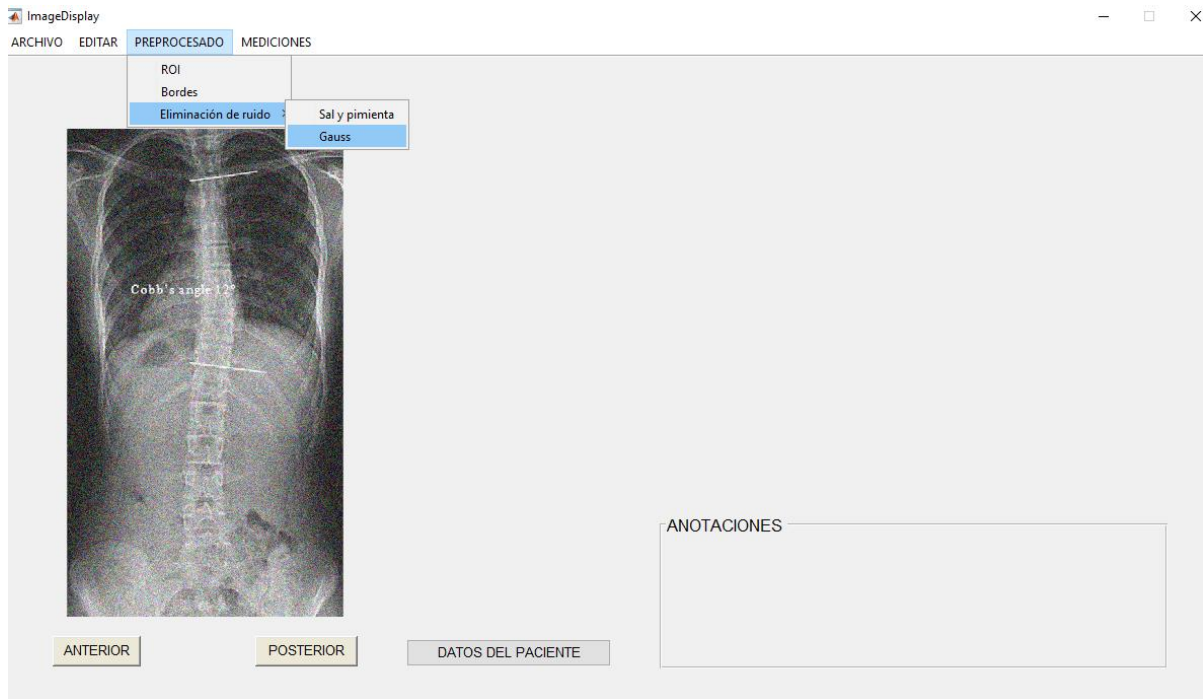


Ilustración 38. Recorrido en el menú para llegar al filtro para eliminar el ruido

Vemos que la imagen ha mejorado considerablemente.

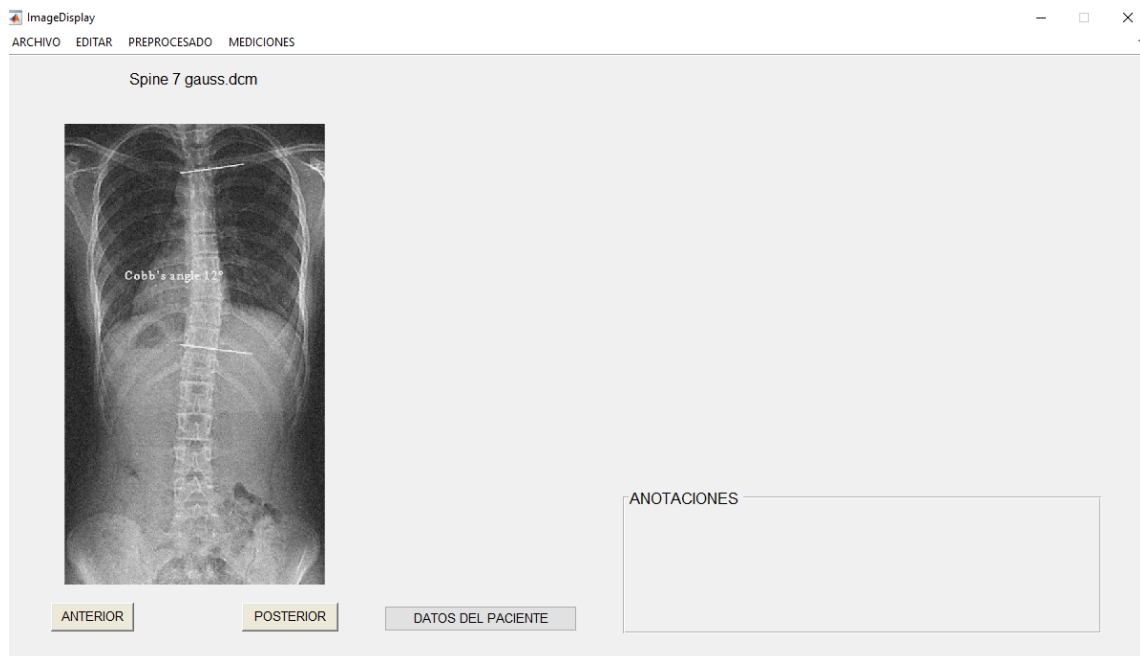


Ilustración 39. Resultado después de aplicar el filtro contra el ruido Gaussiano

4.3.2 HISTOGRAMA

Un histograma es una representación gráfica de una variable en forma de barras. La superficie de cada barra es proporcional a la frecuencia de los valores que se representan. [39]

El histograma de una imagen representa las frecuencias relativas de los niveles de gris de ésta. La probabilidad de que ocurra un determinado valor es:

$$P(g) = \frac{N(g)}{M} \quad (6)$$

En esta ecuación, M es el número de píxeles de la imagen con la que estamos trabajando y N(g) es el número de píxeles que tienen el nivel de intensidad g.

4.3.2.1 Utilidad

Con los datos resultantes del histograma es mucho más fácil saber qué modificaciones hay que hacerle a la imagen para que se vea mejor.

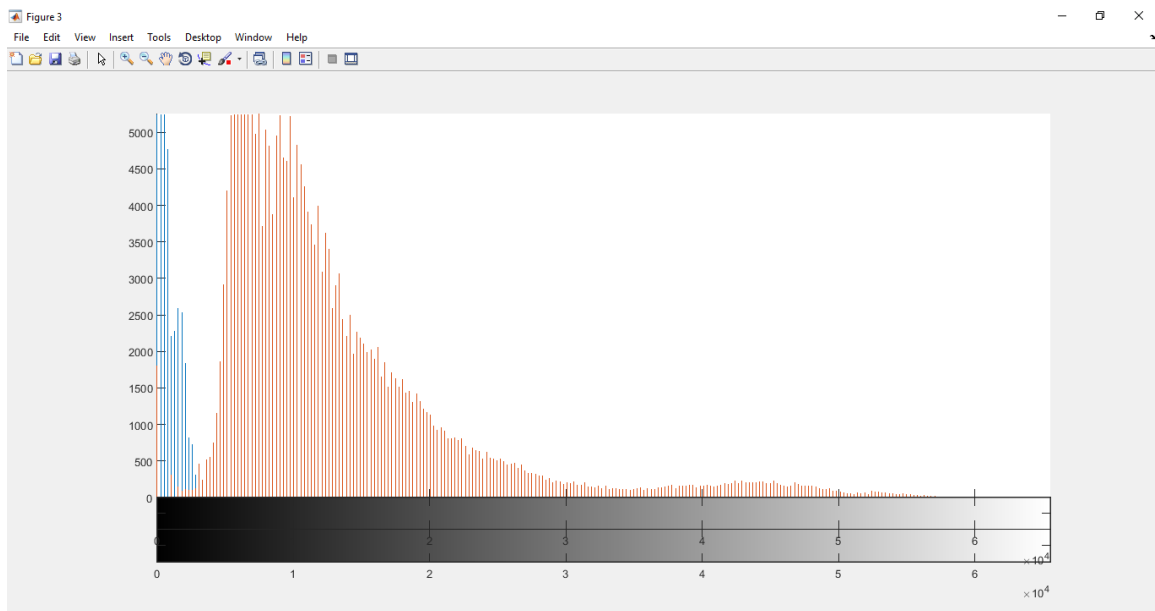


Ilustración 40. Histograma original y expandido de una imagen



Ilustración 41. Resultado después de expandir el histograma.

4.3.2.2 Implementación

```
imhist(a);  
axis off
```

Esta función calcula el histograma de la imagen de intensidad y lo dibuja.

4.3.2.3 Funcionamiento

En el menú de mediciones, se encuentra la función histograma que lo dibuja al lado de la imagen para que podamos evaluarlo.

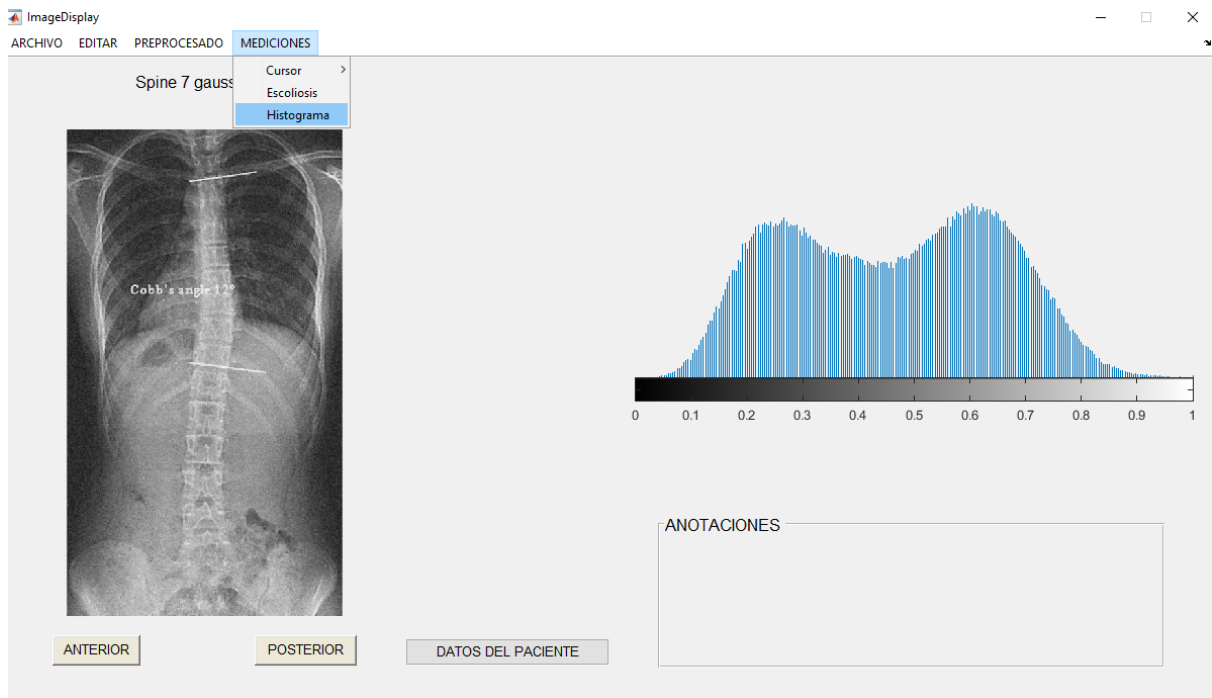


Ilustración 42. Resultado en la plataforma después de calcular el histograma

4.3.2 AJUSTES DE CONTRASTE DE LA IMAGEN

4.3.3.1 Utilidad

Esta función sirve para modificar la intensidad de la imagen para así facilitar su visualización. En imágenes de bajo contraste es necesario realizar un incremento de éste. En el caso de tener imágenes con ruido, es recomendable hacer antes un filtrado ya que, si hay píxeles de igual intensidad a los de ruido, puede que este incremento. [40]

4.3.3.1 Implementación

Ajuste automático

$$J = \text{imadjust}(I)$$

Esta función mapea los valores de intensidad en la imagen en escala de grises I a nuevos valores en J de manera que 1% de los datos se satura en intensidades bajas y altas de I . Esto aumenta el contraste de la imagen de salida J .

Ajuste manual

`g=imadjust (f, [low_in high_in], [low_out high_out], gamma)`

Esta función cambia la intensidad con respecto a unos valores de entrada (`low_in`, `high_in`) y unos de salida (`low_out`, `high_out`).

Cuando se omite `gamma`, éste toma el valor predeterminado 1. En este caso, el mapeo de la imagen es lineal. En el caso de que fuese menor que 1, provoca que la imagen sea más clara. Y con `gamma` mayor que 1 ocurre al contrario, la imagen se oscurece.

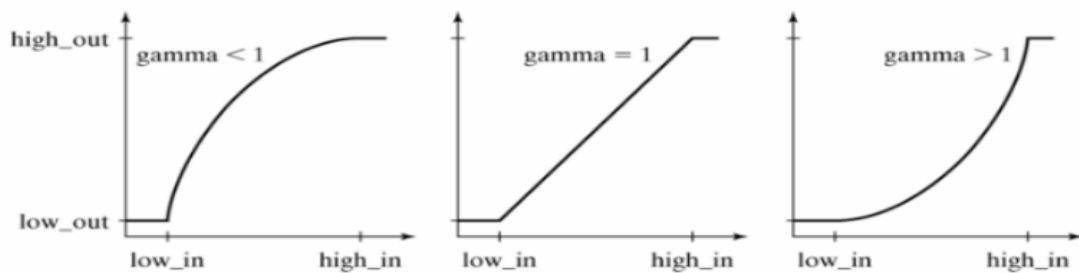


Ilustración 43. Funcionamiento del ajuste de contraste de la imagen

4.3.3.1 Funcionamiento

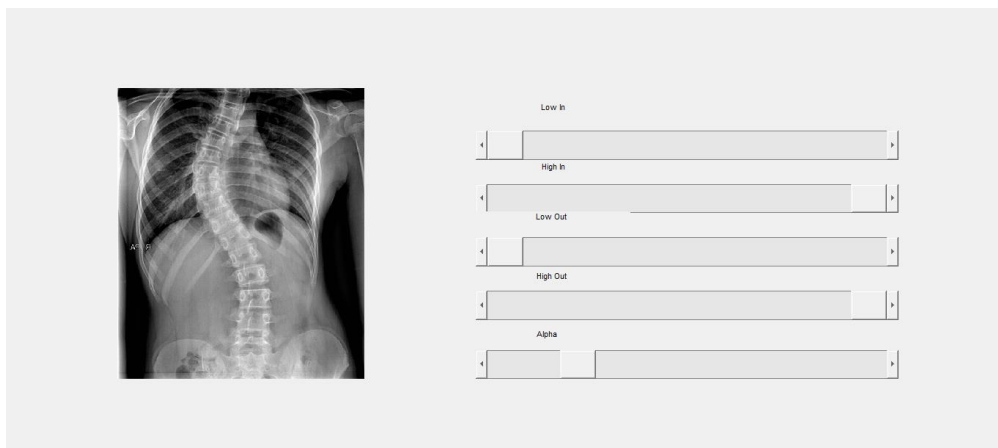


Ilustración 44. Imagen previa al ajuste de contraste

Al darle a esta opción se nos abre una nueva ventana donde podremos seleccionar los distintos valores que hemos visto en la función anteriormente.



Ilustración 45. Imagen después de realizar el contraste

4.4 OPCIONES DE MEJORA DE LA IMAGEN

La segmentación de imágenes tiene la ventaja de poder aislar regiones u objetos para facilitar su estudio. [42]

Los algoritmos de segmentación se basan en una de estas dos propiedades:

- Discontinuidad. Se buscan cambios bruscos en los niveles de gris.
- Similitud: se buscan zonas con valores similares siguiendo unos criterios.

Discontinuidad

Para detectar este tipo de discontinuidades se usa el Laplaciano, ya que la segunda derivada tiene una respuesta más fuerte ante detalles de la imagen.



Ilustración 46. Tipos de bordes

Hay dos tipos de bordes comunes:

- Borde ideal (a la izquierda de la imagen): de un píxel a su adyacente hay un gran cambio en el nivel de gris.
- Borde "rampa" (a la derecha de la imagen): un conjunto de píxeles adyacentes tiene cambios graduales en los niveles de gris.

Las características de los bordes son:

- Dirección del borde: perpendicular a la máxima variación de luminosidad.
- Intensidad: relacionado con el contraste a lo largo del borde.
- Posición dentro de la imagen.

Los pasos fundamentales en la detección de bordes son:

- Suavizado: eliminación del posible ruido que tenga la imagen.
- Mejora y procesado: realzado de los bordes de la imagen haciendo derivadas parciales y acabando en el gradiente (diferenciación).
- Umbralización: se determinan los píxeles que deben descartarse como ruido o como bordes.
- Localización: se determina la posición exacta del borde.

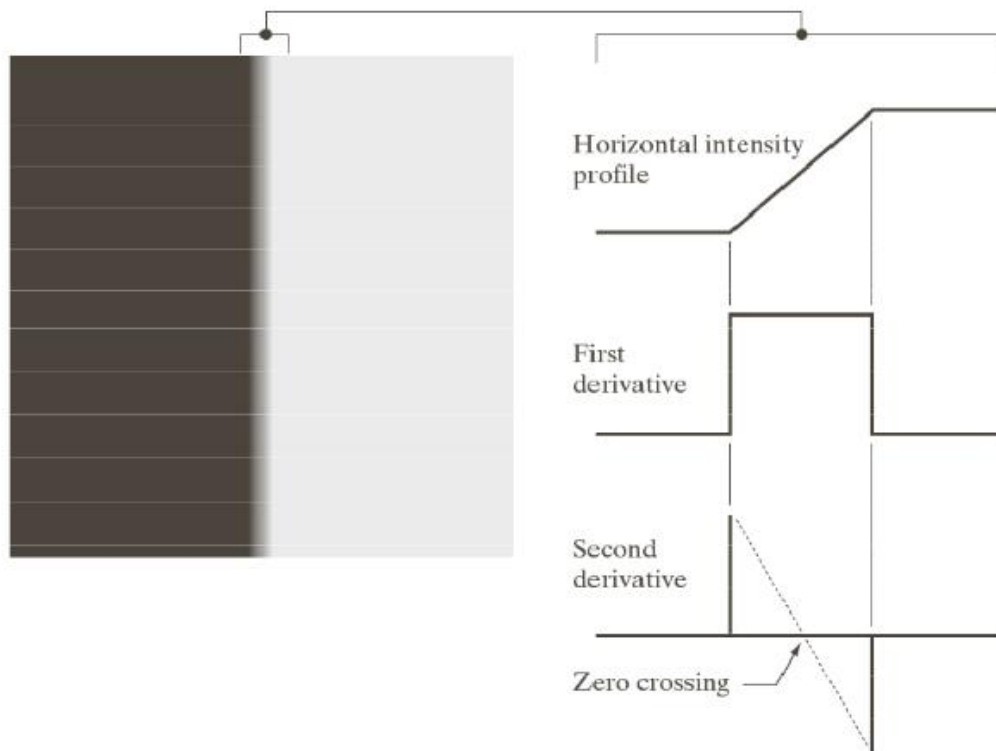


Ilustración 47. Uso de las derivadas para detectar bordes

Para la detección de bordes se hace uso de la primera derivada para ver los máximos/mínimos locales y de la segunda para la detección de cruces por cero. El signo de la segunda derivada nos dice en qué parte del borde se encuentra el píxel.

4.4.1 ROI (Region Of Interest)

Una región de interés es una parte de la imagen que se desea filtrar o sobre la que se desee realizar alguna otra operación.

Una ROI se define creando una máscara binaria, que es una imagen binaria del mismo tamaño que la imagen que se desea procesar. En la imagen de máscara, los píxeles que definen la región de interés se establecen en 1 y todos los demás píxeles se hacen 0.

4.4.1.1 Utilidad

Esta función es útil porque permite elegir una región de la imagen y eliminar el resto, lo cual permite ver algunas zonas con más detalle y aplicar otras funciones exclusivamente en esas zonas.

4.4.1.2 Implementación

```
global a;
[filas,columnas]=size(a);

Mascara = roipoly;

SelectedROI=zeros(filas,columnas);
for i=1:filas
    for j=1:columnas
        if Mascara(i,j)==1
            SelectedROI(i,j)=a(i,j);
        end
    end
end
end
```

`roipoly` especifica una región poligonal de interés dentro de una imagen. La función devuelve una imagen binaria que puede utilizarse como máscara para el filtro enmascarado. [41]

`Mascara = roipoly;` crea una herramienta de polígono interactivo, asociada con la imagen que se muestra en la figura actual. Con la herramienta activa, se especifica la sección seleccionando los vértices del polígono.

4.4.1.3 Funcionamiento

La función se encuentra dentro del menú de preprocesado.

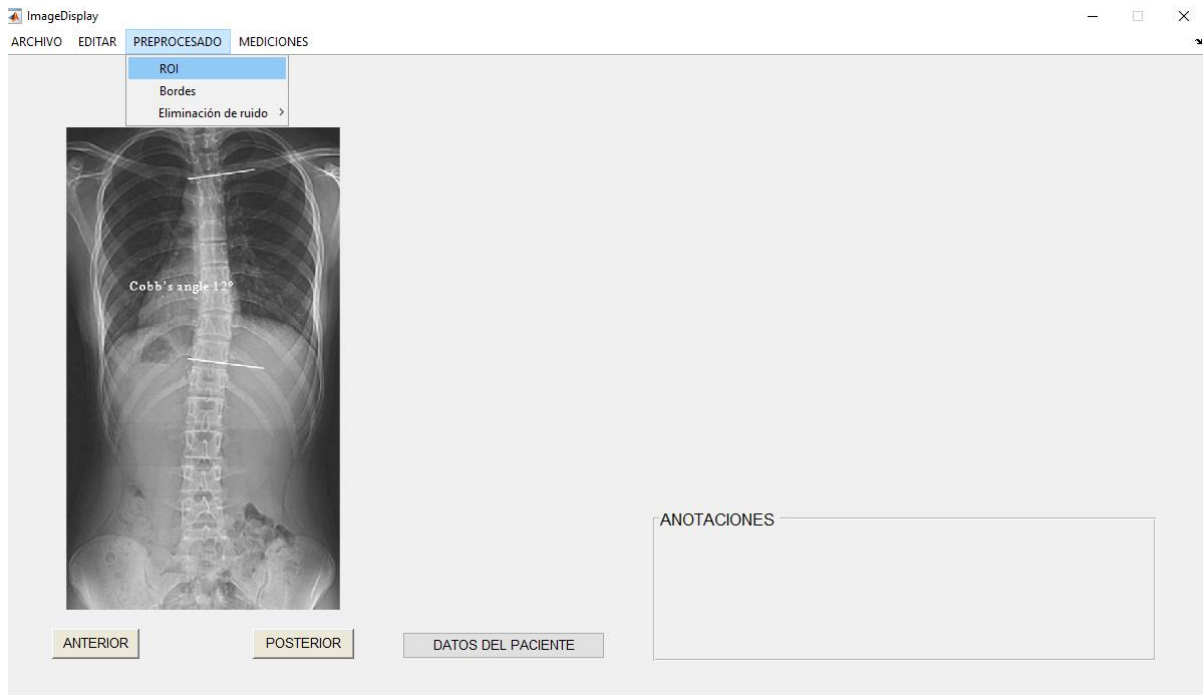


Ilustración 48. Navegación por menú para aplicar la ROI

Con el cursor vamos seleccionando los vértices del polígono que queremos dibujar para obtener la máscara.



Ilustración 49. Selección de la ROI

Una vez dibujada la forma, hacemos doble click sobre ella y se aplica el filtrado.



Ilustración 50. ROI de la imagen

4.4.2 BINARIZACIÓN

4.4.2.1 Utilidad

La binarización consiste en reducir la información de la imagen para que sólo haya dos valores posibles (1 o 0). Estos valores se corresponden con el blanco y el negro.

Mediante esta técnica es posible separar objetos o regiones del resto de la imagen.

4.4.2.2 Implementación

```
level = graythresh(I);
```

Este código calcula un umbral global que puede usarse para convertir una imagen de intensidad en una imagen binaria con:

```
BW = im2bw(I,level);
```

```
imshowpair(I,BW,'montage')
```

La función `graythresh` utiliza el método de Otsu, que elige el umbral para minimizar la varianza intraclase de los píxeles en blanco y negro.

Las matrices multidimensionales se convierten automáticamente en matrices 2D usando la transformación.

4.4.2.3 Funcionamiento

Este sería el resultado después de aplicar la binarización.

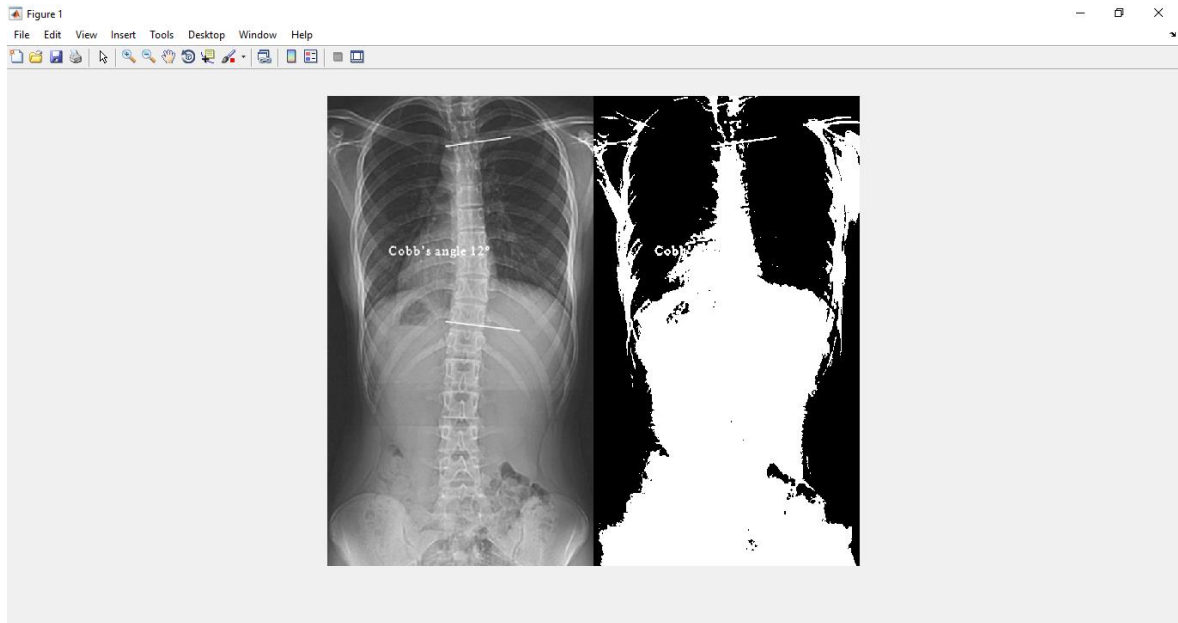


Ilustración 51. Imagen binarizada

4.4.3 BORDES

Aquí se encuentran distintas funciones cuyo objetivo es resaltar los bordes de las imágenes que estamos procesando. Gracias a estas funciones podremos ver de forma más clara el grado de desviación que tiene una columna, ya que no se confundirá con ninguna otra parte del cuerpo.

4.4.3.1 Utilidad

Esta función es útil ya que nos permite delimitar contornos de la imagen y ver mejor los detalles.

Para que un detector de bordes se considere óptimo debe cumplir estos puntos:

- Buena detección: debe marcar el mayor número real de bordes.
- Buena localización: los bordes detectados deben estar lo más cerca posible del borde de la imagen real.
- Respuesta mínima: no se deben crear falsos bordes.

4.4.3.2 Implementación

Los bordes se definen como variaciones fuertes de intensidad que indican la frontera de un objeto. Las técnicas de detección de bordes encuentran esas variaciones y las representan en otra imagen.

Hay distintos métodos de detección de bordes:

- Método Canny

Encuentra bordes a través del uso de la primera derivada de una gaussiana. Esta primera derivada sirve para tomar como cero todas las regiones donde no varía la intensidad y tiene un valor constante en las zonas que sí varía.

El algoritmo de Canny tiene tres pasos:

1. Se obtiene el gradiente: calculando la magnitud y orientación del vector gradiente en cada píxel.
2. Se hace más pequeño el ancho de los bordes localizados.
3. Reducción de contornos falsos: se aplica una función de histéresis basada en dos umbrales.

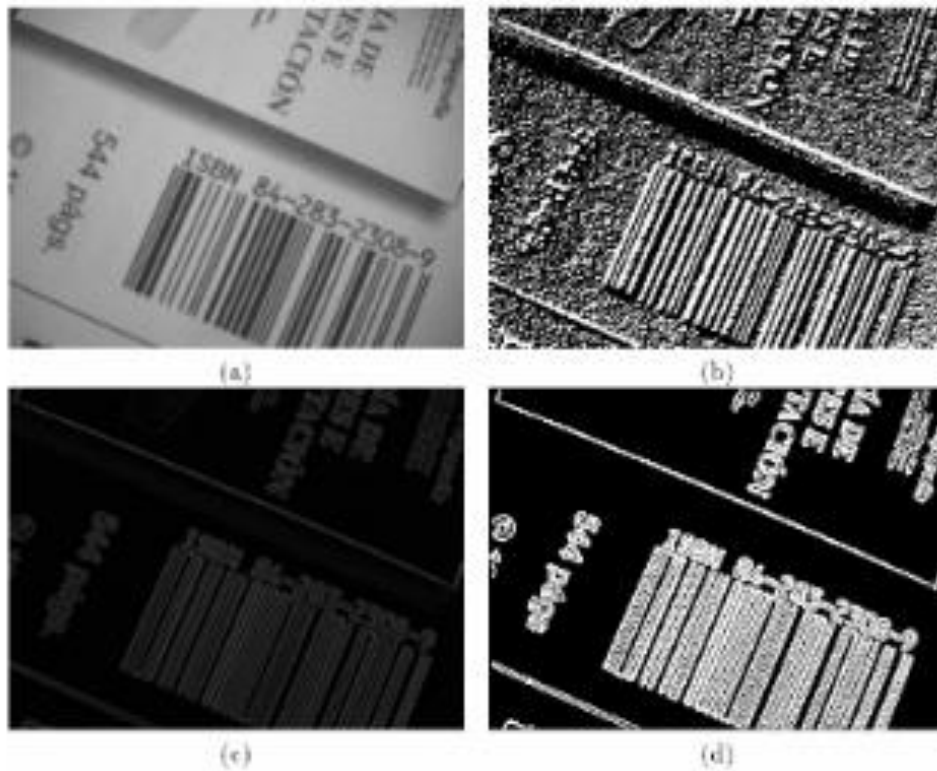


Ilustración 52. Resultado de aplicar el detector de bordes de Canny

- Método Sobel

Calcula el gradiente de la intensidad de una imagen en cada píxel. Para cada uno de los píxeles, se da la magnitud del mayor cambio posible, la dirección y el sentido. El resultado muestra cómo de grande es el cambio en cada punto y cómo de probable es que se presente un borde en esa imagen.

- Método Prewitt

Es un operador de diferenciación discreta, calculando una aproximación del gradiente de la función de intensidad de la imagen. En cada punto de la imagen, el resultado del operador de Prewitt es el vector de gradiente correspondiente o la norma de este vector.

Devuelve bordes en los puntos con máxima pendiente. Los dos operadores, ejes x e y en la dirección del eje y son:

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

- Método de gradiente

Se utiliza la operación del gradiente para la detección de grandes variaciones en la imagen. Tiene los siguientes pasos:

1. Suavizado: eliminación de ruido.
2. Mejora de la imagen realzando los bordes.
3. Umbralización: determina los píxeles que son ruido y los descarta.
4. Localización: determina la posición exacta del borde.

- Método de Roberts

Encuentra bordes usando la aproximación Roberts en el derivado. Devuelve bordes en aquellos puntos en los que la pendiente de que es máxima

4.4.3.3 Funcionamiento

Al seleccionar la opción "bordes" del menú nos sale esta imagen. Aquí se nos muestra la imagen original acompañada de varios resultados después de aplicar diferentes métodos de detección de bordes.

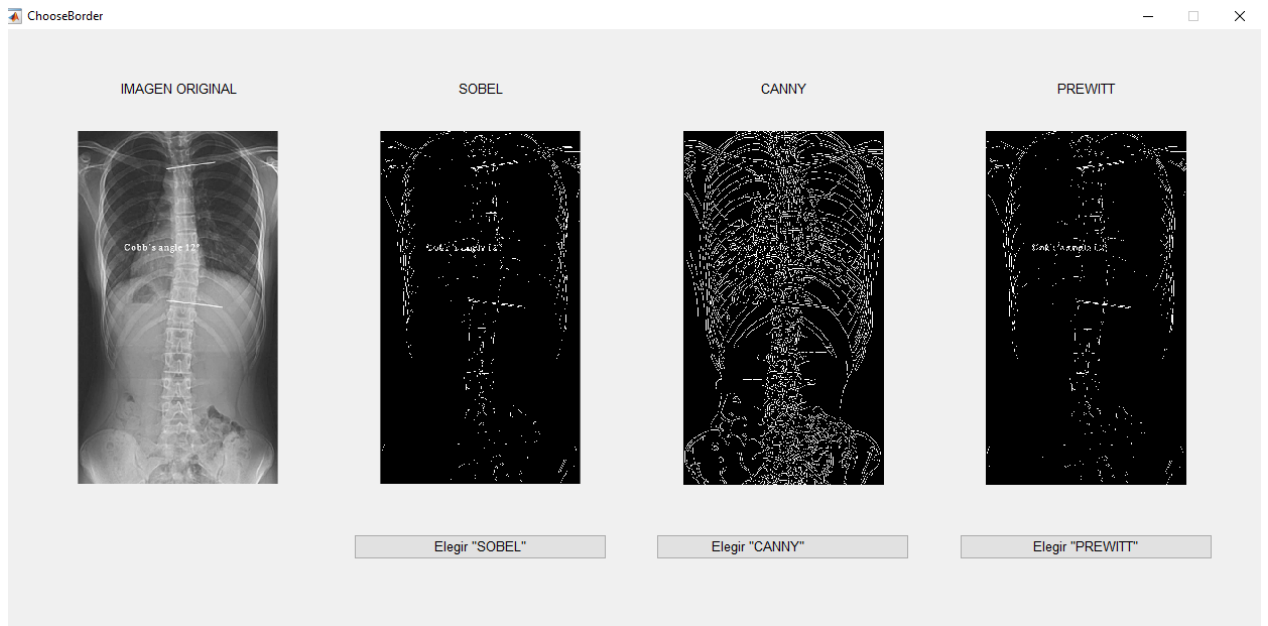


Ilustración 53. Diferentes resultados al aplicar los algoritmos de detección de bordes

Se selecciona la opción que más útil resulte para nuestro trabajo y podremos seguir trabajando sobre ella.

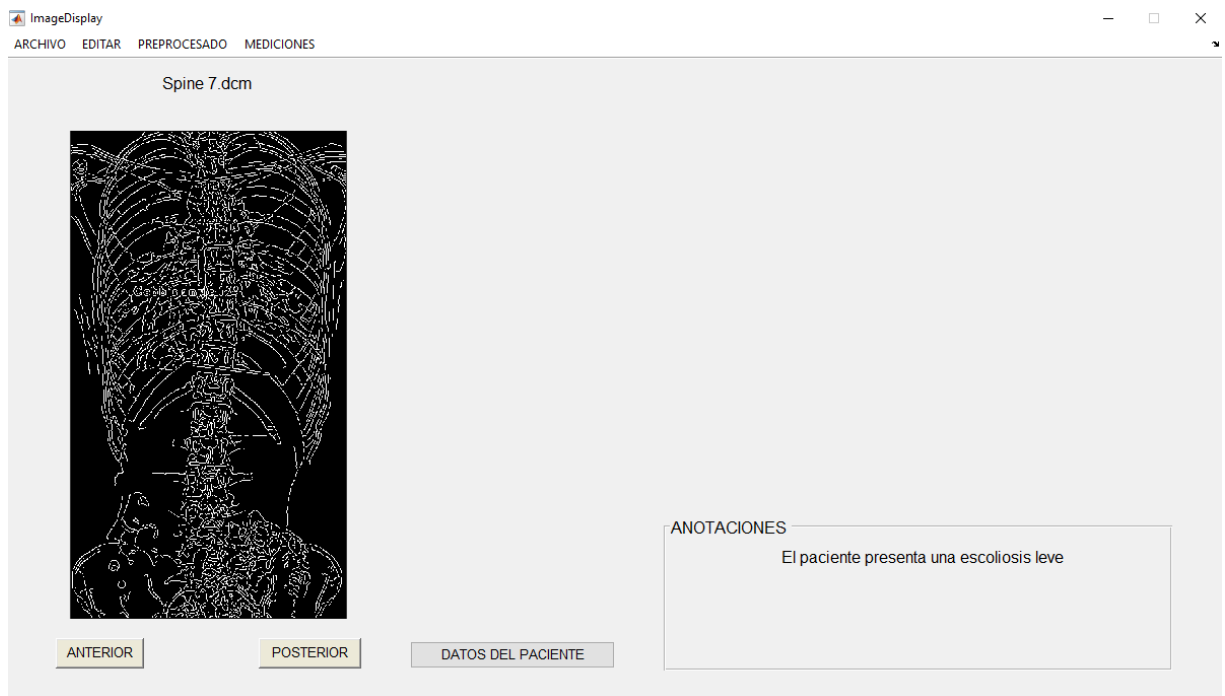


Ilustración 54. Elección del algoritmo de Canny para la detección de bordes

4.5 OPCIONES DE MEDICIÓN

4.5.1 CURSOR

4.5.1.1 Utilidad

Los cursores de datos le permiten leer datos directamente desde un gráfico mostrando los valores de los puntos que selecciona en líneas, superficies, imágenes, etc. Se pueden colocar múltiples cursores en una gráfica y moverlos interactivamente.

4.5.1.3 Funcionamiento

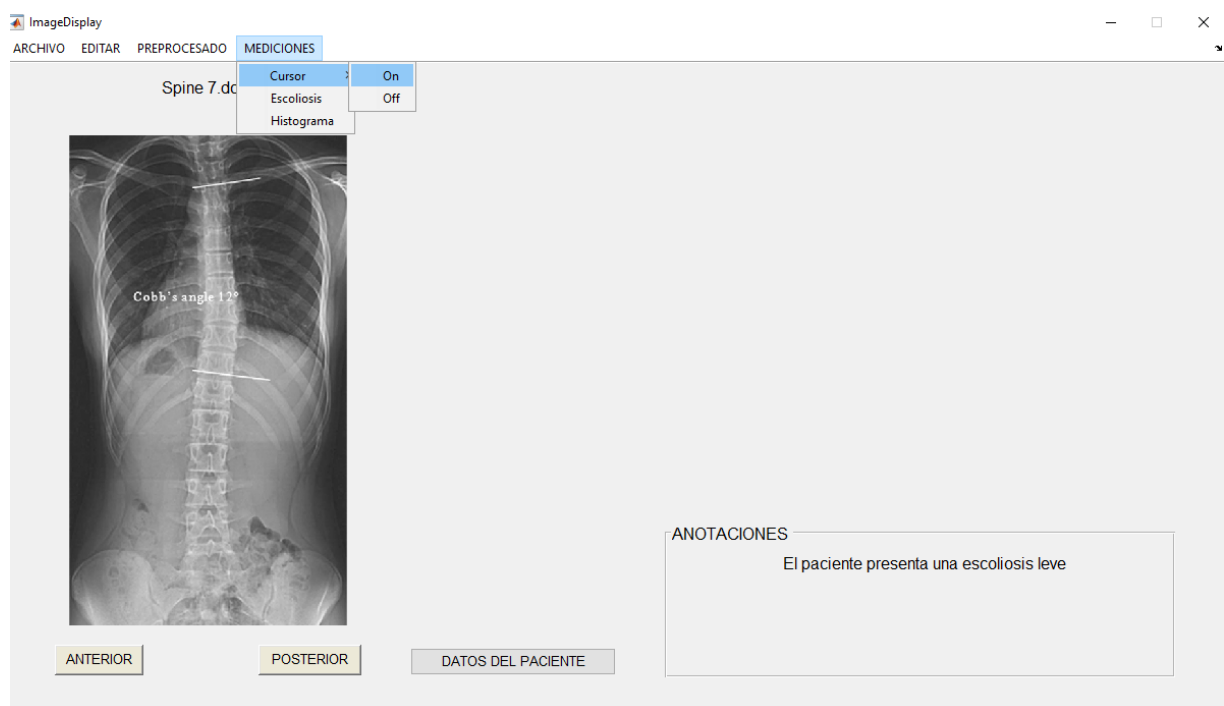


Ilustración 55. Navegación por el menú para encontrar la opción de cursor

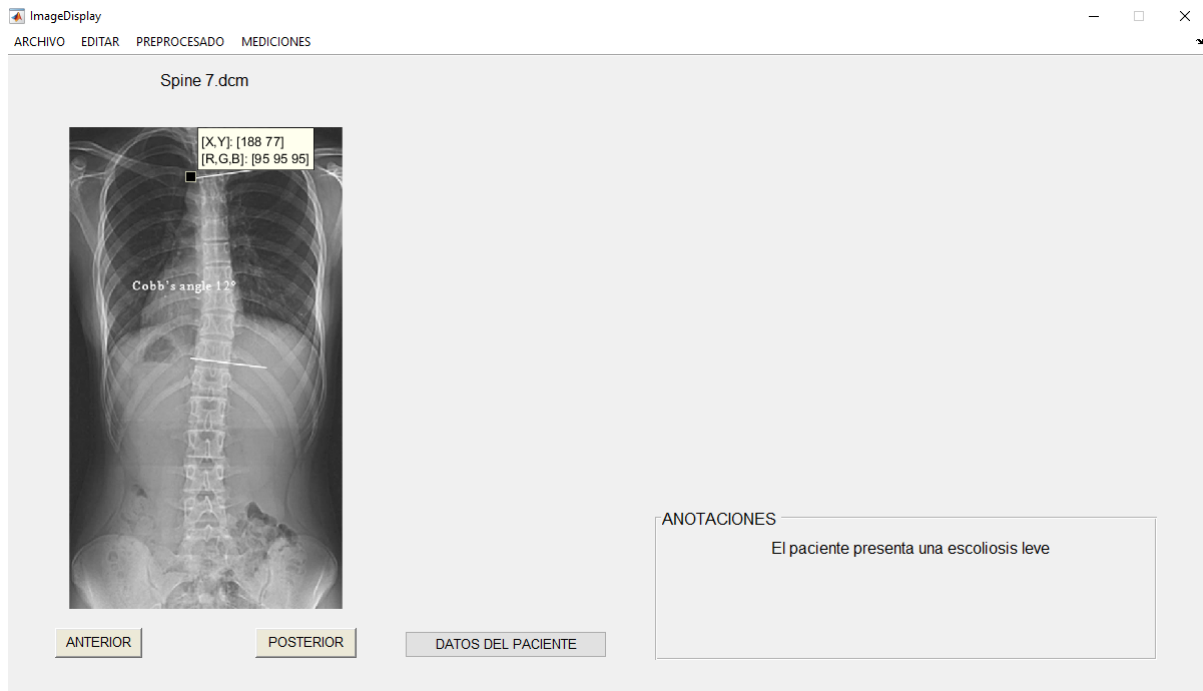


Ilustración 56. Hacemos click en el punto de la imagen donde nos interesa tener el dato

4.5.2 ESCOLIOSIS

4.5.2.1 Utilidad

En esta función se realizarán los cálculos que nos determinen el ángulo de Cobb. La importancia de este ángulo reside en que, al ser un valor numérico y no un dato subjetivo, nos permite observar progresivamente la evolución del paciente. Este método es el más usado para la medición de curvas tanto patológicas como fisiológicas. [43]

Hay dos formas de calcularlo:

- Directo: intersección de dos líneas paralelas a la parte de arriba de la vértebra superior involucrada y la parte de debajo de la vértebra inferior.
- Indirecto: intersección de dos líneas perpendiculares a las dos líneas del método directo.

Para medir este ángulo sobre una imagen médica, se toma de referencia la parte de arriba de la vértebra más inclinada en la zona superior de la curva y se dibuja una línea recta paralela a ella. Con la vértebra inferior de mayor inclinación se hace lo mismo en la parte de debajo de ésta. La intersección de éstas dos líneas nos proporciona el ángulo de

Cobb (cálculo directo). En el caso de que las líneas se salgan de la imagen, también se puede dibujar una perpendicular a cada línea y el ángulo que se forma en la intersección es el ángulo de Cobb (cálculo indirecto).

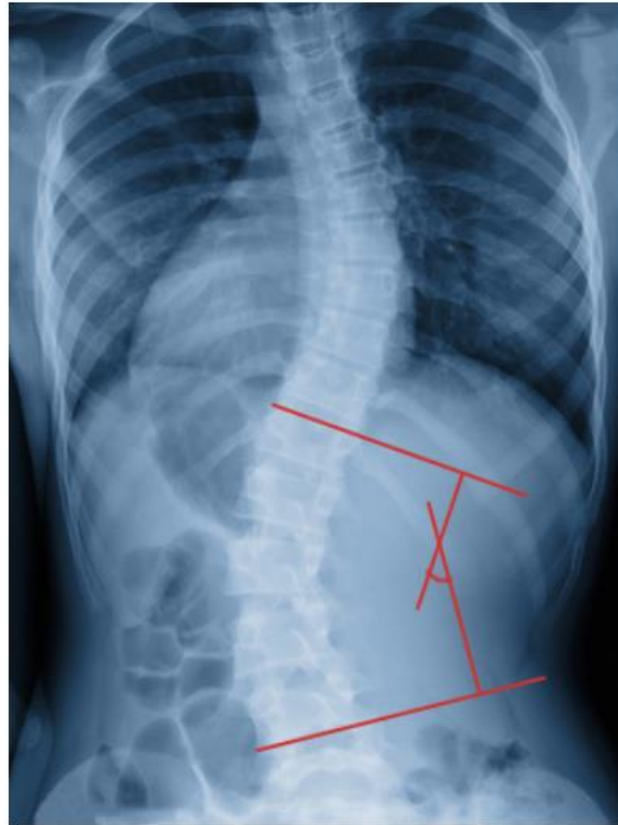


Ilustración 57. Medición del ángulo de Cobb

Este ángulo no es proporcional a la severidad de la escoliosis y la medición del ángulo tiene un error de $\pm 5^\circ$ al medirse sobre una radiografía en papel. Con esta función, éste error se reduce considerablemente.

Una curvatura con un ángulo menor a 10° se considera escoliosis y hasta 25° es leve. De 25° a 50° la escoliosis se considera moderada. Y si el ángulo es de más de 50° , será grave.

4.5.2.2 Implementación

Primero se dibujan las líneas paralelas a las vértebras. Para ello, utilizamos el algoritmo para calcular la ecuación de la recta. Obtenemos primero dos puntos por los que queremos que pase. Luego con la función line se unen los puntos.

```
% Se dibujan las lineas
[x1,y1] = Recta(a);
[x2,y2] = Recta(a);

[x,y]=ginput(2);

m = (y(2) - y(1)) / (x(2) - x(1)); %Pendiente

x(3)=500;
x=[x(1);x(2);x(3)];
x=x';

% Ecuacion de la recta que tiene una pendiente m y que pasa por
el punto A(x(1),y(1))
y(3)=(m*x(3)) - (m*x(1)) + y(1);
y=[y(1);y(2);y(3)];
y=y';

line(x,y)
```

Como vamos a calcular el ángulo de Cobb de forma indirecta, después se dibujan las perpendiculares a estas en el centro de las rectas para seguidamente calcular el ángulo que forman.

```

% Se calculan las perpendiculares
[r1, s1]=Perpendicular(x1,y1,1);
[r2, s2]=Perpendicular(x2,y2,2);

if n==1
    med1 = (diff(y)/diff(x));
    minv1 = -1/med1;
    minvabs1 = abs(minv1);
    r = [mean(x) mean(x)-200];
    s = [mean(y) mean(y)+200*minvabs1];
    line(r,s, 'Color', 'red');

else
    med2 = (diff(y)/diff(x));
    minv2 = -1/med2;
    minvabs2 = abs(minv2);
    r = [mean(x) mean(x)-200];
    s = [mean(y) mean(y)-200*minvabs2];
    line(r,s, 'Color', 'red');

end

```

Ahora se calcula el ángulo que forman ambas.

```

% Se calcula el ángulo entre las dos
perpendiculares (Cobb)
[Angle] = Angulo(r1, r2, s1, s2);

m1 = (s1(2) - s1(1)) / (r1(2) - r1(1));
%Pendiente 1
m2 = (s2(2) - s2(1)) / (r2(2) - r2(1));
%Pendiente 2

syms x % Variable simbolica
y1 = (m1*x) - (m1*r1(1)) + s1(1);
y2 = (m2*x) - (m2*r2(1)) + s2(1);

n1=[-m1,1];
n2=[-m2,1];
coseno=abs(dot(n1,n2)/(norm(n1)*norm(n2)));
Angle=acosd(coseno);

```

4.5.2.3 Funcionamiento

En el menú de mediciones seleccionamos la opción de escoliosis.

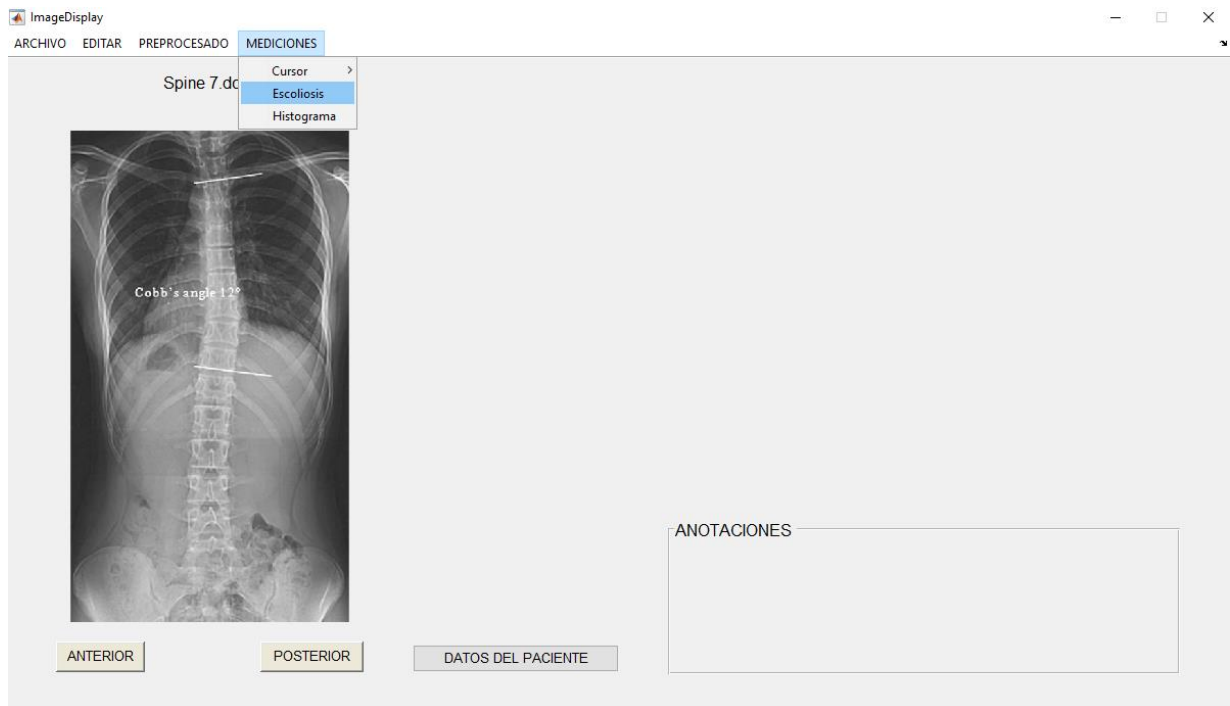


Ilustración 58. Menú para realizar las medidas de escoliosis

En la imagen, dibujamos las líneas necesarias para el cálculo del ángulo de Cobb.

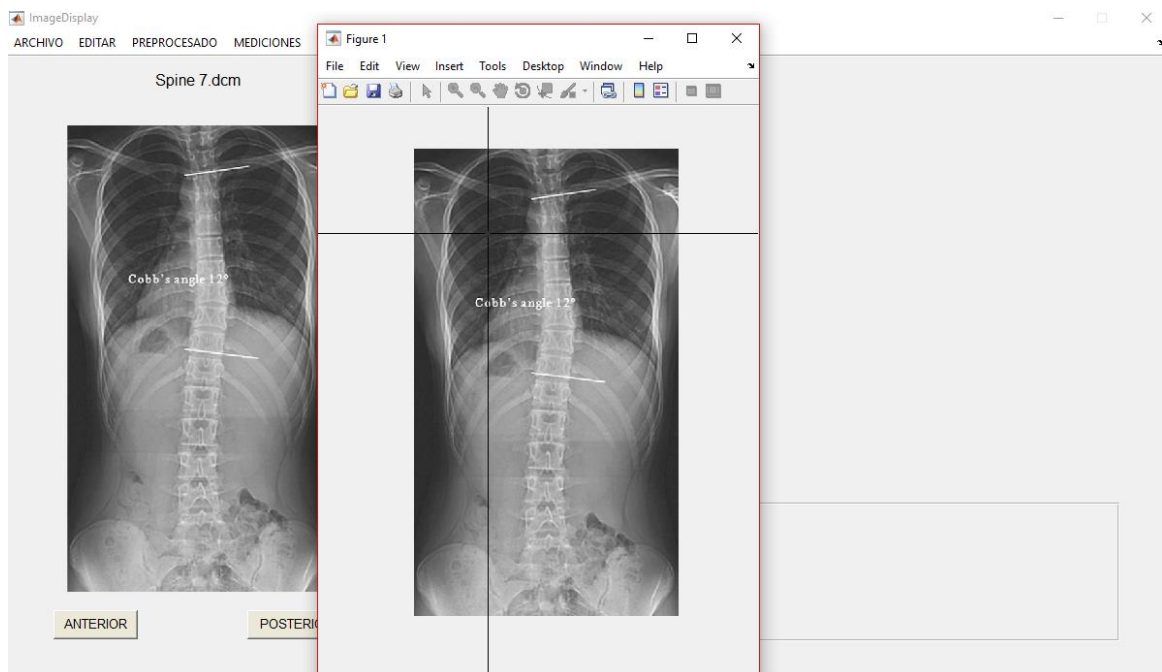


Ilustración 59. Dibujo de las líneas para el cálculo del ángulo de Cobb

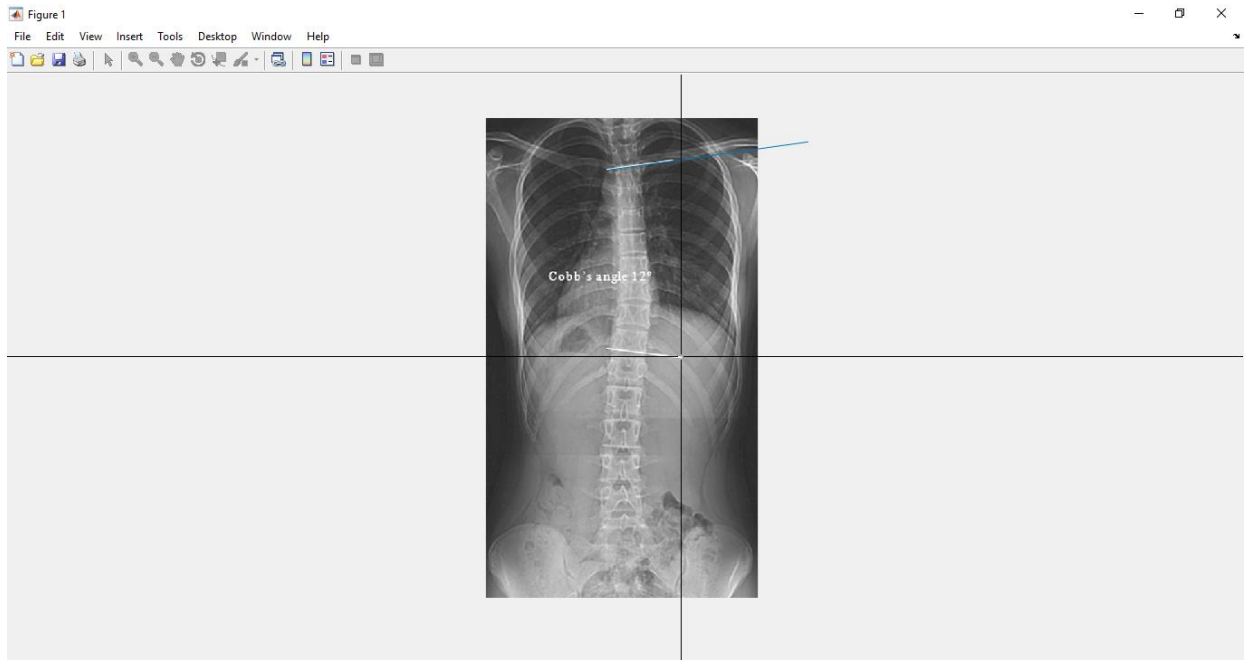


Ilustración 60. Dibujo de las líneas para el cálculo del ángulo de Cobb

Una vez dibujadas las dos líneas, nos saldrá dibujado el ángulo en la imagen junto con el punto donde cortan las dos líneas perpendiculares.

En la interfaz principal saldrá el tipo de escoliosis que se le ha diagnosticado al paciente.

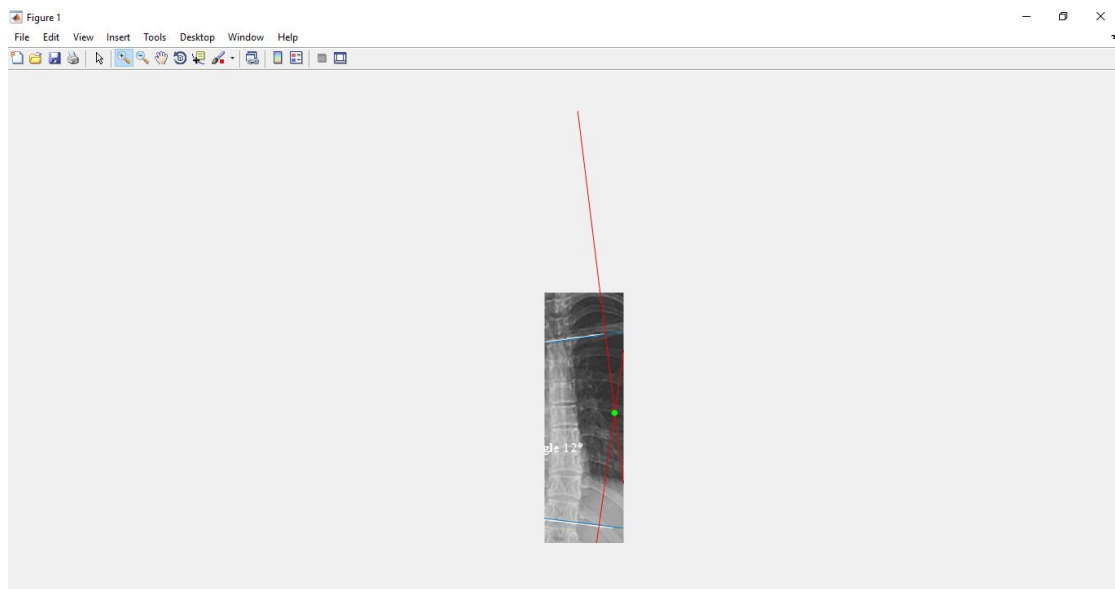


Ilustración 61. Representación del ángulo de Cobb

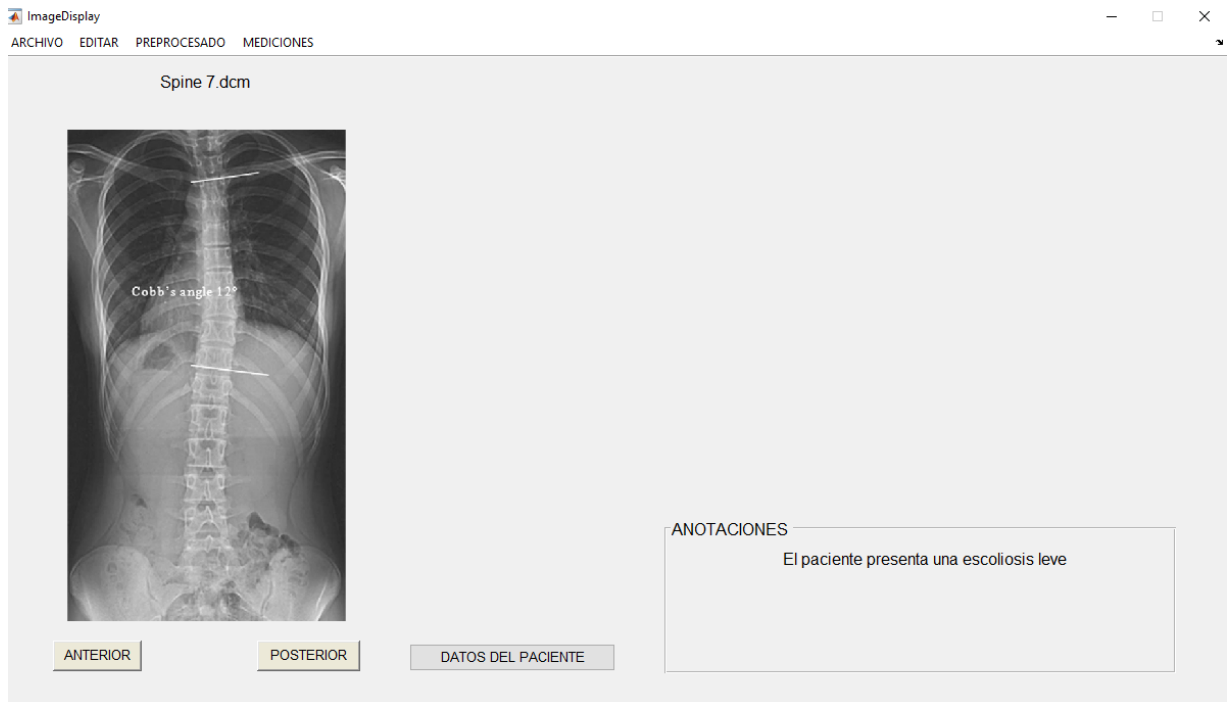


Ilustración 62. Anotaciones sobre el grado de escoliosis del paciente

4.5.4 DATOS DE PACIENTES

4.5.4.1 Utilidad

Esta función extrae los datos relevantes del paciente y los muestra en pantalla. Además, permite modificar esos datos o añadirlos en caso de que faltaran.

4.5.4.2 Implementación

Primero se hace una comprobación para ver si la imagen contiene algún dato que nos pueda interesar.

```
global info;

Nombre=isfield(info, 'PatientAddress');
ID=isfield(info, 'PatientID');
Edad=isfield(info, 'PatientAge');
Sexo=isfield(info, 'PatientSex');
Cumple=isfield(info, 'PatientBirthDate');
Peso=isfield(info, 'PatientWeight');
Altura=isfield(info, 'PatientSize');
Notas=isfield(info, 'PatientComments');
```

En el caso de que faltase algún dato, se pueden añadir de forma sencilla. Por ejemplo, en el caso del DNI del paciente, se nos abriría una pequeña ventana para insertar texto y al acabar de escribir el dato, se guardaría en su lugar correspondiente dentro de la estructura de datos de DICOM.

```
global info;
global userInput;
UserInput='';
waitfor(InsertText);
info.PatientID = UserInput;
save info;
handles = guidata(gcf);
set(handles.DNI, 'String', info.PatientID);
```

Al acabar de editar los datos, habría que guardar la imagen de nuevo ya que la estructura está protegida contra escritura

```
global Y;
global info;
global info2;
info2=info;
[nombre, directorio] = uiputfile('*.dcm',
'Guardar en');
nombre = fullfile(directorio,nombre);
dicomwrite(Y,nombre,info2,'CreateMode','copy',
'WritePrivate',1);
close Datos;
```

4.5.4.3 Funcionamiento

Vemos que no hay datos del paciente en este archivo.

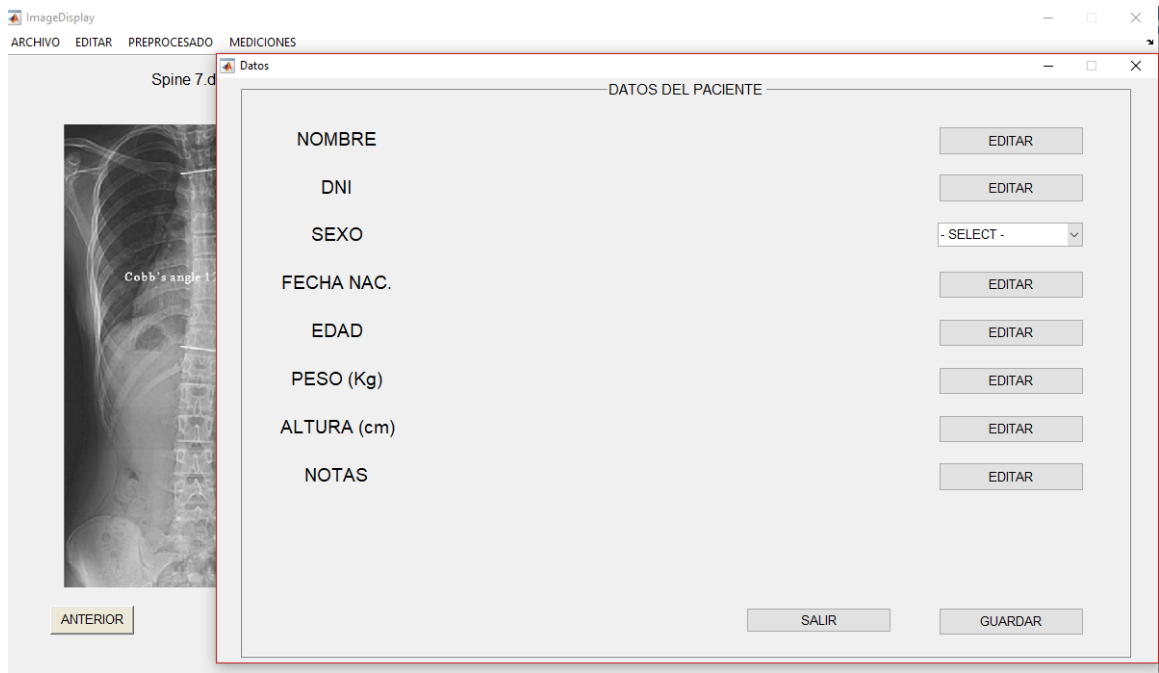


Ilustración 63. Interfaz para la visualización de los datos del paciente

Vamos a introducir el nombre.

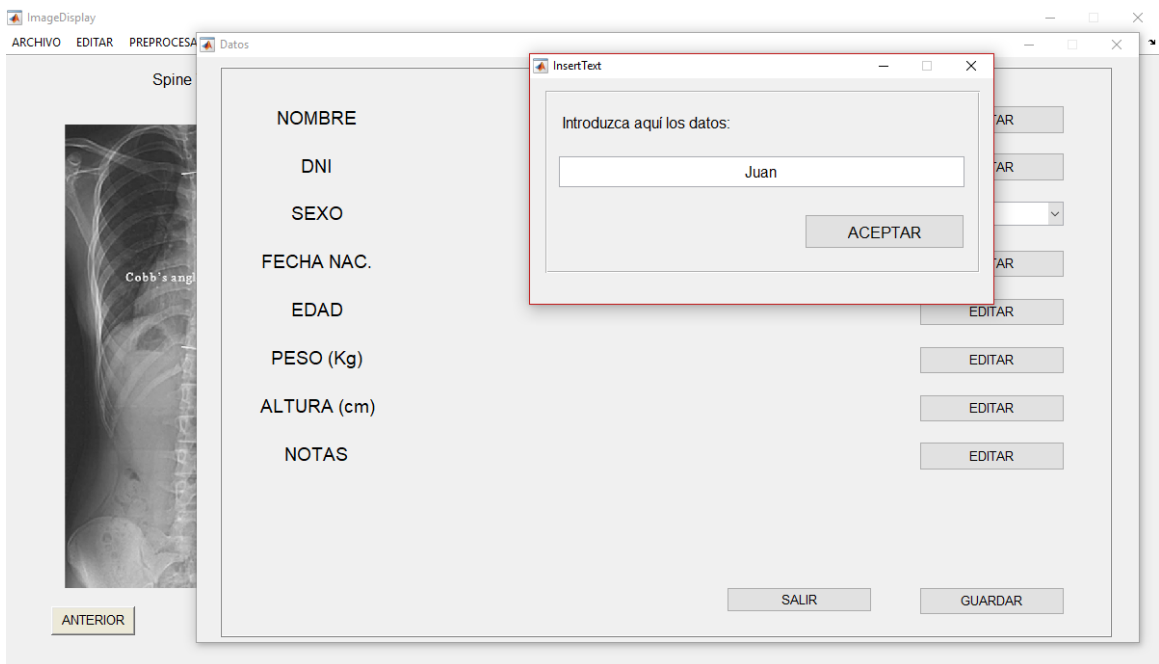


Ilustración 64. Introducción del nombre del paciente

Introducimos todos los datos necesarios.

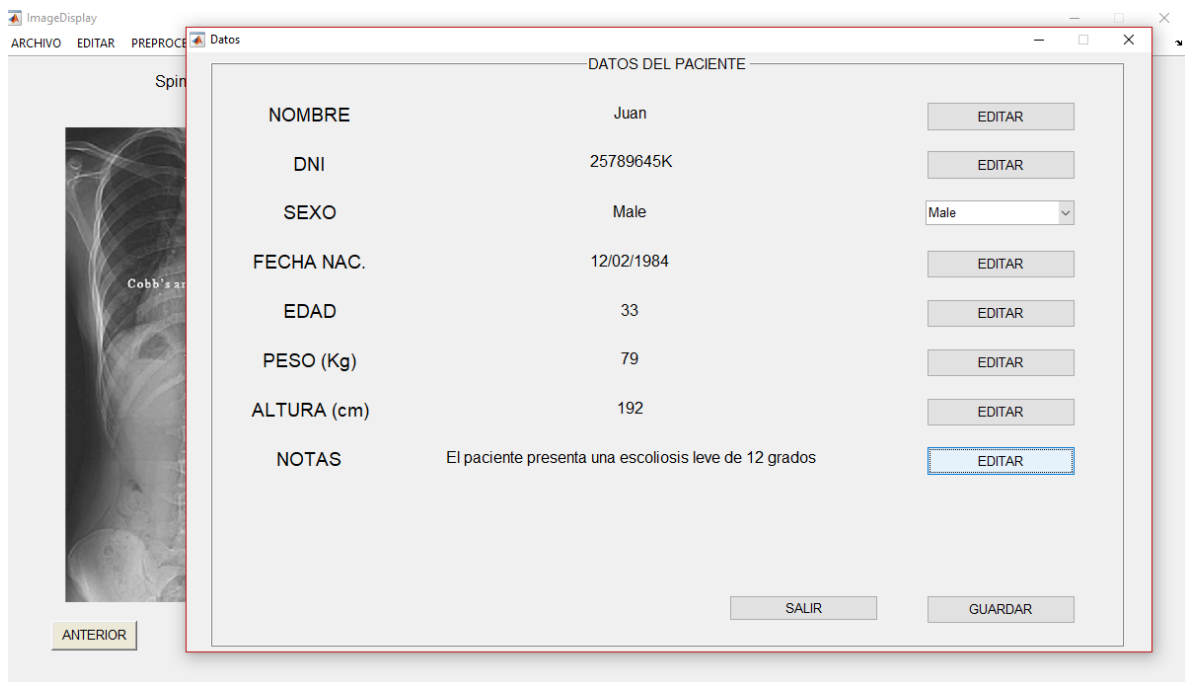


Ilustración 65. Datos del paciente ya introducidos

Una vez introducidos, pulsamos guardar para que se cree una nueva imagen con esta estructura.

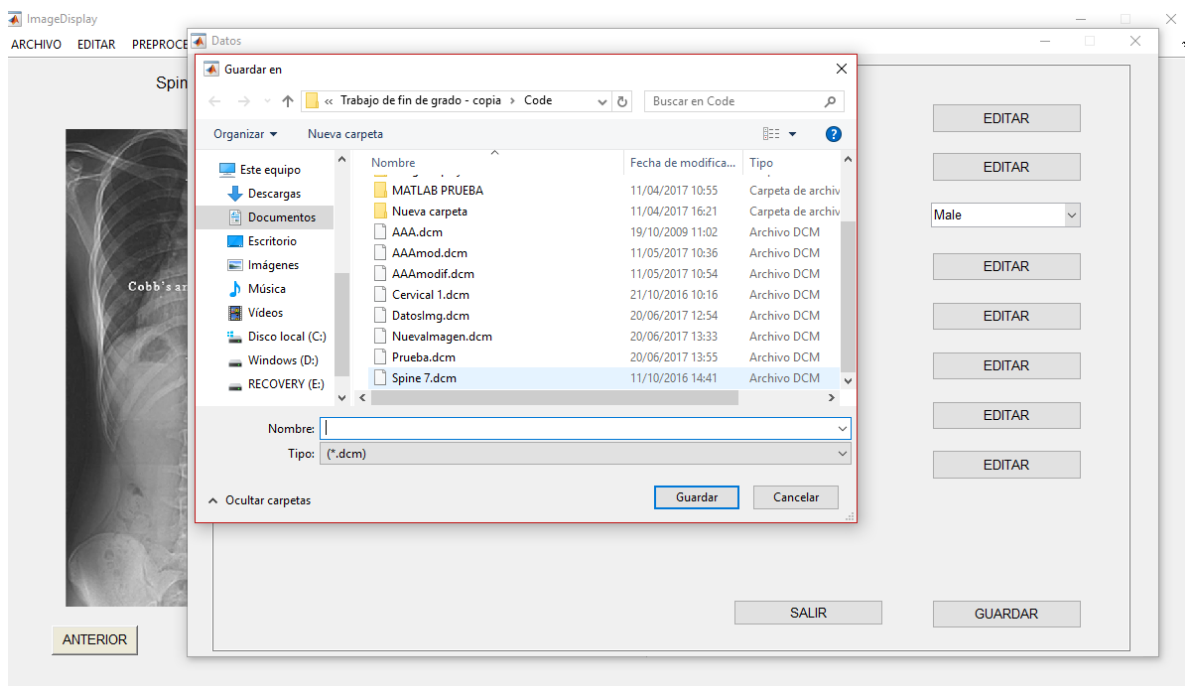


Ilustración 66. Guardar imagen después de añadir los datos

5. CONCLUSIONES

Este proyecto ha sido una introducción al procesado de la imagen médica como un apoyo al diagnóstico de enfermedades en la columna cervical.

Inicialmente se ha hecho una introducción al mundo de la imagen médica mostrando algunas aplicaciones ya existentes para el estudio de enfermedades.

Seguidamente se ha hecho un estudio fisiológico de la columna vertebral y se han explicado las enfermedades que le afectan.

A continuación, se ha introducido el concepto de imagen digital, se ha explicado el concepto de imagen, sus características, tipos y se han introducido las primeras líneas de código básicas para poder trabajar con este tipo de imágenes en Matlab, entorno en el que se ha desarrollado la aplicación.

Después se ha entrado en detalle con el formato DICOM, que es el usado para las imágenes médicas. Se han explicado también las distintas modalidades de imágenes médicas que existen.

Posteriormente, con la implementación del proyecto en Matlab, se ha descubierto que los resultados de esta aplicación no difieren en absoluto de los obtenidos con otras aplicaciones ya en el mercado. Además, la interfaz de este proyecto es mucho más fácil de usar y de entender con respecto a otros programas.

Otros programas del mercado no permiten el cambio de formato de la imagen, y aquí se puede hacer sin problema.

Las herramientas para la eliminación del ruido y la detección de bordes son especialmente útiles, ya que no he encontrado estas funciones en ninguno de los programas que he investigado. Lo mismo ocurre con la herramienta para seleccionar la región de interés en la imagen, si bien es verdad que ésta sí existe en algunos programas, en este proyecto es mucho más sencilla de usar.

La función de histograma resulta muy útil para la ecualización automática de la imagen. Ya que a veces la imagen es demasiado oscura o demasiado clara y es prácticamente imposible distinguir las vértebras.

Se puede llegar a la conclusión de que este proyecto es bastante completo, ya que aglutina todas las herramientas básicas que se necesitan para el procesado de las imágenes médicas y además, es muy sencillo de usar y entender.

6. LÍNEAS DE FUTURO

En futuras versiones de la aplicación, sería bastante útil tener en cuenta la siguiente hoja de ruta para obtener una interfaz completa y orientada a un mercado más amplio:

- La primera línea de acción a tomar sería ampliar el campo de estudio de la aplicación. Además del estudio de la desviación de la columna vertebral sería recomendable extender este estudio a otras enfermedades óseas. Incluso también incluyendo otros tejidos del cuerpo, como estudios sobre tumores o neurología.
- Al aumentar los campos de estudio, habría que aumentar también el número de funciones de procesado de imágenes, ya que las que se han explicado en este proyecto sólo están relacionadas con un campo de estudio.
- En el futuro también se podrían hacer algoritmos más óptimos para reducir el tiempo de carga de algunas funciones de procesamiento.
- Y el mayor reto es automatizar los algoritmos para que el usuario que utilice la aplicación no tenga por qué tener grandes conocimientos sobre medicina, sino que teniendo en cuenta la parte del cuerpo que se quiera estudiar, teniendo una imagen de esa parte y una ligera idea de la enfermedad que pueda tener según unos síntomas, se pueda hacer un diagnóstico aproximado de la enfermedad que tiene.

Teniendo la potencia de Matlab a la hora de procesar imágenes, no cabe duda de que una aplicación completa podría competir con grandes como RadiAnt u Osirix. Además, tiene la ventaja de ser completamente en español, en las otras aplicaciones estudiadas, el idioma común era el inglés y muy pocas tenían la interfaz y el manual de uso en nuestro idioma.

Bien es verdad, que para utilizar la aplicación es necesario contar con una licencia de Matlab y se perderían clientes por esto. Con lo cual, se puede pensar en implementar la aplicación en otros lenguajes de programación para hacer, por ejemplo, una aplicación web o incluso una adaptación para móviles o tablets.

7. BIBLIOGRAFÍA

INTRODUCCIÓN

[1] <http://definicion.de/bioingenieria/>

[2] <http://oa.upm.es/33072/>

[3] <http://bibing.us.es/proyectos/abreproy/11854/fichero/Volumen+1%252FCapitulo+2.pdf>

ANTECEDENTES

[4] <https://www.radiantviewer.com/es/7>

[5] <http://navegatum.com/es/dicom-viewer>

[6] <http://www.osirix-viewer.com/>

[7] <http://www.dicompyler.com/>

[8] <https://microdicom-viewer.programas-gratis.net/>

<http://www.microdicom.com/>

[9] <https://sourceforge.net/projects/opensdicomviewer/>

[10] <http://ginkgo-cadx.com/en/>

[11] <http://www.alma-medical.com/software/alma-mpr-reconstruccion-multiplanar/>

LA COLUMNA VERTEBRAL

[12] <http://columnavertebral.net/>

LESIONES Y ENFERMEDADES DE LA COLUMNA VERTEBRAL

[13] <http://columnavertebral.net/escoliosis/>

Elizabeth D Agabegi; Agabegi, Steven S. (2008). *Step-Up to Medicine (Step-Up Series)*. Hagerstwon, MD: Lippincott Williams & Wilkins.

LA IMAGEN DIGITAL

[14] <http://www.definicionabc.com/tecnologia/imagen-digital.php>

[15] <http://definicion.de/imagen-digital/>

[16] <http://definicion.de/pixel/>

[17] <http://platea.pntic.mec.es/~lgonzale/tic/imagen/conceptos.html>

[18] <http://imagenologia.robustiana.com/20-introducci%C3%B3n-a-la-imagen-digital-en-radiolog%C3%ADa>

IMÁGENES EN MATLAB

[19] <http://informatica.uv.es/iiguia/VC/tutorial.pdf>

[20] Apuntes de la asignatura Fundamentos de la imagen digital

FUNCIONES PARA EL TRATAMIENTO DE IMÁGENES

[21] <https://es.mathworks.com/help/matlab/ref/imread.html>

[22] <https://es.mathworks.com/help/images/ref/dicomread.html>

[23] <https://es.mathworks.com/help/matlab/ref/imwrite.html>

[24] <https://es.mathworks.com/help/images/ref/dicomwrite.html>

[25] <https://es.mathworks.com/help/images/ref/imshow.html>

[26] <https://es.mathworks.com/help/matlab/ref/image.html>

<https://es.mathworks.com/matlabcentral/answers/1288-what-is-the-difference-between-image-and-imshow>

[27] <https://es.mathworks.com/help/matlab/ref/size.html>

[28] <https://es.mathworks.com/help/images/ref/imcrop.html>

[29] <https://es.mathworks.com/help/images/ref/imresize.html>

[30] <https://es.mathworks.com/help/images/ref/gray2ind.html>

<https://es.mathworks.com/help/images/ref/im2bw.html>

<https://es.mathworks.com/help/images/ref/ind2gray.html>

<https://es.mathworks.com/help/images/ref/mat2gray.html>

<https://es.mathworks.com/help/matlab/ref/rgb2gray.html>

<https://es.mathworks.com/help/matlab/ref/rgb2ind.html>

DICOM

[31] <http://dicom.nema.org/standard.html>

CARGA Y VISUALIZACIÓN DE IMÁGENES

[32] <https://es.mathworks.com/help/matlab/ref/getframe.html>

OPCIONES DE EDICIÓN BÁSICAS

[33] <https://es.mathworks.com/help/images/ref/imrect.html>

[34] https://es.mathworks.com/help/matlab/ref/figure-properties.html#property_d0e272791

[35] <https://es.mathworks.com/help/matlab/ref/zoom.html>

[36] <https://es.mathworks.com/help/matlab/ref/pan.html>

TRANSFORMACIONES GEOMÉTRICAS DE LA IMAGEN

[37]

<http://www.alammi.info/2congreso/memorias/Documentos/martes/TRANSFORMGEOMETRICAS.pdf>

RUIDO

[38] <https://es.mathworks.com/help/images/ref/medfilt2.html>

HISTOGRAMA

[39] <https://www.aec.es/web/guest/centro-conocimiento/histograma>

AJUSTES DE CONTRASTE DE LA IMAGEN

[40] <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/171/A9.pdf?sequence=9>

ROI

[41] <https://es.mathworks.com/help/images/ref/roipoly.html>

BORDES

[42] Apuntes de la asignatura Fundamentos de la imagen digital

<http://www.sc.ehu.es/ccwgrrom/transparencias/pdf-vision-1-transparencias/capitulo-6.pdf>

<http://alojamientos.us.es/gtocom/pid/tema4.pdf>

https://www.researchgate.net/profile/Jorge_Valverde-Rebaza/publication/267240432_Deteccion_de_bordes_mediante_el_algoritmo_de_Canny/links/548dd1ae0cf225bf66a5f636/Deteccion-de-bordes-mediante-el-algoritmo-de-Canny.pdf

ESCOLIOSIS

[43] <https://espaldaycuello.com/angulo-de-cobb-en-la-escoliosis/>

http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0717-93082009000300007