



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO  
GRAO EN ENXEÑARÍA INFORMÁTICA  
MENCIÓN EN COMPUTACIÓN

# **Aplicación de técnicas de Visión Artificial en la detección de estructuras anatómicas en imágenes médicas de rodilla**

**Estudiante:** Pablo Valeiro Pena

**Dirección:** Julián Alfonso Dorado de la Calle

A Coruña, febreiro de 2021.



*A mis queridos padres María y Joaquín, a mi hermano Xaquín, a mis abuelos y padrinos Manolo y Natalia, y a mi bisabuela Maruja, por ser como son y apoyarme en tantas ocasiones. A todos mis amigos, los que estaban antes y los que vinieron después, por haberme permitido compartir con ellos grandes momentos. Gracias por tanto, perdón por tan poco.*



### **Agradecimientos**

En primer lugar, me gustaría agradecer al tutor de este trabajo, Julián Dorado, por su labor a lo largo del desarrollo del trabajo, consiguiendo ponerse en contacto con el personal del CHUAC para poder obtener los recursos necesarios para el desarrollo, y prestándome ayuda, aportando consejos e ideas muy valiosos. También quiero aprovechar para agradecer al personal del CHUAC que facilitó los estudios médicos que se utilizaron en proyecto, y quiero destacar el gesto teniendo en cuenta la situación sanitaria que vivimos actualmente, y que hace que sea complicado para un profesional de la salud encontrar tiempo libre para realizar este tipo de tareas.

Por supuesto, quiero agradecer a mi familia todo el apoyo y cariño aportado, no sólo a lo largo del desarrollo del trabajo, sino durante toda la etapa universitaria. Sin su ayuda hubiera sido prácticamente imposible que yo hubiese logrado finalizar esta etapa de mi vida con éxito.

Por último, pero no por ello menos importante, quiero agradecer a todos los amigos que he conocido en estos años en la Facultad de Informática, que me han aportado valioso conocimiento y gran ayuda en muchas ocasiones.



## **Resumen**

El importante avance tecnológico experimentado en las últimas décadas ha supuesto la incorporación de la informática en la mayoría de ámbitos de la vida humana. El mundo de la medicina no es ajeno a todos estos avances, que han permitido apoyar al personal médico en la tarea de diagnóstico de enfermedades, patologías o lesiones.

En la actualidad existen diversos sistemas informáticos que permiten al personal sanitario diagnosticar de manera automática o con mayor exactitud que la que podría tener una evaluación humana. Estos sistemas mejoran la labor del personal sanitario, puesto que les proporciona un ahorro de tiempo y una mayor precisión en los diagnósticos.

El objetivo de este trabajo es implementar un sistema automático de cálculo de medidas en imagen médica obtenida a través de Tomografía Computarizada (TAC) para el apoyo al personal médico en la detección de lesiones en rodillas de pacientes. Para ello, se han diseñado, implementado y probado las siguientes funcionalidades :

- Un sistema de detección de estructuras óseas a través de técnicas de Visión Artificial, que es capaz de obtener sus puntos de interés y realizar automáticamente la medición de distancias entre regiones.
- Una interfaz gráfica que permita al usuario elegir las imágenes a procesar, visualizar resultados parciales y finales y realizar un refinado de las medidas obtenidas.

---

## Abstract

The great technological progress during last decades has caused the incorporation of computing in most of human life habits. Medicine world is not foreign to these progress, which have allowed to support medical staff in disease, pathologies or injuries diagnostic.

Nowadays there are many computer systems that allow medical staff to diagnose automatically or with higher accuracy than a human evaluation. These systems improve the medical staff work, since they provide them a saving of time and a higher accuracy in diagnostics.

This goal of this work is to develop an calculation of measurements in medical image obtained by Computerized Tomography (CT) automatic system for the support to the medical staff in the detection of injuries in patients knees. Because of this, they have been designed, developed, and tested the following functionalities:

- A bone structure detection system trough Computer Vision techniques, which is able to obtain their interesting points and perform automatically the measurements between regions.
- A Graphical User Interface (GUI) that allows user to choose images to process, to visualize results and to perform a enhancement of obtained measures.

### Palabras clave:

- TA-GT
- Básica Rotuliana
- Python
- Fémur
- Rótula
- Cóndilo
- Tuberosidad
- Tróclea
- Tibia
- Segmentación
- DICOM
- OpenCV
- Interfaz
- Medida





# Índice general

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Problemática . . . . .	1
1.2	Solución propuesta . . . . .	2
<b>2</b>	<b>Fundamentos</b>	<b>3</b>
2.1	Fundamentos tecnológicos . . . . .	3
2.1.1	DICOM . . . . .	3
2.1.1.1	Meta información de ficheros DICOM . . . . .	4
2.1.1.1.1	Tag Pixel Array . . . . .	4
2.1.1.1.2	Tag Window Center y tag Window Width . . . . .	4
2.1.1.1.3	Tag Rescale Slope y tag Rescale Intercept . . . . .	5
2.1.1.1.4	Tag Pixel Spacing . . . . .	6
2.1.2	Tecnologías utilizadas . . . . .	6
2.1.2.1	JSON . . . . .	6
2.1.2.2	Python . . . . .	6
2.1.2.2.1	Numpy . . . . .	7
2.1.2.2.2	OpenCV . . . . .	7
2.1.2.3	GTK . . . . .	7
2.2	Fundamentos médicos . . . . .	8
2.2.1	TAC . . . . .	8
2.2.2	Puntos de interés anatómicos . . . . .	8
2.2.2.1	Cóndilos femorales . . . . .	8
2.2.2.2	Tróclea femoral . . . . .	8
2.2.2.3	Rótula . . . . .	9
2.2.2.4	Tuberosidad anterior de la tibia . . . . .	9
2.2.2.5	Representación de los puntos de interés en las imágenes TAC	10
2.2.2.5.1	Imágenes TAC de sección fémur . . . . .	10

2.2.2.5.2	Imágenes TAC de sección tibia . . . . .	11
2.2.2.6	Medidas anatómicas . . . . .	11
2.2.2.6.1	TA-GT . . . . .	12
2.2.2.6.2	Básica Rotuliana . . . . .	13
<b>3</b>	<b>Estado del Arte</b>	<b>15</b>
3.1	RadiAnt DICOM Viewer . . . . .	15
3.2	PostDICOM . . . . .	15
3.3	MedDream DICOM Viewer . . . . .	16
3.4	Conclusiones . . . . .	16
<b>4</b>	<b>Metodología</b>	<b>17</b>
4.1	Iteraciones propuestas . . . . .	18
4.2	Planificación del proyecto . . . . .	18
<b>5</b>	<b>Desarrollo</b>	<b>21</b>
5.1	Iteración 0 . . . . .	21
5.1.1	Análisis del sistema . . . . .	21
5.1.2	Descomposición de las iteraciones . . . . .	22
5.1.3	Recursos empleados . . . . .	24
5.1.3.1	Formato de las imágenes . . . . .	24
5.2	Primera iteración . . . . .	26
5.2.1	Análisis . . . . .	26
5.2.1.1	Análisis de requisitos . . . . .	26
5.2.1.2	Casos de uso . . . . .	27
5.2.2	Diseño . . . . .	27
5.2.2.1	Entrada y salida de datos . . . . .	27
5.2.2.2	Estructura del código . . . . .	28
5.2.3	Implementación . . . . .	29
5.2.3.1	Módulo dicom-utills . . . . .	30
5.2.3.1.1	Transformación a la Escala de Unidades Hounsfield . . . . .	30
5.2.3.2	Módulo image-processing . . . . .	30
5.2.3.2.1	Problemática de la escala de Unidades Hounsfield . . . . .	30
5.2.3.2.2	Función <i>thresholdHUImage</i> . . . . .	31
5.2.3.3	Módulo interesting-points . . . . .	31
5.2.3.3.1	Obtención de mitades izquierda-derecha . . . . .	32
5.2.3.3.2	Obtención de fémur o rótula . . . . .	32
5.2.4	Modelo de datos . . . . .	33

5.2.5	Pruebas . . . . .	33
5.3	Segunda iteración . . . . .	33
5.3.1	Análisis . . . . .	33
5.3.1.1	Análisis de requisitos . . . . .	34
5.3.1.2	Casos de uso . . . . .	34
5.3.2	Diseño . . . . .	35
5.3.2.1	Entrada y salida de datos . . . . .	35
5.3.2.2	Estructura del código . . . . .	35
5.3.3	Implementación . . . . .	36
5.3.3.1	Módulo <i>image-processing</i> . . . . .	36
5.3.3.1.1	Problemática con la orientación de los huesos en las imágenes . . . . .	36
5.3.3.1.2	Función <i>rotateFemur</i> . . . . .	37
5.3.3.1.3	Funciones <i>rotateTibia</i> y <i>rotateRotula</i> . . . . .	38
5.3.3.1.4	Funciones <i>getDrawedTAGTImage</i> y <i>GetDrawedBRI-image</i> . . . . .	39
5.3.3.2	Módulo <i>interesting-points</i> . . . . .	39
5.3.3.2.1	Función <i>getPointsCondiles</i> . . . . .	39
5.3.3.2.2	Función <i>getPointsTrochlea</i> . . . . .	40
5.3.3.2.3	Función <i>getPointsTibia</i> . . . . .	40
5.3.3.2.4	Función <i>drawPoints</i> . . . . .	41
5.3.3.2.5	Función <i>getPointsTibia</i> . . . . .	41
5.3.3.3	Módulo <i>dicom-utils</i> . . . . .	41
5.3.3.4	Módulo <i>measures</i> . . . . .	41
5.3.3.4.1	Función <i>getBasRot</i> . . . . .	42
5.3.3.4.2	Función <i>getTA-GT</i> . . . . .	42
5.3.4	Pruebas . . . . .	43
5.4	Tercera iteración . . . . .	43
5.4.1	Análisis . . . . .	44
5.4.1.1	Análisis de requisitos . . . . .	44
5.4.1.2	Casos de uso . . . . .	44
5.4.2	Diseño . . . . .	45
5.4.2.1	Arquitectura diseñada . . . . .	45
5.4.2.2	Diseño de la interfaz . . . . .	46
5.4.3	Implementación . . . . .	48
5.4.3.1	Modelo . . . . .	49
5.4.3.2	Vista . . . . .	49

5.4.3.2.1	Ventana de inicio . . . . .	50
5.4.3.2.2	Carga de estudios . . . . .	50
5.4.3.2.3	Selección de imágenes . . . . .	50
5.4.3.2.4	Selección de medidas . . . . .	51
5.4.3.2.5	Resultado de la medida . . . . .	51
5.4.3.3	Controlador . . . . .	51
5.4.3.4	Exportación a formato <i>.PNG</i> de estudios DICOM . . . . .	51
5.4.4	Pruebas . . . . .	52
5.5	Cuarta iteración . . . . .	52
5.5.1	Análisis . . . . .	52
5.5.1.1	Análisis de requisitos . . . . .	52
5.5.1.2	Casos de uso . . . . .	53
5.5.2	Diseño . . . . .	54
5.5.2.1	Modelo de datos . . . . .	54
5.5.2.2	Persistencia de datos . . . . .	55
5.5.2.3	Arquitectura diseñada . . . . .	56
5.5.2.4	Diseño de la interfaz . . . . .	56
5.5.2.4.1	Añadir nuevo estudio . . . . .	56
5.5.2.4.2	Listado de estudios . . . . .	57
5.5.2.4.3	Detalles del estudio . . . . .	58
5.5.2.4.4	Resultado de la medida . . . . .	58
5.5.2.4.5	Refinado de medidas . . . . .	59
5.5.3	Implementación . . . . .	59
5.5.3.1	Modelo . . . . .	59
5.5.3.2	Controlador . . . . .	60
5.5.3.3	Vista . . . . .	61
5.5.3.3.1	Nuevo estudio médico . . . . .	61
5.5.3.3.2	Listado de estudios . . . . .	62
5.5.3.3.3	Detalles del estudio . . . . .	62
5.5.3.3.4	Resultado de la medida realizada . . . . .	63
5.5.3.3.5	Refinar medidas manualmente . . . . .	63
5.5.4	Pruebas . . . . .	64
<b>6</b>	<b>Conclusiones y líneas futuras de trabajo</b>	<b>65</b>
6.1	Conclusiones . . . . .	65
6.2	Líneas futuras de trabajo . . . . .	66

<b>A</b>	<b>Guía de instalación</b>	<b>71</b>
A.1	Instalación del sistema . . . . .	71
A.1.1	Instalación en Ubuntu . . . . .	71
A.1.2	Instalación en Windows . . . . .	71
A.1.2.1	. . . . .	71
A.1.2.2	Configurar variables de entorno . . . . .	72
A.1.2.3	Instalación de módulos con PIP . . . . .	72
A.2	Ejecución . . . . .	73
<b>B</b>	<b>Guía de uso</b>	<b>75</b>
B.1	Introducción . . . . .	75
B.2	Compatibilidad del sistema . . . . .	75
B.3	Flujo de trabajo . . . . .	75
B.3.1	Ventana de bienvenida . . . . .	76
B.3.2	Añadir nuevo estudio . . . . .	76
B.3.3	Selección de imágenes . . . . .	78
B.3.4	Detalles del estudio . . . . .	79
B.3.5	Selección de medidas . . . . .	79
B.3.6	Resultado de la medida . . . . .	81
B.3.7	Refinar medidas . . . . .	82
B.3.8	Listado de estudios . . . . .	83
	<b>Glosario</b>	<b>89</b>
	<b>Bibliografía</b>	<b>91</b>



# Índice de figuras

---

2.1	Contenido de un fichero DICOM . . . . .	4
2.2	Atributos de un fichero DICOM . . . . .	5
2.3	Rango de unidades Hounsfield, para cada tipo de tejido o sustancia . . . . .	6
2.4	Ubicación de los cóndilos femorales en el fémur . . . . .	8
2.5	Ubicación de la tróclea en la rodilla . . . . .	9
2.6	Representación de una tibia y de sus diferentes partes. . . . .	10
2.7	Imagen obtenida a través de una prueba TAC en una rodilla a la altura del fémur. . . . .	10
2.8	Imagen obtenida a través de una prueba TAC en una rodilla a la altura de la tibia, en la que ha sido coloreada la región de la TAT en verde. . . . .	11
2.9	Ejemplo de cálculo de la distancia TA-GT . . . . .	12
2.10	Ejemplo de cálculo de Básica Rotuliana . . . . .	13
4.1	Metodología incremental . . . . .	17
4.2	Diagrama de Gantt para la planificación del proyecto . . . . .	19
5.1	Diagrama informal que explica el flujo básico de ejecución y las funcionalidades principales del sistema propuesto. . . . .	22
5.2	Ejemplo de imagen DICOM perteneciente a la sección del fémur . . . . .	25
5.3	Ejemplo de imagen DICOM perteneciente a la sección de la tibia . . . . .	25
5.4	Casos de uso para el actor "Usuario" en la Iteración 1 . . . . .	28
5.5	Modelo de datos diseñado para el manejo de las diferentes imágenes médicas . . . . .	29
5.6	Ejemplo de ejecución de la función thresholdHUImage. A la izquierda, la imagen que se le pasa como entrada, y a la derecha, la imagen resultante del proceso de segmentación. . . . .	32
5.7	Casos de uso para el actor "Usuario" en la segunda iteración . . . . .	34
5.8	Modelo de datos diseñado para el manejo de las diferentes imágenes médicas, dónde se reflejan los cambios diseñados en esta iteración. . . . .	35
5.9	Ejemplo de rotación de un objeto con respecto a la horizontal de una imagen . . . . .	36



---

5.10	Ejemplo de uso de la función <i>minAreaRect</i> . En rojo, el rectángulo de área mínimo calculado por la función. . . . .	37
5.11	Ejemplo de uso de la función <i>rotateFemur</i> . A la izquierda, la imagen, cuya mitad derecha ha sido eliminada, antes de la ejecución de la función. A la derecha, el resultado de la ejecución de la función, con los contornos ya rotados.	38
5.12	Imagen obtenida tras la ejecución del paso 3 de la función <i>getPointsTrochlea</i> . . . . .	41
5.13	Diagrama de casos de uso para la tercera iteración . . . . .	45
5.14	Arquitectura diseñada . . . . .	46
5.15	Ventana de carga de un estudio DICOM . . . . .	47
5.16	Ventana de selección de imágenes . . . . .	47
5.17	Ventana de selección de opciones para la medida a realizar . . . . .	48
5.18	Ventana de visualización de los resultados de la medición . . . . .	48
5.19	Diagrama de flujo de las ventanas . . . . .	49
5.20	Casos de uso para la cuarta iteración . . . . .	54
5.21	Estructura diseñada para los objetos JSON . . . . .	55
5.22	Cambios realizados en la arquitectura del sistema . . . . .	56
5.23	Ventana de introducción de datos de un nuevo estudio . . . . .	57
5.24	Ventana de listado de estudios . . . . .	57
5.25	Ventana de detalles de estudio . . . . .	58
5.26	Ventana de resultado de la medida . . . . .	59
5.27	Ventana de refinado de la medida . . . . .	60
5.28	Diagrama de flujo de las ventanas del sistema . . . . .	61
5.29	Ejemplo de selección de línea en la ventana de refinado . . . . .	64
A.1	Configuración de las variables de entorno . . . . .	72
A.2	Comprobación de la versión de Python instalada . . . . .	73
B.1	Ventana de bienvenida . . . . .	76
B.2	Ventana de nuevo estudio . . . . .	77
B.3	Ventana de selección de imágenes . . . . .	78
B.4	Ventana de detalles del estudio . . . . .	80
B.5	Ventana de selección de detalles de la medida . . . . .	81
B.6	Ventana de resultado de la medida . . . . .	82
B.7	Ventana de refinado de la medida . . . . .	83
B.8	Ventana de refinado de la medida con una línea seleccionada . . . . .	84
B.9	Ventana de listado de estudios . . . . .	85
B.10	Ventana de listado de estudios tras haber seleccionado un elemento de la lista . . . . .	85
B.11	Ventana de confirmación de eliminación de un estudio . . . . .	86

# Índice de tablas

---

4.1	Planificación de las iteraciones . . . . .	19
5.1	Resultados del cálculo TA-GT automático frente al cálculo manual . . . . .	43



# Introducción

---

LA implementación de sistemas informáticos ubicados en el ámbito de la informática biomédica que asesoran, apoyan o complementan al personal sanitario en sus labores permite una optimización de su tiempo y un aumento en la precisión de los diagnósticos.

Estos sistemas se utilizan en una amplia lista de tareas diferentes: tratamiento de datos médicos, diagnósticos automáticos, cálculos de probabilidad de un paciente de sufrir una enfermedad en el futuro...

Entre las herramientas de diagnóstico automático se encuentran las herramientas de diagnóstico automático a través de imagen médica, que toman como entrada el resultado de una prueba médica como una radiografía, o un TAC, y, a través del análisis de características de la imagen, como la morfología de los elementos encontrados, colores y texturas, ubicación o distancia entre elementos devuelven un diagnóstico sobre el estado de salud del paciente.

## 1.1 Problemática

Entre los sistemas informáticos de apoyo al diagnóstico comentados anteriormente existen diversas plataformas que permiten el visionado de imágenes DICOM y la realización de medidas anatómicas sobre ellas para la realización de diagnósticos. Sin embargo, no existe una amplia oferta sistemas informáticos de uso libre que permitan realizar medidas de manera automática.

El cálculo de medidas anatómicas en imágenes médicas de rodilla es una técnica de diagnóstico que permite detectar lesiones en esta región de la anatomía humana. Dos de las principales medidas que se usan para esta detección de lesiones son las medidas TA-GT y Básica Rotuliana. Estas medidas permiten un diagnóstico fiable, pero su cálculo manual puede ser poco preciso y su resultado es el proceso de la ejecución de unos pasos idénticos para cada paciente. Por tanto, contar con una herramienta de cálculo automático permitiría al personal sanitario reducir el tiempo de diagnóstico necesario para la obtención del paciente y reduciría

considerablemente los errores de precisión que se pudieran cometer a lo largo del proceso.

## 1.2 Solución propuesta

Este trabajo propone un sistema informático que resuelva la problemática mencionada anteriormente con las siguientes características.

- Un proyecto software ubicado en el ámbito de la Visión Artificial encargado de realizar la segmentación de regiones, búsqueda de puntos de interés y cálculo de las medidas TA-GT y Básica Rotuliana de manera automática.
- Una interfaz gráfica que permita cargar las imágenes necesarias para el cálculo de las medidas TA-GT y Básica Rotuliana y el visionado de resultados del proceso realizado por el sistema.

# Fundamentos

---

**E**N este capítulo se describen los fundamentos tecnológicos y médicos en los que se basa el proyecto. En los fundamentos tecnológicos se comentan los detalles de interés sobre el estándar de imagen médica DICOM y los lenguajes de programación y librerías asociadas utilizadas, y en los fundamentos médicos se introducen todos los conceptos médicos, medidas anatómicas y puntos de interés anatómicos necesarios para comprender el alcance del problema.

## 2.1 Fundamentos tecnológicos

### 2.1.1 DICOM

DICOM (Digital Imaging and Communications in Medicine) es un estándar para el manejo, dibujado y transmisión de imágenes médicas. DICOM está implementado en casi cualquier aparato destinado a imagen médica y se está implementando de manera incremental en aparatos de otros dominios médicos como oftalmología y odontología [1].

Los ficheros DICOM (figura 2.1) contienen una parte de cabecera, una parte de meta información y una instancia única SOP. La cabecera está formada por un preámbulo de 128 bytes, seguido por los caracteres 'DICM' en mayúscula[2].

Tras la cabecera se sitúa la meta información. Esta parte sigue un formato de fichero con etiquetas. En su contenido está información acerca del fichero, del estudio al que pertenece y del paciente al que fue extraído. Para cumplir con la normativa de Protección de Datos, en este trabajo se han utilizado tan sólo imágenes cuya información sobre el paciente fue borrada previamente a la obtención de las mismas.

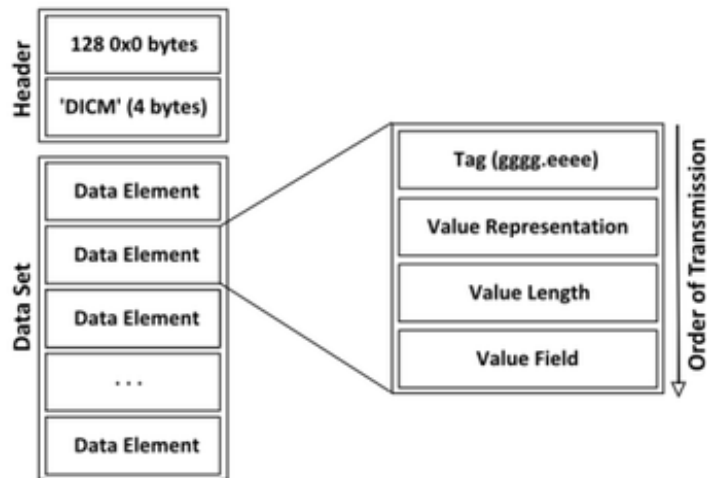


Figura 2.1: Contenido de un fichero DICOM

### 2.1.1.1 Meta información de ficheros DICOM

La meta información de los ficheros DICOM (figura 2.2) está compuesta por una serie de atributos. Estos atributos siguen una representación con las siguientes partes más importantes[3]:

- Una etiqueta ("tag") que identifica el atributo, normalmente en el formato (XXXX, XXXX) en codificación hexadecimal, que pueden ser divididos en el Número de Grupo DICOM y el Número de Elemento DICOM.
- Una Representación de Valor DICOM (VR) que describe el tipo de dato y el formato del valor del atributo.
- Una descripción del tag contenido.
- Un valor, del tipo perteneciente al valor almacenado en el VR.

#### 2.1.1.1.1 Tag Pixel Array

Este tag contiene los valores de los píxeles de la imagen. La representación de estos valores da como resultado la imagen médica. Este campo viene representado en forma de una matriz. Los valores de los píxeles de la imagen no pertenecen a una escala de grises, sino que son el resultado de la intensidad de la señal emitida en cada tipo de prueba médica, como las ondas ultrasonido en ecografías o la radiación en pruebas TAC.

#### 2.1.1.1.2 Tag Window Center y tag Window Width

Tag ID	VR	VM	Length	Description	Value
(0028,0004)	CS	1	12	Photometric Interpretation	MONOCHROME2
(0028,0010)	US	1	2	Rows	512
(0028,0011)	US	1	2	Columns	512
(0028,0030)	DS	2	18	Pixel Spacing	0.488281 0.488281
(0028,0100)	US	1	2	Bits Allocated	16
(0028,0101)	US	1	2	Bits Stored	16
(0028,0102)	US	1	2	High Bit	15
(0028,0103)	US	1	2	Pixel Representation	1
(0028,0120)	SS	1	2	Pixel Padding Value	-2000
(0028,1050)	DS	1	2	Window Center	40
(0028,1051)	DS	1	4	Window Width	400
(0028,1052)	DS	1	6	Rescale Intercept	-1024
(0028,1053)	DS	1	2	Rescale Slope	1
(0028,1054)	LO	1	2	Rescale Type	HU
(0040,0000)	UL	1	4	Group Length	66

Figura 2.2: Atributos de un fichero DICOM

El centro de ventana y ancho de ventana en una imagen DICOM son unos parámetros propios de este tipo de imágenes que pueden ser modificados para variar el rango de valores de píxel mostrados por pantalla. Este rango es de la forma  $[c-w/2, c+w/2]$ , siendo  $c$  el centro de la ventana, y  $w$  el ancho de ventana. Estos valores están dentro de los ficheros DICOM, en los tags WindowCenter y WindowWidth respectivamente, y pueden venir dados como un único valor numérico, o como un array de valores, siendo cada elemento del array un valor útil para mostrar por pantalla los diferentes tejidos encontrados en la imagen (e.g. un valor para mostrar tejidos blandos, y otro para mostrar tejido óseo).

### 2.1.1.1.3 Tag Rescale Slope y tag Rescale Intercept

Para poder reconocer y diferenciar los diferentes tejidos encontrados en la imagen, es necesario transformar los valores de la imagen contenidos en el tag Pixel Array a la escala de Unidades Hounsfield.

La escala Hounsfield [4] es una escala cuantitativa para describir la radiodensidad. La transformación a unidades Hounsfield (figura 2.3) viene dado por la fórmula  $HU = 1000 \times \frac{\mu_t - \mu_{agua}}{\mu_{agua}}$ , siendo  $\mu_{agua}$  el coeficiente de atenuación lineal del agua destilada, y  $\mu_t$  el coeficiente de atenuación lineal del tejido de interés.

Gracias a los tags Rescale Slope y Rescale Intercept es posible realizar una transformación lineal del valor contenido en el tag Pixel Array al valor de la escala Hounsfield usando la fórmula  $y = m \times x + b$ , siendo  $y$  el valor de salida en la escala Hounsfield,  $x$  el valor original para cada píxel de la imagen,  $m$  el valor contenido en el tag Rescale Slope, y  $b$  el valor contenido en el tag Rescale Intercept.



SUSTANCIA	UNIDADES HOUNSFIELD (NÚMEROS CT)
aire	-1000
grasa	-100
agua	0
líquido céfalo-raquídeo	15
sustancia blanca	20-30
sustancia gris	36-46
sangre	56-76
hueso	1000

Figura 2.3: Rango de unidades Hounsfield, para cada tipo de tejido o sustancia

#### 2.1.1.1.4 Tag Pixel Spacing

En este tag se almacena un array unidimensional de longitud dos que contiene la relación píxel-milímetro. Contiene un valor para la distancia en el eje  $x$ , y otro para la distancia en el eje  $y$ .

### 2.1.2 Tecnologías utilizadas

En esta sección se explican las tecnologías utilizadas en el desarrollo de este trabajo.

#### 2.1.2.1 JSON

JavaScript Object Notation es un formato ligero de intercambio de datos. Como formato, es de escritura y lectura sencilla para humanos, y de generación e interpretación sencilla para máquinas. Tiene un formato de texto independiente del lenguaje de programación pero que es ampliamente conocido por los programadores de lenguajes como C, C++, Java, Python, JavaScript, entre otros [5].

JSON está constituido por dos estructuras:

- Una colección de pares nombre/valor.
- Una lista ordenada de valores.

#### 2.1.2.2 Python

Python es un lenguaje orientado a objetos, interpretado, de alto nivel, y con semánticas dinámicas. Es un lenguaje atractivo para el Desarrollo Rápido de Aplicaciones [6]. Python

es simple, tiene una sintaxis sencilla y debido a eso reduce el coste de mantenimiento de un programa. Este lenguaje trabaja con módulos y paquetes, que permite la modularidad del código y la reusabilidad. El código fuente y los binarios de Python están disponibles de forma gratuita en la mayoría de plataformas, y se pueden distribuir de manera libre.

A continuación se enumeran los principales módulos de Python usados en el desarrollo del trabajo.

#### **2.1.2.2.1 Numpy**

Numpy es un módulo de utilidad científica para Python. Es una librería que provee de objetos de tipo array multidimensional, varios objetos derivados y un surtido de rutinas para operaciones rápidas sobre arrays, incluyendo matemáticas, lógicas, manipulación de formas, ordenación, etc. [7].

El núcleo de Numpy es el objeto ndarray. Este encapsula arrays n-dimensionales de tipos de datos homogéneos [7].

#### **2.1.2.2.2 OpenCV**

OpenCV (Open Source Computer Vision Library) es una librería de visión por computadora y machine learning. OpenCV fue construido para proveer una infraestructura común para aplicaciones de visión por computadora [8].

Cuenta con más de 2500 algoritmos optimizados. Estos algoritmos pueden ser usados para tareas como la detección de objetos, caras, etc.

Está disponible para los lenguajes de programación C++, Python, Java y MATLAB y tiene soporte para Windows, Linux, Android y Mac OS. Está en desarrollo la compatibilidad con CUDA y OpenCL.

#### **2.1.2.3 GTK**

GTK (conocido hasta febrero de 2019 como GTK+) o The GIMP Toolkit es una biblioteca de componentes gráficos multiplataforma para desarrollar interfaces de usuario (GUI) [9]. Forma parte del proyecto GNU y se utiliza en proyectos como GIMP, GNOME o Xfce entre otros.

Es uno de los kits de desarrollo de interfaces más populares para el sistema operativo GNU/Linux. Además, es compatible con otros sistemas Unix-like, y Microsoft Windows o Mac OS, entre otros.

GTK está escrito en C, pero está disponible en otros lenguajes de programación como C++, JavaScript o Python, permitiendo a los desarrolladores realizar aplicaciones GTK utilizando esos lenguajes[10].

## 2.2 Fundamentos médicos

### 2.2.1 TAC

La tomografía computarizada, comunmente conocida como TAC, es un examen médico de diagnóstico por imágenes. Al igual que los rayos X tradicionales, produce múltiples imágenes o fotografías del interior del cuerpo [11].

### 2.2.2 Puntos de interés anatómicos

En este apartado se explican todos los puntos anatómicos que son necesarios para el diagnóstico de las patologías en la rodilla que se tratan en la sección 2.2.2.6.

#### 2.2.2.1 Cóndilos femorales

Un cóndilo es la cabeza, eminencia o protuberancia rodeada en la extremidad de un hueso que encaja en el hueco de otro para formar una articulación[12].

El fémur posee dos cóndilos (figura 2.4) que, junto con la rodilla y la rótula, forma la articulación de la rodilla.

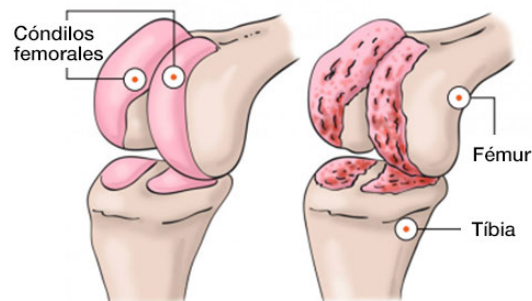


Figura 2.4: Ubicación de los cóndilos femorales en el fémur

Para el cálculo de las medidas, es necesario obtener los puntos que correspondan al borde posterior de cada cóndilo en la imagen TAC que contenga una sección del fémur.

#### 2.2.2.2 Tróclea femoral

La tróclea es una estructura ósea que forma parte del único hueso del brazo (húmero) y del hueso de la pierna (fémur)[13]. Tiene una particularidad con respecto al resto de estructuras óseas, y es que cuenta con forma de polea.

La tróclea femoral (figura 2.5) está situada en la zona interna de la superficie articular, justamente en la parte del interior del fémur. Esta se une a la tibia y a la rótula y forma la articulación de la rodilla.

Para el cálculo de medidas TA-GT, es necesario obtener el punto más profundo de la tróclea femoral.

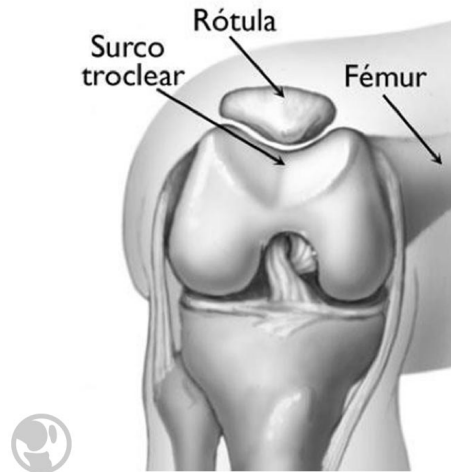


Figura 2.5: Ubicación de la tróclea en la rodilla

### 2.2.2.3 Rótula

La rótula es un hueso situado en el plano anterior de la rodilla, como se puede ver en la figura 2.5, engastada en el tendón anterior del cuádriceps. Su vértice se articula con el fémur. Tal y como se menciona en el apartado 2.2.2.2, forma la articulación de la rodilla junto con el fémur y la tibia.

Para el cálculo de la medida Básica Rotuliana, es necesario encontrar la recta que pase por los puntos pertenecientes al eje transversal de la rótula.

### 2.2.2.4 Tuberosidad anterior de la tibia

La tuberosidad anterior de la tibia (figura 2.6) es una elevación en la cara anterior de la tibia, justo debajo de los cóndilos del hueso de la tibia [14].

La tuberosidad anterior de la tibia forma parte de la estructura que sirve como palanca para extender la articulación de la rodilla y previene de colapsar a la rodilla cuando el pie golpea el suelo.

Para calcular las medidas TA-GT, es necesario encontrar el margen más saliente de la tuberosidad anterior de la tibia en la imagen.

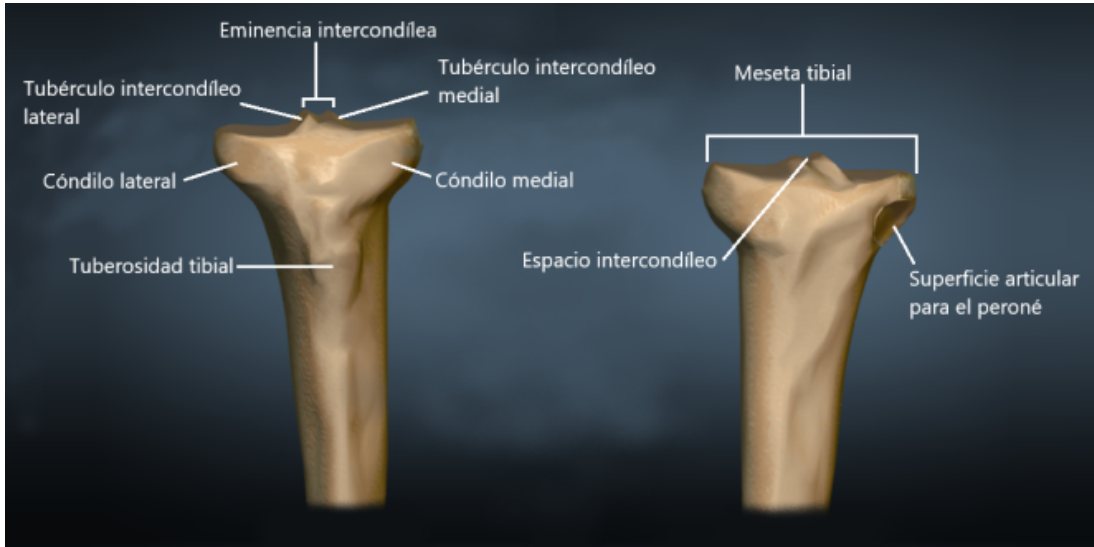


Figura 2.6: Representación de una tibia y de sus diferentes partes.

### 2.2.2.5 Representación de los puntos de interés en las imágenes TAC

En esta sección se señalan de manera gráfica todos los puntos de interés comentados en los apartados 2.2.2.1, 2.2.2.2, 2.2.2.3 y 2.2.2.4.

#### 2.2.2.5.1 Imágenes TAC de sección fémur

La representación de una imagen obtenida a través de una prueba TAC en una rodilla a la altura del fémur tiene una morfología similar a la expuesta en la figura 2.7.

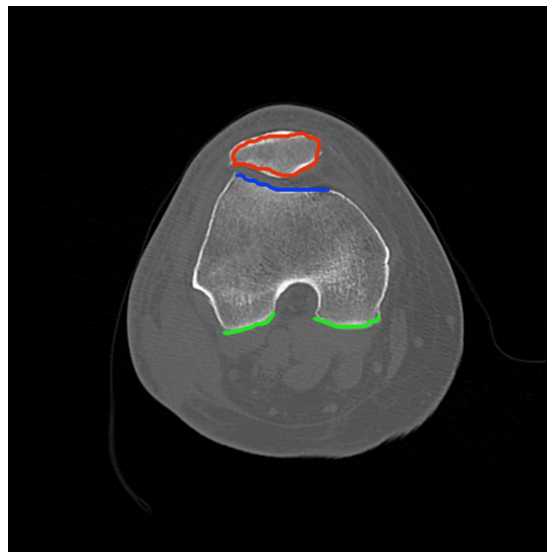


Figura 2.7: Imagen obtenida a través de una prueba TAC en una rodilla a la altura del fémur.

En esta imagen fueron señaladas a color ciertas regiones, con la siguiente explicación:

- Las regiones pintadas en color **verde** corresponden a las zonas pertenecientes a los **cóndilos femorales**.
- La región pintada en color **azul** corresponde a la zona perteneciente a la **tróclea femoral**.
- El área pintada en color **rojo** señala el área dónde se ubica la **rótula** en la imagen.

#### 2.2.2.5.2 Imágenes TAC de sección tibia

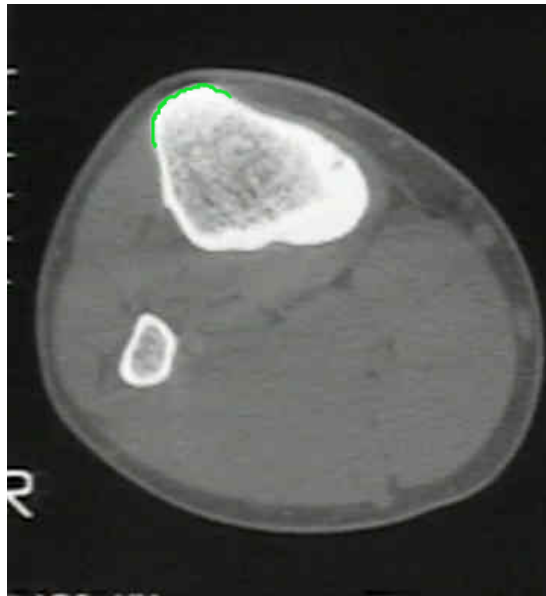


Figura 2.8: Imagen obtenida a través de una prueba TAC en una rodilla a la altura de la tibia, en la que ha sido coloreada la región de la TAT en verde.

La representación de una imagen obtenida a través de una prueba TAC en una rodilla a la altura de la tibia tiene una morfología similar a la expuesta en la figura 2.8.

En esta figura la región pintada en verde corresponde a la zona perteneciente a la tuberosidad anterior de la tibia.

#### 2.2.2.6 Medidas anatómicas

En esta sección se explican las medidas anatómicas que trata de automatizar el sistema propuesto en este proyecto.

## 2.2.2.6.1 TA-GT

La distancia TA-GT (Tuberosidad Anterior - Garganta Troclear) es una medida anatómica que se realiza para el diagnóstico de la inestabilidad patelar[15]. El proceso de cálculo de la medida es el siguiente:

1. Se escoge del estudio realizado con la prueba TAC la imagen del fémur que **mejor muestre** la tróclea, y de entre las imágenes de la tibia la imagen que mejor muestre la tuberosidad anterior, y se superponen ambas imágenes.
2. Se traza una línea tangencial al **borde posterior** (línea 3 en la figura 2.9) de los cóndilos del fémur.
3. Se traza una segunda línea, **perpendicular a la línea de los cóndilos** (línea 2 en la figura 2.9), que atraviese el punto más profundo de la tróclea.
4. Se traza una tercera línea, (línea 1 en la figura 2.9) **perpendicular a la línea de los cóndilos**, que atraviese al punto más saliente de la TAT.
5. Se calcula las **distancia** entre la línea de la tróclea y la línea de la TAT.

Un resultado por debajo de los 15 mm se considera un resultado normal. Las distancias mayores que 20 mm se consideran patológicas [15]. Esta medida cuenta con un margen de error de 4 mm [16].

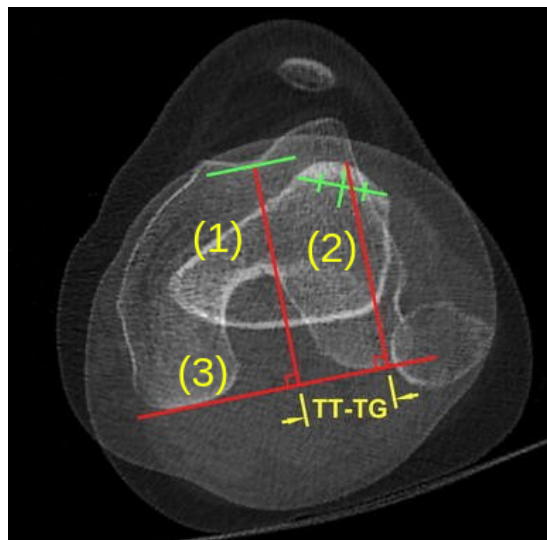


Figura 2.9: Ejemplo de cálculo de la distancia TA-GT

### 2.2.2.6.2 Básica Rotuliana

La medida Básica Rotuliana es una medida anatómica que se realiza para el diagnóstico de desviación rotuliana. El proceso de cálculo es el siguiente [17]:

1. Se escoge del estudio realizado con la prueba TAC la imagen del fémur que **mejor muestre** la tróclea.
2. Se traza una línea tangencial (línea 3 en la figura 2.10) al **borde posterior de los cóndilos del fémur**.
3. Se traza una segunda línea (línea 1 en la figura 2.10) que atraviese el eje transversal de la **rótula**.
4. Se forma un **ángulo** con una tercera línea (línea 2 en la figura 2.10) paralela a la línea de los cóndilos.

Un ángulo entre 10 y 12 ° se considera normal. Un ángulo por encima de los 20° se considera patológico.

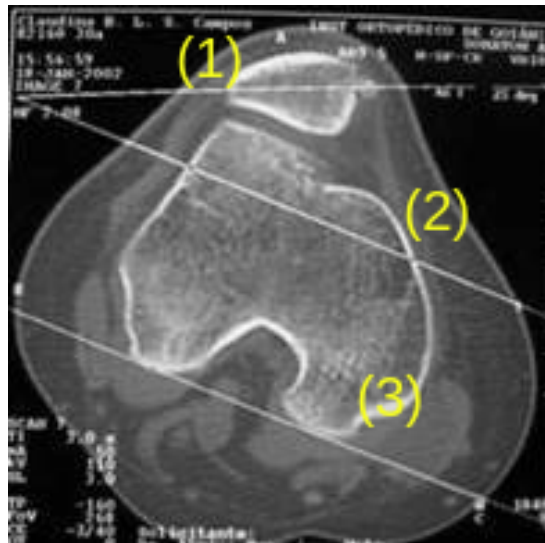


Figura 2.10: Ejemplo de cálculo de Básica Rotuliana





# Estado del Arte

---

**D**E manera previa a la realización de este trabajo se investigó sobre la existencia de algún software que realizase de manera automática las medidas Básica Rotuliana y TA-GT que se automatizan en este trabajo y son necesarias para el diagnóstico de lesiones en rodillas. En esta investigación se encontraron varias plataformas de acceso libre que permiten la visualización de imágenes DICOM, pero ninguna de estas plataformas contaban con una herramienta de cálculo automático de medidas Básica Rotuliana y TA-GT. A continuación se detallan en profundidades los sistemas más relevantes.

### 3.1 RadiAnt DICOM Viewer

RadiAnt DICOM Viewer es un visor DICOM PACS de imágenes médicas para PC disponible para Windows desde la versión XP en adelante. Es compatible con imágenes de diferentes pruebas médicas como TACs, resonancias magnéticas o radiografías, imágenes en color o monocromáticas... Tiene gran utilidad como herramienta de estudio de las imágenes gracias a características como la posibilidad de aplicar zoom a las imágenes, cambiar el contraste de las imágenes o calcular medidas en segmentos de las imágenes, entre otras [18].

A pesar de todo lo comentado, este programa no cuenta con certificado de producto médico.

### 3.2 PostDICOM

PostDICOM es un visor DICOM con versión para Windows, Linux y MacOS. PostDICOM cuenta con un PACS en la nube, el cuál permite acceder al usuario desde cualquier dispositivo [19].

### **3.3 MedDream DICOM Viewer**

MedDream DICOM Viewer es un visor web DICOM cuyo objetivo es la visualización, archivo y transmisión de archivos DICOM[20]. Cuenta con la capacidad ser integrado en algún sistema de computación en la nube como Google Cloud o Amazon Web Services.

En el apartado de visualización de imágenes, cuenta con la capacidad de poder realizar el trazado de líneas, polígonos y ángulos, entre otros, y poder realizar el cálculo de distancias y longitudes, dotando la capacidad de realizar medidas anatómicas en estas imágenes.

### **3.4 Conclusiones**

Tras haber investigado de manera exhaustiva las diferentes alternativas de uso libre en el sector de los visores de imágenes DICOM, no se ha encontrado ninguna que implemente ningún mecanismo para el cálculo automático de medidas anatómicas. Todos los visores encontrados implementan una solución genérica al visionado de todo tipo de imágenes médicas, y no existe ninguna alternativa que cuente con herramientas específicas para el cálculo de medidas anatómicas en imágenes de rodilla.

Debido a esta falta de herramientas de cálculo automático de medidas anatómicas en imagen médica, resulta adecuado implementar un sistema informático que dé solución a los problemas específicos del cálculo de medidas anatómicas en imagen de rodilla.

# Metodología

**T**Ras llevar a cabo un estudio del desarrollo del sistema a realizar y las diferentes fases que deberían seguir a lo largo del tiempo, se optó por utilizar una metodología incremental, basada en la noción de obtener productos parciales para cada iteración del desarrollo. Esta metodología se combina con un diseño dirigido por las pruebas[21], TDD por sus siglas en inglés, que obliga al desarrollador a contemplar todos los casos de prueba posibles antes de realizar el desarrollo, y a que el software realizado se evalúe continuamente siguiendo todos los casos de prueba.

La metodología incremental permite planificar el proyecto en diferentes bloques temporales denominados iteraciones. Estas iteraciones se descomponen en 4 fases cada una: análisis, diseño, implementación y pruebas, tal y como se recoge en la figura 4.1, que se realizarán de manera secuencial. Uno de los grandes beneficios del uso de esta metodología es que permite la entrega de productos parciales que facilitan la obtención de *feedback* por parte del usuario/cliente que puede ser tomado en cuenta para las futuras iteraciones.

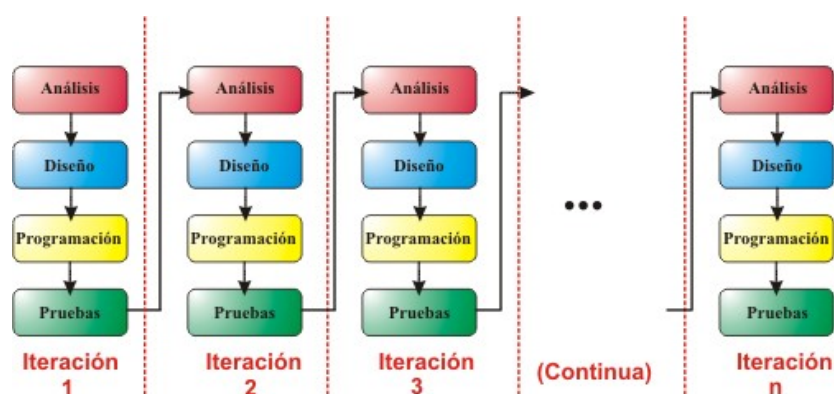


Figura 4.1: Metodología incremental

Para el desarrollo del proyecto se propone seguir un diseño propuesto por las pruebas para los componentes del sistema que sea posible, escribiendo las pruebas unitarias del código

antes que la implementación, para obligar a refactorizar todo el código que no pase las pruebas propuestas.

## 4.1 Iteraciones propuestas

De manera previa al comienzo del desarrollo del sistema, se realizó un análisis del problema a resolver, teniendo en cuenta todos los requisitos funcionales y no funcionales que debería cumplir el proyecto. Se realizó una lista de requisitos funcionales y se realizó una planificación del proyecto en 4 iteraciones que siguen el esquema de la figura 4.1, y una iteración previa al desarrollo, llamada "Iteración 0", donde se realiza un análisis del sistema a realizar y una descomposición compleja de las funcionalidades a implementar para cada una de las iteraciones. La lista de iteraciones es la siguiente:

- Iteración 0: análisis del sistema y descomposición en detalle de las iteraciones.
- Iteración 1: Lectura de ficheros DICOM, segmentación de huesos y regiones óseas.
- Iteración 2: Obtención de puntos de interés necesarios para el cálculo de las medidas y funciones automáticas de cálculo.
- Iteración 3: Desarrollo de una interfaz gráfica que permita cargar un estudio DICOM y elegir las imágenes que se vayan a utilizar en las medidas, y visionado por pantalla del resultado de las medidas y los puntos de interés obtenidos.
- Iteración 4: Adición de nuevas funcionalidades a la interfaz: posibilidad de guardar los resultados de las medidas y datos del estudio en disco y refinado de las medidas de manera manual por parte del usuario.

## 4.2 Planificación del proyecto

Una vez fueron definidas de manera definitiva las iteraciones del proyecto, se llevó a cabo la planificación del proyecto. En esta planificación se estableció una estimación del proyecto que queda reflejada en la tabla 4.1.

Ante la imposibilidad de establecer una carga de trabajo constante para cada día, la estimación se llevó a cabo asumiendo una carga de trabajo semanal de 40 horas a ser distribuidas en función de la disponibilidad horaria y de las necesidades del proyecto.

Exceptuando las iteraciones 0 y 3, la planificación del proyecto sufrió desviaciones positivas en la estimación.

Estos fueron los principales motivos que provocaron las desviaciones:

Iteración	Tiempo estimado	Tiempo real	Desviación	Fecha inicio	Fecha fin
Iteración 0	17 h	15 h	-2 h	16/07/2020	17/07/2020
Iteración 1	160 h	200 h	+40 h	20/07/2020	21/08/2020
Iteración 2	240 h	300 h	+60 h	24/08/2020	14 /10/2020
Iteración 3	200 h	180 h	-20 h	19/10/2020	18/11/2020
Iteración 4	240 h	300 h	+60 h	23/11/2020	13/01/2021
<b>Total</b>	<b>855 h</b>	<b>993 h</b>	<b>+138 h</b>	<b>29/06/2020</b>	<b>15/01/2021</b>

Tabla 4.1: Planificación de las iteraciones

- En las iteraciones 1 y 2 los problemas surgidos fueron, en primer lugar, la necesidad de formación del alumno en las técnicas de visión artificial y en los diferentes conceptos médicos necesarios para comprender el dominio del problema. Además, la saturación del sistema sanitario provocada por la pandemia de COVID-19 provocó que el personal médico encargado de facilitar los estudios DICOM de prueba no tuviese tiempo para facilitar estos recursos hasta pasadas unas semanas de desarrollo, causando varios bloqueos.
- En la iteración 4, fue necesario realizar una investigación exhaustiva de los diferentes componentes que pudieran ser útiles para implementar el sistema de refinado manual de medidas tras haber encontrado varios problemas en la fase de implementación en relación a la carga de imágenes en las ventanas o a la captura de eventos.

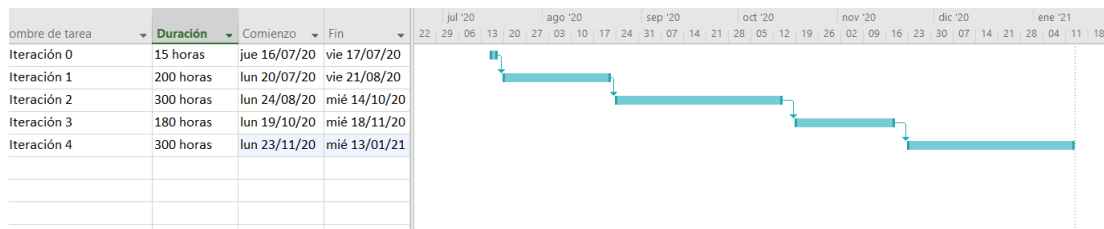


Figura 4.2: Diagrama de Gantt para la planificación del proyecto

En la figura 4.2 se muestra el diagrama de Gantt realizado con la herramienta MS Project para la planificación del proyecto.



## Capítulo 5

# Desarrollo

---

**E**N este capítulo se desgranar los detalles de cada iteración del desarrollo mencionada en el capítulo anterior. Se separan las fases de análisis, diseño, implementación y pruebas para cada iteración, y se explican en detalle.

### 5.1 Iteración 0

Tal y como se explica en el capítulo 4.1, la iteración 0 consiste en la realización del análisis del sistema y la descomposición de las futuras iteraciones del proyecto. Esta iteración sigue una estructura completamente diferente a las siguientes iteraciones, y no es una iteración como tal, sino un punto de partida para poder tener un marco de referencia de cara al desarrollo del sistema.

#### 5.1.1 Análisis del sistema

En primer lugar, se realizó un diagrama informal para comprender el dominio del problema a solucionar e indicar de una manera sencilla las funcionalidades que aporta al usuario el sistema propuesto.

Siguiendo la estructura del diagrama de la figura 5.1, queda explicado de manera resumida la funcionalidad del sistema:

1. Se exportan las imágenes médicas obtenidas en la prueba de TAC a un ordenador a ficheros que siguen el estándar DICOM.
2. El sistema carga ese estudio y permite al usuario ver las imágenes de los ficheros, para escoger las que considere que son mejores para el cálculo de las medidas.
3. El sistema visualiza la imagen final y el resultado obtenido por la medida anatómica.

Además, el sistema debe ser capaz de cubrir las siguientes funcionalidades adicionales:



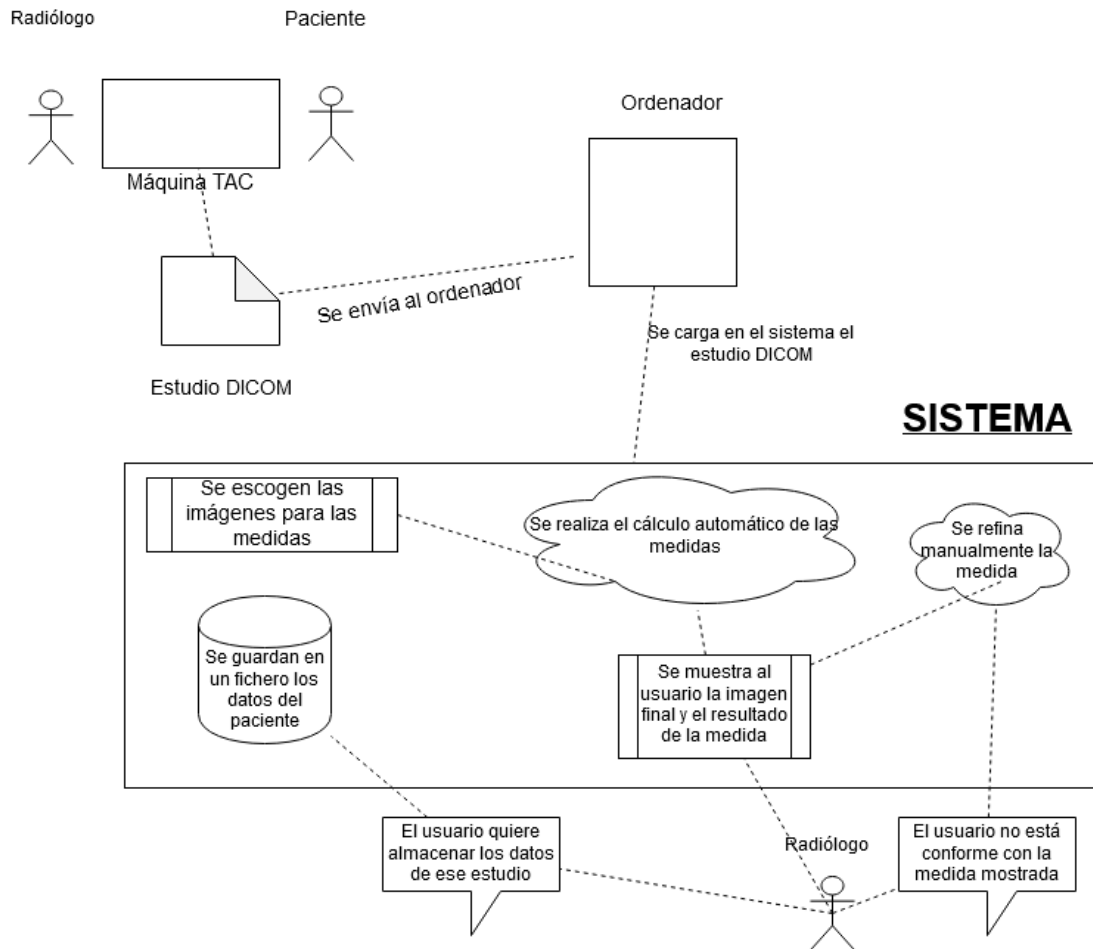


Figura 5.1: Diagrama informal que explica el flujo básico de ejecución y las funcionalidades principales del sistema propuesto.

- En caso de que el usuario quiera conservar los resultados del estudio, poder almacenar los resultados de las pruebas y los datos del paciente a través de algún sistema de persistencia de datos.
- Si el usuario considera que el sistema de cálculo automático ha trazado mal alguna de las líneas necesarias para el cálculo, poder corregirlas de manera manual y refinar la medida realizando el cálculo con las líneas que haya corregido.

### 5.1.2 Descomposición de las iteraciones

Para realizar el desarrollo del sistema planteado en este proyecto se realizó una descomposición de las tareas y funcionalidades principales del desarrollo para cada iteración, con el siguiente resultado:

- **Iteración 1:** Desarrollo de un sistema capaz de procesar las imágenes TAC necesarias para el cálculo de las medidas, implementando las siguientes funcionalidades.
  - un sistema de lectura ficheros DICOM en formato .dcm, con la capacidad para extraer la imagen contenida y los TAGs contenidos.
  - un sistema de conversión la imagen médica a la escala de Hounsfield, modificar el rango de valores de la imagen, y normalizarla a diferentes escalas de valores.
  - un método de umbralización robusto, capaz de obtener unos resultados aceptables para cualquier imagen probada, que permita segmentar los contornos de los huesos fémur, rótula y tibia, sin deformar sus contornos.
  - un sistema de reconocimiento de las figuras principales de la imagen(fémur, rótula, tibia) que permita eliminar el resto de la información de la imagen.
- **Iteración 2:** Desarrollo de un sistema capaz de obtener los puntos de interés de las imágenes necesarios para realizar los cálculos, y con la capacidad de automatizar el cálculo de las medidas.
  - Un sistema capaz de obtener los puntos de interés de cada región ósea (fémur, rótula, tibia).
  - Un método de cálculo de medidas Básica Rotuliana automatizado a través de los puntos obtenidos, que permita minimizar el error cometido en las medidas realizadas manualmente.
  - Un sistema de visualización de los puntos de interés encontrados en cada imagen.
- **Iteración 3:** Desarrollo de una interfaz gráfica que permita utilizar todas las funcionalidades desarrolladas en las iteraciones 1 y 2 de una manera cómoda y sencilla.
  - Una interfaz gráfica que permita al usuario la carga y visualización de imágenes y la obtención del resultado tras el cálculo de las medidas TA-GT y Básica Rotuliana.
  - Una integración del código desarrollado en las iteraciones 1 y 2, respetando el modelo de datos diseñado anteriormente, y aplicando los principios de diseño del patrón Modelo-Vista-Controlador.
- **Iteración 4:** Implementación de nuevas funcionalidades a la interfaz gráfica, permitiendo guardar un registro de estudios médicos y un refinado manual de las medidas TA-GT y Básica Rotuliana.
  - Un sistema de persistencia que permita almacenar en un fichero los datos personales del paciente, acceder a las imágenes médicas que se hayan utilizado en el cálculo de medidas y los resultados de estas medidas.

- Una herramienta de refinado de las medidas TA-GT y Básica Rotuliana, para que, en caso de que el usuario observe que las medidas calculadas de manera automática no son precisas, pueda indicar manualmente los puntos de interés que considere que no han sido calculados correctamente y calcular nuevamente las medidas con los puntos corregidos por el usuario.

### 5.1.3 Recursos empleados

Para poder probar y validar los métodos y funcionalidades relacionados con el procesado de imágenes, obtención de puntos característicos y cálculo de medidas, era condición necesaria contar con varios estudios médicos. Estos estudios fueron facilitados por el Complejo Hospitalario Universitario de A Coruña(CHUAC), contando con los siguientes recursos:

- Dos estudios completos, con las imágenes más apropiadas para realizar medidas TA-GT, y con las medidas calculadas en ambos lados de la rodilla, sirviendo estos resultados de TA-GT como marco de referencia para poder comparar con los resultados aportados por el sistema automático.
- Cuatro estudios completos, sin resultados de medidas TA-GT, que no sirven para validar los resultados TA-GT, pero sirven para poder comprobar si la segmentación de regiones óseas y la extracción de puntos característicos devuelven resultados coherentes.

Debido a la saturación del sistema sanitario durante la elaboración de este proyecto causada por la pandemia de COVID-19, no fue posible contactar con más médicos ni centros sanitarios que pudieran aportar más estudios que sirviesen de prueba, lamentando especialmente la ausencia de estudios de prueba para el cálculo de Básica Rotuliana.

Todos estos estudios facilitados habían sido anonimizados previamente y no contenían ningún dato que relacionaran el estudio médico con el paciente al que pertenecía con el fin de cumplir con la legislación vigente en relación a la Protección de Datos.

#### 5.1.3.1 Formato de las imágenes

Para poder simplificar los criterios de procesado de las imágenes de las series DICOM, se identificaron una serie de características que son comunes a todas las imágenes pertenecientes a las series:

- Todas las imágenes obtenidas cuentan con un estudio TAC realizado en ambas piernas del paciente. Esto quiere decir que siempre se encuentra la imagen obtenida para cada pierna en cada una de las mitades de la imagen.

- En cuanto a las imágenes pertenecientes a la sección del fémur, todas tendrán una morfología similar a la de la figura 5.2: en cada una de las mitades de la imagen se puede observar el fémur del paciente en cada pierna, y la rótula situada encima de este.
- En cuanto a las imágenes pertenecientes a la sección de la tibia, todas tendrán una morfología similar a la de la figura 5.3: en cada una de las imágenes se puede observar la tibia del paciente en cada pierna.

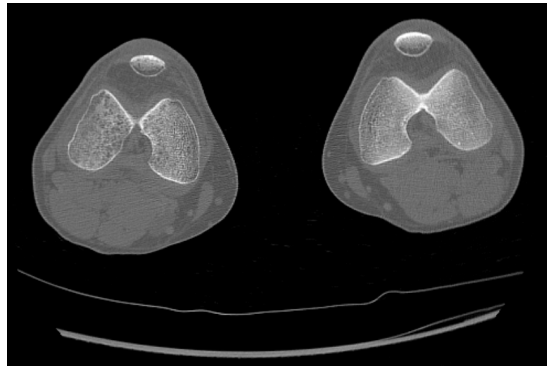


Figura 5.2: Ejemplo de imagen DICOM perteneciente a la sección del fémur

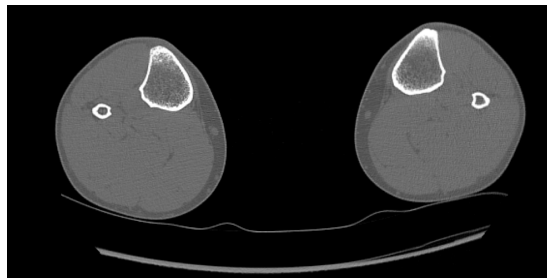


Figura 5.3: Ejemplo de imagen DICOM perteneciente a la sección de la tibia

De esta manera, se asume que todas las imágenes utilizadas para probar la implementación del sistema cumplen los siguientes puntos:

- Todas las imágenes poseen las características anteriormente descritas.
- Todas las imágenes pertenecientes a un mismo estudio tienen las mismas dimensiones.
- Todas las imágenes pertenecientes a un mismo estudio fueron tomadas desde la misma posición.

En caso de que se utilicen imágenes que no cumplen los puntos anteriormente descritos, no está garantizado que la medida aportada por el sistema sea fiable.

## 5.2 Primera iteración

Tal y como se explica en el capítulo anterior, la primera iteración del desarrollo consiste en el desarrollo de un sistema capaz de leer imágenes DICOM en formato .dcm, procesar las imágenes y los TAGs contenidos en los ficheros, y realizar una segmentación de los contornos útiles para las medidas.

Para conseguirlo, primero se diseñó un procedimiento de lectura de los ficheros .dcm. En esta iteración, se introducen los nombres de los ficheros a través de argumento de entrada tras la llamada al script desarrollado en Python correspondiente.

También se desarrollaron unos scripts de procesado de las imágenes extraídas, que permitían su conversión a la escala de Hounsfield y su modificación de rango de valores para cada píxel de la imagen.

Por último, se desarrollaron también unos scripts que permiten la obtención de los contornos de los huesos encontrados en la imagen, y su separación del resto del contenido de la imagen.

### 5.2.1 Análisis

En primer lugar, se realizó un análisis de requisitos necesario para definir los casos de uso. A continuación se detallan el análisis de requisitos y los casos de uso.

#### 5.2.1.1 Análisis de requisitos

Tras un estudio de las funcionalidades a implementar en la primera iteración, se obtuvo la siguiente lista de requisitos.

- **Lectura de los ficheros DICOM**, en formato .dcm, y acceso a los TAGs que contienen, pudiendo leer los siguientes TAGs necesarios para la manipulación de las imágenes:
  - **Pixel Array**, donde se almacena la imagen médica.
  - **Window Width y Window Center**, para obtener el centro y ancho de ventana de la imagen abierta.
  - **Rescale Slope y Rescale Intercept**, para poder transformar la imagen contenida en Pixel array a la escala de unidades Hounsfield.
  - **Pixel Spacing**, para poder calcular la distancia en milímetros entre dos píxeles.
- **Procesado de las imágenes médicas** para obtener los contornos de las estructuras anatómicas, siendo necesarias las siguientes herramientas:

- **Un sistema de transformación de valores de los píxeles a escala de Unidades Hounsfield**, el cual permita diferenciar de manera sencilla los tejidos óseos del resto de tejidos y sustancias existentes en la imagen.
- **Un sistema de segmentación de los tejidos óseos de la imagen**, aplicando una heurística que encaje con el contexto y las particularidades del dominio del problema, y que obtenga unos resultados globales aceptables.
- **Un sistema de segmentación que permita separar en imágenes diferentes cada uno de los huesos encontrados en las imágenes**, separando por regiones (rodilla izquierda y derecha), y por cada uno de los huesos encontrados (fémur, rótula y tibia).

### 5.2.1.2 Casos de uso

En esta sección se detallan los casos de uso implementados en la primera iteración, en la que se define un único actor "Usuario".

A continuación se profundiza en cada uno de los casos de uso.

- **Lectura de ficheros .dcm y acceso a sus TAGs:** se implementa un servicio que dispone de una herramienta de lectura de ficheros .dcm, y que permita acceder a los TAGs relevantes y obtener sus valores.
- **Obtención de las estructuras óseas de la imagen:** se desarrolla una herramienta de segmentación de estructuras óseas basada en técnicas de Visión Artificial que permite obtener una imagen con todos los componentes no óseos eliminados.
- **Segmentación de las regiones óseas de la imagen:** se desarrolla una herramienta de segmentación de regiones óseas obteniendo imágenes que contienen por separado cada uno de los huesos existentes en la imagen, obteniendo una imagen por cada hueso, y por cada rodilla.

## 5.2.2 Diseño

En esta sección se detalla el diseño elegido para la implementación de los casos de uso detallados en el apartado anterior.

### 5.2.2.1 Entrada y salida de datos

Como en esta iteración no se implementa todavía una interfaz gráfica, la entrada y salida de datos se realizará vía consola. Como tampoco se implementa la realización automática de

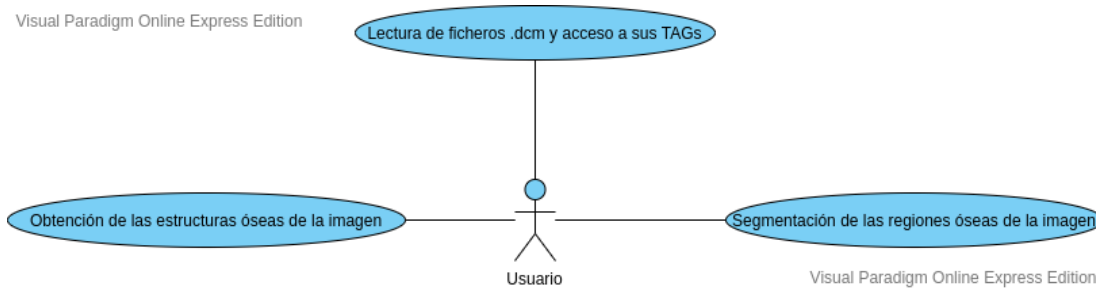


Figura 5.4: Casos de uso para el actor "Usuario" en la Iteración 1

medidas TA-GT, en esta iteración el sistema sólo soportará la apertura de un único fichero DICOM, que tendrá que estar en formato *.dcm*.

Para ejecutar cada uno de los casos de uso, se especificará también como argumento de entrada otro parámetro que indicará qué operación ha de realizarse.

En el caso de obtener los TAGs del fichero, estos se mostrarán por salida estándar en la consola en la que se ejecute el proyecto. En el caso de obtener alguna imagen, el sistema abrirá una ventana donde se mostrará la imagen.

### 5.2.2.2 Estructura del código

El objetivo principal del diseño de la estructura del código es la separación de la lógica. De esta manera, se crearán tres módulos escritos en Python, en los que se separará la lógica de lectura de ficheros DICOM y la lógica de procesamiento de imágenes. Finalmente, el código queda estructurado de la siguiente manera:

- **Un módulo llamado *dicom-utils***, donde se implementará todo el código relativo a la lectura de ficheros DICOM y obtención de valores de los TAGs.
- **Un módulo llamado *image-processing***, donde se implementará todo el código relativo al procesado de las imágenes.
- **Un módulo llamado *interesting-points***, donde se implementarán las funciones relativas a la extracción de sub-imágenes para cada región ósea o mitad de la imagen.
- **Un script de nombre *main***, que se encarga de hacer las llamadas al código implementado en los dos módulos, y que devuelve los resultados de la ejecución.

De cara a facilitar el trabajo de futuras iteraciones, se tuvo en cuenta desarrollar un **modelo de datos utilizando Programación Orientada a Objetos** que se acople como modelo de la **arquitectura MVC** (Modelo-Vista-Controlador) cuando se desarrolle la interfaz gráfica

a partir de la tercera iteración. De esta manera, se constituyen las siguientes clases de las que se podrán crear objetos únicos:

- **Una clase llamada MedicalImage**, donde se almacenan en forma de atributos los TAGs que se utilizan para el procesamiento de imagen, mencionados en el análisis de requisitos. Además, permite almacenar imágenes procesadas tras un proceso de segmentación, llamando a las funciones implementadas en los módulos dicom-utils y image-processing.
- **Dos clases** que heredan de MedicalImage, para almacenar imágenes de los segmentos de Fémur-Rótula y Tibia, con nombres **FemurRotulaImage** y **TibiaImage** respectivamente, que añaden atributos donde se almacenan resultados parciales de segmentación de rótula y fémur en el caso de FemurRotulaImage, y de segmentación de tibia en el caso de TibiaImage.

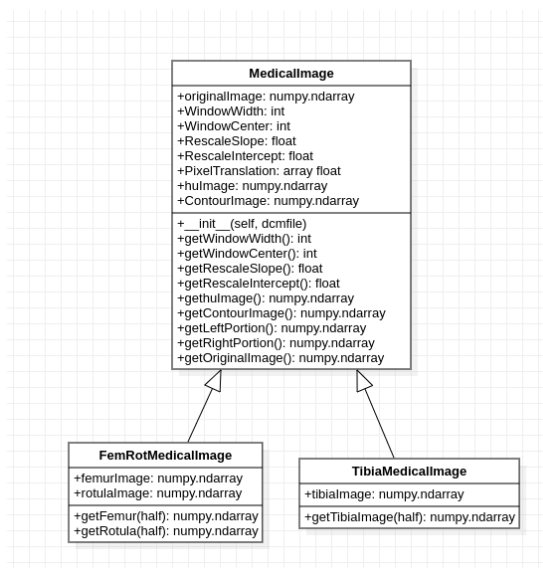


Figura 5.5: Modelo de datos diseñado para el manejo de las diferentes imágenes médicas

### 5.2.3 Implementación

En esta sección se explican en profundidad los detalles de implementación del código propuesto en la primera iteración. En concreto, se implementan los módulos dicom-utils y image-processing y el modelo de datos explicado en la sección de Diseño.



### 5.2.3.1 Módulo dicom-utils

En este módulo se implementan (e implementarán en futuras iteraciones) las funciones relacionadas con el procesado de datos provenientes de ficheros DICOM. En concreto, en esta iteración se implementan funciones para procesar el TAG pixel-array de los ficheros DICOM.

Los ficheros DICOM son leídos por el código a usando la función *dcmread* de la librería *pydicom* [22], que devuelve un objeto que contiene los valores de todos los tags DICOM del fichero.

#### 5.2.3.1.1 Transformación a la Escala de Unidades Hounsfield

Tal y como se explica en el apartado 2.1.1.3, la escala Hounsfield es una escala cuantitativa para describir la radiodensidad. Esta transformación es útil para poder diferenciar los diferentes tipos de tejidos, puesto que el valor en su píxel se ajustará a los valores establecidos en la escala de unidades Hounsfield.

Para poder realizar la transformación de la imagen a escala de Hounsfield, se implementa la función *transformToHu*. Esta función aplica la fórmula descrita en el apartado 2.1.1.3: se aplica la conversión lineal para cada píxel utilizando la función  $y = m \times x + b$ , siendo  $y$  el valor de salida en la escala Hounsfield,  $x$  el valor original para cada píxel de la imagen,  $m$  el valor contenido en el TAG Rescale Slope, y  $b$  el valor contenido en el TAG Rescale Intercept.

### 5.2.3.2 Módulo image-processing

En este módulo se implementan (e implementarán en futuras iteraciones) las funciones relacionadas con el procesado de las imágenes extraídas de los ficheros DICOM y convertidas a escala de unidades Hounsfield gracias al módulo dicom-utils, tal y como se explicó en el apartado anterior.

#### 5.2.3.2.1 Problemática de la escala de Unidades Hounsfield

Tal y como se explica en apartados anteriores, la escala de unidades Hounsfield permite dotar de un valor dentro de unos rangos definidos a cada tipo de tejido, tal y como se puede observar en la figura 2.3. A priori, esto nos induce a pensar que el proceso de segmentación sería trivial, ya que podría resultar del proceso de extraer sólo aquellos píxeles que pertenezcan al rango de valores adecuado (e.g. en este sistema, extraer sólo aquellos píxeles con valores mayores a 1000 para obtener el tejido óseo).

Sin embargo, la escala de Unidades Hounsfield es una escala de unidades relativas, y no absolutas. Esto quiere decir que las Unidades Hounsfield pueden depender de ciertas características de la máquina de TAC que toma la imagen [23], como el haz de luz de rayos X emitido por la máquina, o el voltaje con el que cuenta.

Esta circunstancia obliga a buscar una heurística aplicable de manera global a un algoritmo de segmentación de regiones óseas que permita obtener resultados aceptables independientemente de las características que pueda tener la máquina que haya tomado la imagen.

#### 5.2.3.2.2 Función *thresholdHUImage*

En la función *thresholdHUImage* se dota de un método de segmentación de los tejidos óseos de las imágenes TAC. La función recibe como entrada la imagen previamente transformada a escala de Unidades Hounsfield, y devolverá una imagen producto de un proceso de Umbralización dónde los tejidos óseos tengan un valor en los píxeles correspondientes a 1, y el resto de entidades de la imagen, a 0.

Tras la implementación de varios modelos de funciones candidatas de umbralización, se comparó el resultado obtenido por cada uno de ellos y se eligió una función como la función a utilizar, descartando el resto de modelos. La heurística de la función de umbralización elegida es la siguiente:

1. Se obtiene un valor de umbral para cada iteración del algoritmo, inicialmente  $t_0 = (\min(\text{Imagen}) + \max(\text{Imagen}))/2$ .
2. Se crean dos imágenes, inicialmente con todos los píxeles a valor cero, y de mismas dimensiones que la imagen de entrada, que servirán para separar en una imagen los tejidos óseos, y en la otra todo lo demás. Tendrán de nombre *imgforeground* y *imgbackground*, respectivamente.
3. Se recorre cada píxel de la imagen original, comparando el valor almacenado en el píxel con el valor de umbral para la iteración correspondiente,  $t_n$ . Si el valor de píxel es más alto que el valor de umbral  $t_n$ , el píxel se pintará en *imgforeground*, y en caso contrario, se pintará en *imgbackground*.
4. Se calcula el valor medio de píxel de *imgforeground*, *avgfore*, y el valor medio de píxel de *imgbackground*, *avgback*. Con esto se calcula el umbral para la siguiente iteración del algoritmo,  $t_{n+1} = (\text{avgfore} + \text{avgback})/2$ . Si  $t_{n+1} = t_n$  el algoritmo ha terminado y se devuelve como resultado *imgforeground*. En caso contrario se vuelve al paso 2.

En la figura 5.6 se muestra un ejemplo del resultado obtenido por la función *thresholdHUImage*.

#### 5.2.3.3 Módulo *interesting-points*

En esta función se implementan las funciones que permiten obtener imágenes de una mitad de la imagen original (izquierda derecha) o de un único hueso (femur o rótula en el

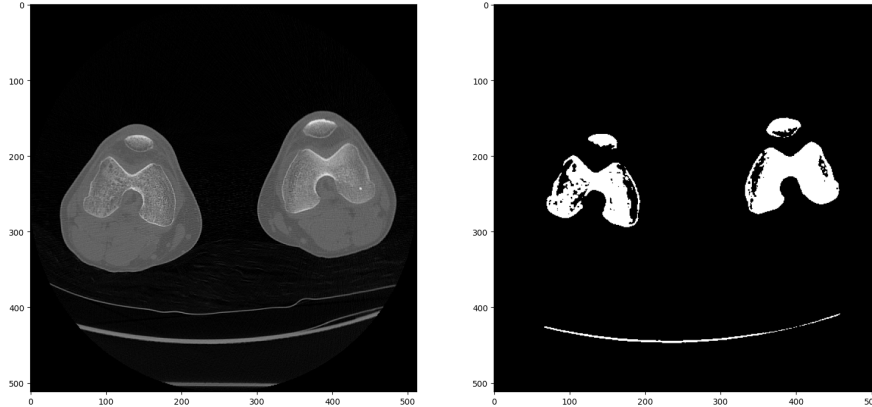


Figura 5.6: Ejemplo de ejecución de la función `thresholdHUImage`. A la izquierda, la imagen que se le pasa como entrada, y a la derecha, la imagen resultante del proceso de segmentación.

caso de las imágenes que contengan estos huesos).

#### 5.2.3.3.1 Obtención de mitades izquierda-derecha

Para obtener la mitad izquierda o derecha de una imagen, se utiliza un algoritmo sencillo. Primero se obtiene el ancho de la imagen, y se calcula la posición central de la imagen dividiendo el ancho entre dos. Una vez obtenido este valor, se pintan únicamente los píxeles a la izquierda de ese valor en el caso de querer obtener la imagen izquierda, y únicamente los píxeles a la derecha en caso de querer obtener la imagen derecha.

#### 5.2.3.3.2 Obtención de fémur o rótula

Para obtener una imagen que contenga únicamente el fémur (izquierdo o derecho) o la rótula (izquierda o derecha), primero se ha de obtener la imagen parcial izquierda o derecha aplicando el algoritmo aplicado en el apartado anterior.

A esta imagen se le aplica la función `findContours` perteneciente al módulo OpenCV, que nos devuelve una lista con los contornos encontrados en la imagen. Esta lista se debe ordenar por orden de tamaño de los contornos. Ahora, aplicando la información a priori de que el mayor contorno encontrado pertenecerá al fémur, se extrae la primera ocurrencia de la lista de contornos y se calcula la coordenada "norte" de ese contorno. Esta coordenada "norte" corresponde al píxel cuyo número de fila sea el más pequeño.

Una vez obtenido esta coordenada "norte", para obtener la imagen del fémur se pintarán

sólo los píxeles en la misma fila o en filas posteriores, y para obtener la imagen de la rótula se pintarán los píxeles en filas a anteriores a la coordenada "norte".

#### 5.2.4 Modelo de datos

Tal y como se explica en el apartado de Diseño, se implementó un modelo de datos utilizando la Programación Orientada a Objetos de Python. Se creó una clase *MedicalImage*, dónde se almacenarán todos los TAGs que sean útiles para el desarrollo del sistema en forma de atributos, y un método constructor que tiene como parámetro de entrada la ruta del fichero *.dcm* donde se encuentra la imagen DICOM.

Heredando de *MedicalImage* se crean dos clases más, *FemRotMedicalImage* y *TibiaMedicalImage*, con atributos propios para almacenar imágenes de fémur y rótula, y tibia, respectivamente. Además, se implementan unas funciones de obtención de estas imágenes, que se limitan a llamar a las funciones implementadas en los módulos *dicom-utils*, *image-processing* y *interesting-points*.

#### 5.2.5 Pruebas

En esta iteración se diseñaron unos test para el modelo de datos. Se implementaron utilizando el módulo *unittest* de Python. Además, se realizaron pruebas funcionales.

Para probar el correcto funcionamiento del código perteneciente al procesado de imágenes, se realizaron ejecuciones de todas las funciones implementadas utilizando un banco de imágenes de prueba, comprobando que el funcionamiento global era correcto.

### 5.3 Segunda iteración

Esta iteración corresponde al desarrollo de mecanismos que permitan obtener los puntos características. En concreto, se añaden funciones que permitan reconocer los puntos característicos de las regiones óseas necesarios para realizar las medidas de Básica Rotuliana y TA-GT, y automatizar el proceso de medida.

Para poder llevarlo a cabo, se añadieron nuevas funciones a los módulos ya existentes, y se implementaron nuevos métodos en las clases diseñadas en la primera iteración.

#### 5.3.1 Análisis

En primer lugar, se realizó un análisis de requisitos necesario para definir los casos de uso. A continuación se detallan el análisis de requisitos y los casos de uso.

### 5.3.1.1 Análisis de requisitos

Tras un estudio del cálculo de medidas de Básica Rotuliana y TA-GT para poder entender en profundidad la complejidad del problema y conocer qué puntos se deben extraer y cómo se realizan las medidas, se obtuvo la siguiente lista de requisitos.

- Obtención de **puntos de interés del fémur**:
  - Punto más bajo de cada cóndilo, para poder trazar una línea que cruce el borde posterior de ambos cóndilos.
  - Punto más profundo de la tróclea
- Obtención del **eje transversal de la rótula**
- Punto más saliente de la **tuberosidad anterior de la tibia (TAT)**.
- Cálculo de medida **Básica Rotuliana**.
- Cálculo de medidas **TA-GT**.

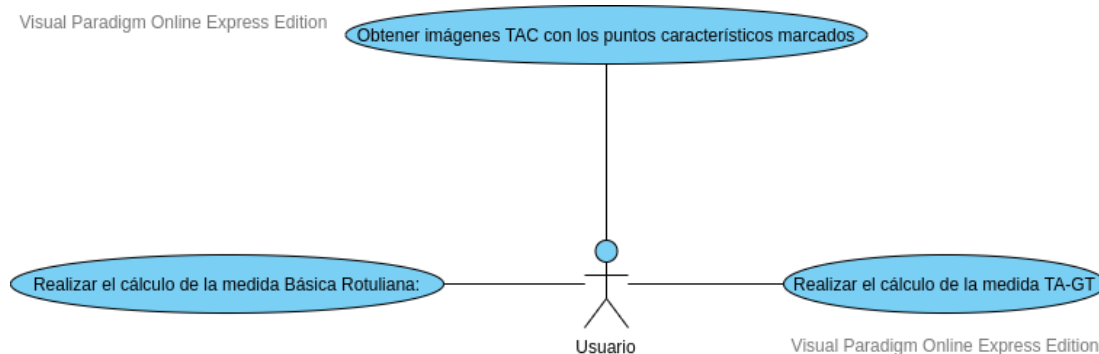


Figura 5.7: Casos de uso para el actor "Usuario" en la segunda iteración

### 5.3.1.2 Casos de uso

En esta sección se explican los diferentes casos de uso correspondientes a la segunda iteración (Figura 5.7).

Todos los casos de uso van a ser ejecutados por un único rol de usuario, que corresponde al personal sanitario al que va destinado el uso del sistema.

En concreto, los casos de uso correspondientes a esta iteración son:

- Obtener imágenes TAC con los **puntos característicos** marcados : Se implementa un sistema que permita devolver las imágenes de entrada del sistema con sus puntos de interés destacados a color para que sean fácilmente identificables por el usuario.

- Realizar el cálculo de la medida **Básica Rotuliana**: Se desarrollan las funciones necesarias para poder realizar este cálculo, obteniendo los puntos de interés necesarios, y las funciones matemáticas que permitan automatizar el cálculo de manera precisa y rigurosa.
- Realizar el cálculo de la medida **TA-GT**: Igual que en el caso anterior, se desarrollan las funciones necesarias para poder realizar este cálculo, obteniendo los puntos de interés necesarios, y las funciones matemáticas que permitan automatizar el cálculo de manera precisa y rigurosa.

### 5.3.2 Diseño

En esta sección se detallan todos los detalles relativos al diseño elegido para el desarrollo de la segunda iteración.

#### 5.3.2.1 Entrada y salida de datos

En esta iteración se mantiene el sistema de entrada de imágenes por consola a través de argumento de la llamada de ejecución del script *main*. Como novedad, ahora sí que se permite la entrada de **dos imágenes**, puesto que es necesario trabajar con la imagen rodilla-fémur y tibia para realizar el cálculo de medidas TA-GT.

#### 5.3.2.2 Estructura del código

La estructura del código propuesta en la primera iteración se mantiene.

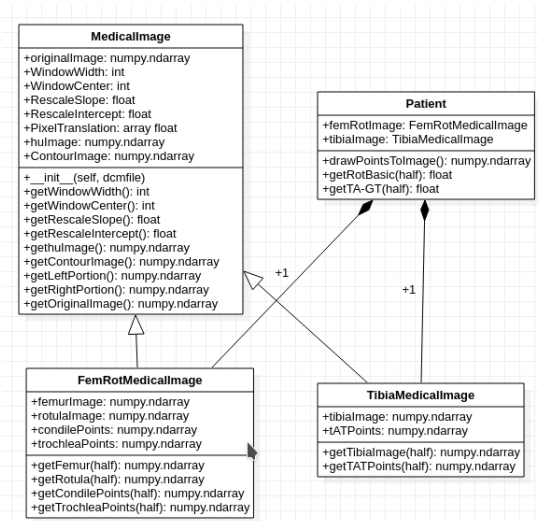


Figura 5.8: Modelo de datos diseñado para el manejo de las diferentes imágenes médicas, dónde se reflejan los cambios diseñados en esta iteración.

En esta iteración se añaden las funciones necesarias para el desarrollo a los módulos, y además se implementa todo el código relativo al cálculo matemático necesario para obtener las medidas. Además, se añade un nuevo módulo, *measures*, donde se implementarán las funciones que realicen el cálculo de medidas TA-GT y Básica Rotuliana.

Con respecto al modelo de datos, se añaden nuevos atributos y métodos a las clases *FemRotMedicalImage* y *TibiaMedicalImage*, y se diseña una nueva clase, *Patient* (figura 5.8), que contiene una instancia de cada una de las clases anteriores, y que tiene funciones que devuelven como resultado el cálculo de las medidas a implementar.

### 5.3.3 Implementación

En esta sección se explican en profundidad todos los detalles de implementación de las funciones y estructuras de datos desarrolladas en la segunda iteración.

#### 5.3.3.1 Módulo *image-processing*

En este módulo se añaden funciones para poder rotar las figuras de los huesos encontrados en las imágenes, necesario para poder obtener los puntos de interés de cada uno de los huesos.

##### 5.3.3.1.1 Problemática con la orientación de los huesos en las imágenes

Para encontrar los puntos de interés de las imágenes, basta en la mayoría de casos con buscar el píxel mas orientado a un punto cardinal (e.g., obtener el píxel más al norte).

Pero para poder obtener los puntos de interés de los huesos correctamente siguiendo criterios de puntos cardinales, es necesario realizar una rotación de los huesos con respecto a la horizontal de la imagen, tal y como se muestra en la figura 5.9.

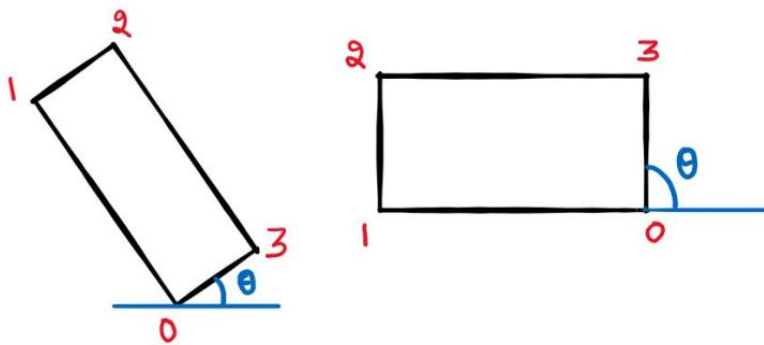


Figura 5.9: Ejemplo de rotación de un objeto con respecto a la horizontal de una imagen

Con estas imágenes con los huesos rotados, se simplifica el proceso de obtención de puntos

de interés en los huesos y se obtienen un nivel de precisión suficientemente alto como para aceptar los resultados obtenidos.

### 5.3.3.1.2 Función *rotateFemur*

Esta función devuelve como resultado una imagen con el fémur rotado con respecto a la horizontal de la imagen. Como entrada, recibe la imagen umbralizada de una única mitad de la imagen (e.g. mitad izquierda).

El primer paso de la ejecución consiste en la obtención de todos los **contornos** que exista la imagen a través de la función de OpenCV *findContours*. Una vez obtenida la lista de contornos, se calculan las **áreas** contenidas en cada uno de los contornos con la función *contourArea* de OpenCV, y se extrae de la lista de áreas el área más grande, debido a que gracias a la información a priori se sabe que el fémur siempre será el contorno al que le corresponde tener el área más grande.

Con este área se calcula el rectángulo de área mínima que podría encerrar al contorno del fémur llamando a la función de OpenCV *minAreaRect*. Esta función tiene, entre otros valores, como salida el ángulo que forma el área mínima calculada. Este ángulo es el ángulo que existe entre el lado del rectángulo dónde se encuentra la arista más a la derecha de la imagen, y el borde vertical de la imagen. En la figura 5.10 se muestra un ejemplo de este cálculo.

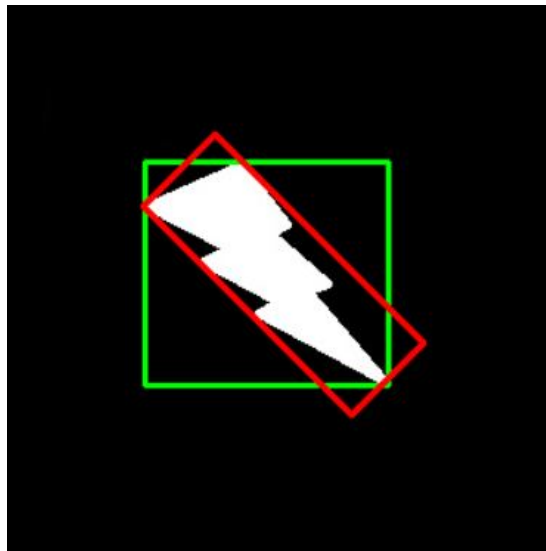


Figura 5.10: Ejemplo de uso de la función *minAreaRect*. En rojo, el rectángulo de área mínimo calculado por la función.

El ángulo calculado por esta función estará siempre entre  $[-90,0)$ , debido a que si el ángulo es mayor que 90, la función elegirá el siguiente lado en sentido horario para realizar el cálculo[24].



Con el ángulo obtenido gracias a *minAreaRect* se puede calcular la **matriz de transformación** para realizar la rotación al fémur. Una vez aplicada, la imagen resultante (figura 5.11) es una imagen que contiene una mitad umbralizada, y **rotada** en función del ángulo del fémur. Si se le volviese a aplicar la función *minAreaRect*, obtendríamos un rectángulo con dos lados paralelos a la horizontal de la imagen, y los otros dos lados paralelos a la vertical de la imagen.

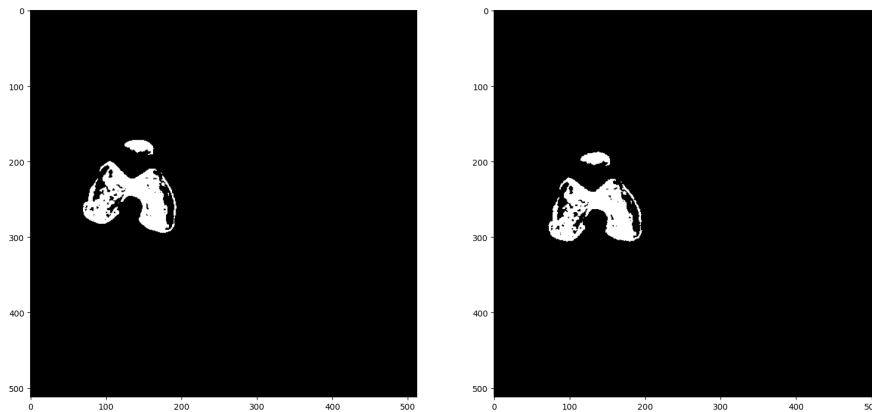


Figura 5.11: Ejemplo de uso de la función *rotateFemur*. A la izquierda, la imagen, cuya mitad derecha ha sido eliminada, antes de la ejecución de la función. A la derecha, el resultado de la ejecución de la función, con los contornos ya rotados.

### 5.3.3.1.3 Funciones *rotateTibia* y *rotateRotula*

Estas dos funciones servirán para, igual que *rotateFemur*, rotar la imagen en función de los ángulos de la **tibia y la rótula**.

La función de *rotateFemur* funciona también para rotar la imagen en función del ángulo de la tibia, así que *rotateTibia* se limita simplemente a llamar a la función *rotateFemur*.

En el caso de la función *rotateRotula*, el algoritmo es el mismo que el utilizado en *rotateFemur*, con una única diferencia: cuando en *rotateFemur* se escoge el área más grande para realizar los cálculos posteriores, en *rotateRotula* se escoge el **segundo área más grande**, utilizando la información a priori de que el contorno de la rótula será siempre el segundo contorno más grande de la imagen.

#### 5.3.3.1.4 Funciones *getDrawedTAGTImage* y *GetDrawedBRImage*

Estas funciones pintarán las líneas y puntos que conforman las medidas TA-GT y Básica Rotuliana, respectivamente. El método de obtención de estos puntos será explicado en la siguiente sección.

Estas funciones reciben como entrada la imagen sobre la que se pintarán las líneas, y las coordenadas de estas líneas. En el caso de TA-GT, pasan como argumentos de entrada los puntos calculados de los cóndilos, la garganta troclear y la TAT. En el caso de Básica Rotuliana, pasan como argumentos de entrada los puntos de los cóndilos y la rótula.

En el caso de *getDrawedTAGTImage*, cómo la imagen de entrada ha de ser las imágenes de la sección fémur-rótula y tibia superpuestas, esa imagen ha de obtenerse previamente almacenando en una variable la suma de los tags *Pixel Array* de ambas imágenes.

Para el pintado de las líneas, se utilizan la función *line* de OpenCV.

#### 5.3.3.2 Módulo *interesting-points*

En esta iteración se implementaron funciones para obtener los siguientes puntos de interés de las diferentes regiones óseas:

- Para el **fémur**, obtener los puntos correspondientes al borde posterior de los **cóndilos**, y el punto más profundo de la **tróclea**.
- En el caso de la **rótula**, obtener los puntos pertenecientes al **eje transversal**.
- En la **tibia**, obtener el punto más saliente de la **TAT**.

##### 5.3.3.2.1 Función *getPointsCondiles*

En esta función se extraen los puntos del borde posterior de los cóndilos del fémur. El primer paso del algoritmo consiste en rotar la imagen umbralizada que recibe como entrada utilizando la función *rotateFemur*. En esta imagen rotada se le aplica la función *findContours* para extraer el contorno del fémur, y se llama a la función *getROI\_Femur* de *interesting-points* para obtener una imagen que contenga únicamente el fémur. Con el contorno del fémur, se calcula el punto más occidental y más oriental del contorno, y se calcula el punto que se encuentre en el medio de esos dos puntos.

Todos los puntos que se encuentren en la misma columna que el punto calculado serán borrados de la imagen, es decir, sus píxeles se pondrán todos a valor 0. Con esto, obtenemos una imagen del fémur "**partida**", donde cada uno de los cóndilos se encuentra en una mitad de la imagen. Con este fémur "partido" en dos se vuelve a aplicar la función *findContours*, obteniendo los contornos de cada una de las mitades del fémur.

Con estos contornos calculados, la obtención de los puntos de los cóndilos es sencilla. Basta con obtener el **punto más al sur** de cada una de las mitades. Estos dos puntos forman

parte de la línea tangencial al borde posterior de ambos cóndilos que es necesaria para calcular las medidas TA-GT y Básica Rotuliana.

#### 5.3.3.2.2 Función *getPointsTrochlea*

Esta función permite obtener el punto dónde se encuentra la parte más profunda de la troclea. La metodología utilizada para implementar el algoritmo es similar a la de *getPointsCondiles*. El algoritmo es el siguiente:

1. Se obtiene una **imagen rotada** utilizando la función *rotateFemur*. Se guarda una copia de esa imagen en otra variable.
2. Se separa el fémur en dos mitades, aplicando el mismo procedimiento que en *getPointsCondile*, en la imagen copiada. Para cada una de estas mitades, se calcula el punto más **septentrional**. El punto norte del lado izquierdo lo llamaremos noroeste, y el del lado derecho, noreste. Se calculan también los puntos más meridionales de cada mitad, y se calcula la distancia en el medio de los valores norte y sur de cada mitad, y se obtienen los puntos que se encuentran a la mitad de esa distancia.
3. Se **eliminan** todos los píxeles de la imagen (figura 5.12) que no cumplan la siguiente condición: han de estar en una columna a la izquierda de noreste, y a la derecha de noroeste. Además, deben estar por encima del punto medio de la distancia norte-sur.
4. A la imagen resultante del paso anterior, se le aplica la función *findContours*.
5. Utilizando la función de OpenCV *pointPolygonTest* que permite saber para un píxel dado, y un contorno dado, si el píxel se encuentra dentro, fuera del contorno, o forma parte de él, se puede conseguir el punto deseado de entre varios puntos candidatos aplicando el siguiente criterio:
  - El punto ha de formar parte del contorno.
  - El píxel situado una columna más **arriba** ha de tener su valor a 0.
  - El píxel situado una columna más **abajo** ha de tener su valor a 1.
  - El punto elegido es el punto situado en la **columna más abajo** de entre todos los píxeles candidatos.

#### 5.3.3.2.3 Función *getPointsTibia*

Esta función permite obtener el punto más saliente de la TAT en imágenes de tibia. El procedimiento consiste en rotar la imagen a través de la función *rotateTibia*, obtener su contorno con *findContours* y de ese contorno, el punto deseado es el punto más al norte del contorno.

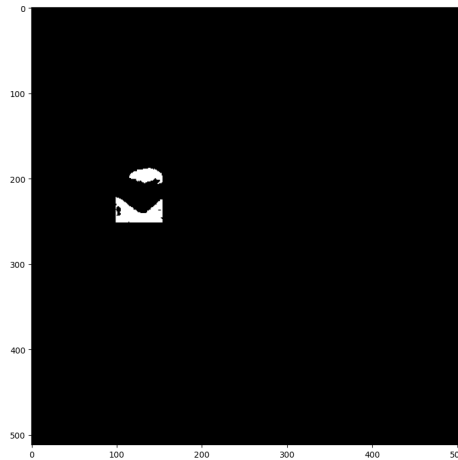


Figura 5.12: Imagen obtenida tras la ejecución del paso 3 de la función *getPointsTrochlea*.

#### 5.3.3.2.4 Función *drawPoints*

Con esta función es posible obtener una imagen con los puntos obtenidos pintados en colores para que sea sencillo visualizarlos en pantalla.

#### 5.3.3.2.5 Función *getPointsTibia*

Esta función permite obtener el punto más saliente de la TAT en imágenes de tibia. El procedimiento consiste en rotar la imagen a través de la función *rotateTibia*, obtener su contorno con *findContours* y de ese contorno, el punto deseado es el punto más al norte del contorno.

#### 5.3.3.3 Módulo *dicom-utils*

En este módulo se añaden a las funciones ya existentes funciones de cálculo matemático básico como son el obtener el ángulo que forman dos rectas secantes o la distancia entre dos rectas paralelas, así como las funciones que permiten obtener los cálculos de Básica Rotuliana y TA-GT.

#### 5.3.3.4 Módulo *measures*

En este nuevo módulo se implementan las funciones que permiten obtener el resultado de las medidas anatómicas. Las medidas a implementar son TA-GT y Básica Rotuliana, pudiendo obtener este cálculo en la región izquierda o derecha de las imágenes.

#### 5.3.3.4.1 Función *getBasRot*

Esta función calcula el ángulo obtenido tras el cálculo de la Básica Rotuliana. Para comprender el proceso, se realizará una explicación sencilla del procedimiento de cálculo de Básica Rotuliana:

1. Se traza una línea tangencial al borde posterior de ambos cóndilos.
2. Se traza una segunda línea en el eje transversal de la rótula.
3. Se traza una tercera línea paralela a la primera, y se calcula el ángulo que forma con la segunda línea.

El algoritmo de esta función es similar al procedimiento aquí mostrado, siguiendo los siguientes pasos:

1. Se obtienen los puntos de los cóndilos con la función *getPointsCondiles* y se calcula la función de la línea que pasa por esos dos puntos.
2. Se obtienen los puntos que pasan por el eje transversal de la rótula con la función *getPointsRotula* y se calcula función de la línea que pasa por esos dos puntos
3. Se calcula el ángulo que forman las dos líneas, y se devuelve ese valor como resultado.

#### 5.3.3.4.2 Función *getTA-GT*

Esta función realiza el cálculo de la distancia TA-GT. Igual que con la Básica Rotuliana, primero se explica el procedimiento médico a seguir para calcularla:

1. Se traza una línea tangencial al borde posterior de ambos cóndilos.
2. Se calcula el punto más profundo de la tróclea, y el punto más saliente de la TAT.
3. Se visualizan las dos imágenes unidas, una sobre la otra, y se trazan dos líneas perpendiculares a la línea de los cóndilos: la primera pasa por el punto más profundo de la tróclea, y la segunda por el punto más saliente de la TAT.
4. Se realiza el cálculo de la distancia entra las dos líneas, cuyo resultado es la distancia TA-GT.

En este caso, el algoritmo de la función también sigue un procedimiento similar, con los siguientes pasos:

1. Se obtienen los puntos de los cóndilos con la función *getPointsCondiles* y se calcula la función de la línea que pasa por esos dos puntos.

2. Se obtienen el punto correspondiente a la parte más profunda de la tróclea, y el punto correspondiente a la parte más saliente de la TAT.
3. Se calculan las líneas perpendiculares a la primera que pasen por los puntos calculados en el paso anterior, y se calcula la distancia entre esas dos líneas, dando como resultado la distancia TA-GT.

### 5.3.4 Pruebas

En esta iteración se diseñaron unos test para el modelo de datos. Se implementaron utilizando el módulo *unittest* de Python. En concreto, se realizaron pruebas para los nuevos atributos y funciones de *FemRotImage* y *TibiaImage*, y para la nueva clase *Patient*.

Además, se realizaron unas pruebas de ejecución para evaluar la precisión en el cálculo de las medidas TA-GT, comparado los resultados obtenidos por el sistema con los obtenidos por personal médico realizando los cálculos manualmente, utilizando los estudios médicos de prueba proveídos por personal médico.

Estudio	Valor TA-GT Manual	Valor TA-GT Sistema	Error
Estudio 1 (Izquierda)	6.6 mm	5.5 mm	1.1 mm
Estudio 1 (Derecha)	12.1 mm	11.3 mm	0.8 mm
Estudio 2 (Izquierda)	13.5 mm	12.3 mm	0.8 mm
Estudio 2 (Derecha)	11.2 mm	10.5 mm	0.7 mm
<b>Media</b>	<b>10.85 mm</b>	<b>9.9 mm</b>	<b>0.85 mm</b>

Tabla 5.1: Resultados del cálculo TA-GT automático frente al cálculo manual

Los resultados de esta prueba, contemplados en la tabla 5.1, arrojan un resultado satisfactorio. En ninguna de las pruebas realizadas el margen de error supera el umbral de 4 mm que se establece como error máximo tolerable en las medidas TA-GT.

Para probar la ejecución de medidas Básica Rotuliana no fue posible llevar a cabo un proceso similar a causa de la imposibilidad de obtener estudios de prueba que sirvieran de valores de referencia. Por tanto, para probar su correcto funcionamiento, se realizaron pruebas de ejecución en diferentes imágenes de estudios, comprobando que la extracción de puntos característicos y el trazado de líneas fuese correcto, y que el resultado del cálculo de la medida entrase dentro de un rango de valores coherente con la medida correspondiente.

## 5.4 Tercera iteración

El desarrollo de la tercera iteración corresponde a las primeras funcionalidades de la interfaz gráfica del sistema. En concreto, se desarrollan las siguientes de funcionalidades:

- Pantalla de carga de las imágenes **DICOM**.
- Menú de selección de medidas a realizar (TA-GT o Básica Rotuliana, escogiendo la mitad izquierda o derecha de las imágenes, o ambas mitades).

### 5.4.1 Análisis

De manera previa al desarrollo de la tercera iteración del proyecto, se realizó un análisis de requisitos. Con el resultado del análisis de requisitos se establecieron los casos de uso a desarrollar. En este apartado se explican en profundidad todos los detalles.

#### 5.4.1.1 Análisis de requisitos

La recogida de requisitos para esta iteración dio como resultado la siguiente lista:

- Plataforma de carga de los estudios DICOM en formato **.dcm** contenidos en una única carpeta, con las siguientes funcionalidades:
  - Carga de los ficheros introduciendo la ruta del fichero **por teclado**.
  - Carga de los ficheros a través de un **explorador de archivos**.
- **Selector de imágenes**, donde se puede navegar entre todas las imágenes del estudio DICOM cargado, y escoger las imágenes de fémur y tibia óptimas para la realización de medidas.
- **Selector de medidas**, pudiendo elegir en la interfaz entre todas las posibilidades para cada medida. En concreto, se deben implementar las siguientes selecciones:
  - Medida TA-GT: medida en mitad izquierda o derecha de la imagen.
  - Medida Básica Rotuliana: medida en mitad izquierda o derecha de la imagen.
- **Visualizador de imágenes médicas** con todas las líneas y puntos utilizados pintados en la imagen, junto con el resultado final de la **medida realizada**.

#### 5.4.1.2 Casos de uso

En esta sección se detallan los casos de uso para la tercera iteración (figura 5.13).

A continuación, se detallan cada uno de los casos de uso:

- **Carga de estudios:** se implementa una ventana gráfica donde el usuario pueda especificar la ruta de la carpeta donde se almacena el estudio DICOM que se desea cargar. Esta ruta se puede obtener a través de un explorador de ficheros, o introduciendo la ruta por teclado.

- **Selección de imágenes:** se implementa una ventana gráfica donde el usuario pueda desplazarse entre todas las imágenes contenidas en el estudio cargado, escogiendo entre ellas las imágenes que se vayan a utilizar para el cálculo de las medidas.
- **Selección de medida y opciones:** se implementa una ventana gráfica con un diálogo con dos desplegables dónde se especificará, respectivamente, qué medida se desea realizar y en qué sección de la imagen.
- **Visualizar imagen con medidas:** se implementa una ventana gráfica dónde se mostrará la imagen resultante de la selección del usuario, y el resultado obtenido en la medición.

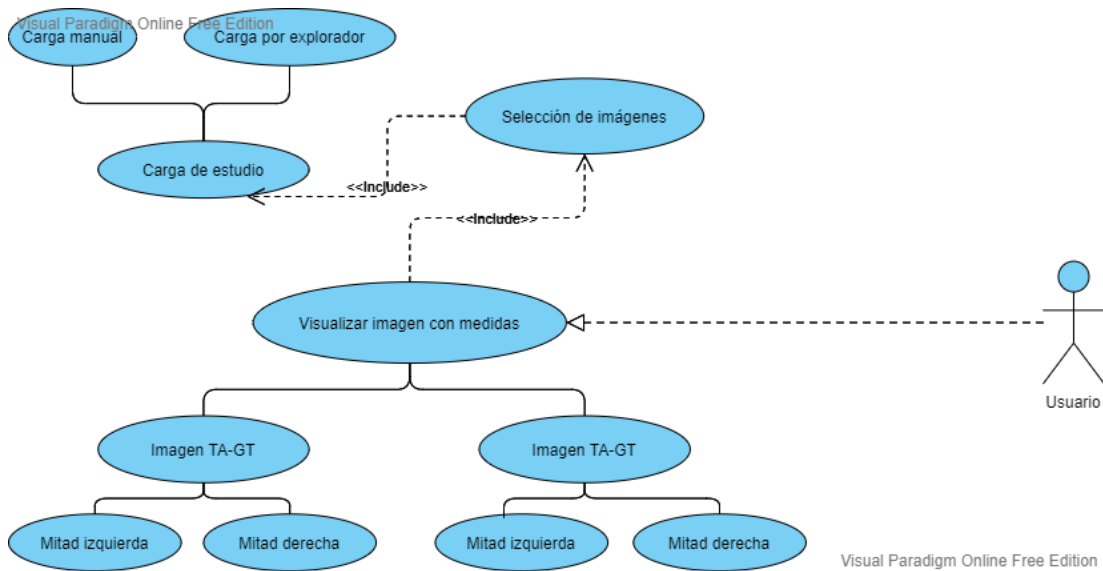


Figura 5.13: Diagrama de casos de uso para la tercera iteración

## 5.4.2 Diseño

En esta sección se detalla el diseño elegido para la implementación de los casos de uso detallados en el apartado anterior.

### 5.4.2.1 Arquitectura diseñada

Para llevar a cabo el desarrollo de la interfaz, primero se diseñó una arquitectura (figura 5.14) para definir los diferentes componentes y poder integrar la interfaz con la lógica de segmentación de imágenes desarrollada en las dos iteraciones anteriores.

Se optó por aplicar el patrón de diseño Modelo-Vista-Controlador, debido a que permite una separación explícita de los componentes de la aplicación y permite la integración de



manera cómoda de las nuevas funcionalidades que se desarrollarán en la cuarta iteración. De esta manera, quedan definidos los siguientes componentes:

- Vista: encargado de la creación de instancias gráficas.
- Modelo: encargado de la creación, modificación y procesado de los datos. Se comunicará con el resto de componentes para transmitir los datos.
- Controlador: encargado de la comunicación entre Modelo y Vista, y de gestionar la llamada a los diferentes módulos de procesado de imagen médica creados en las iteraciones 1 y 2.

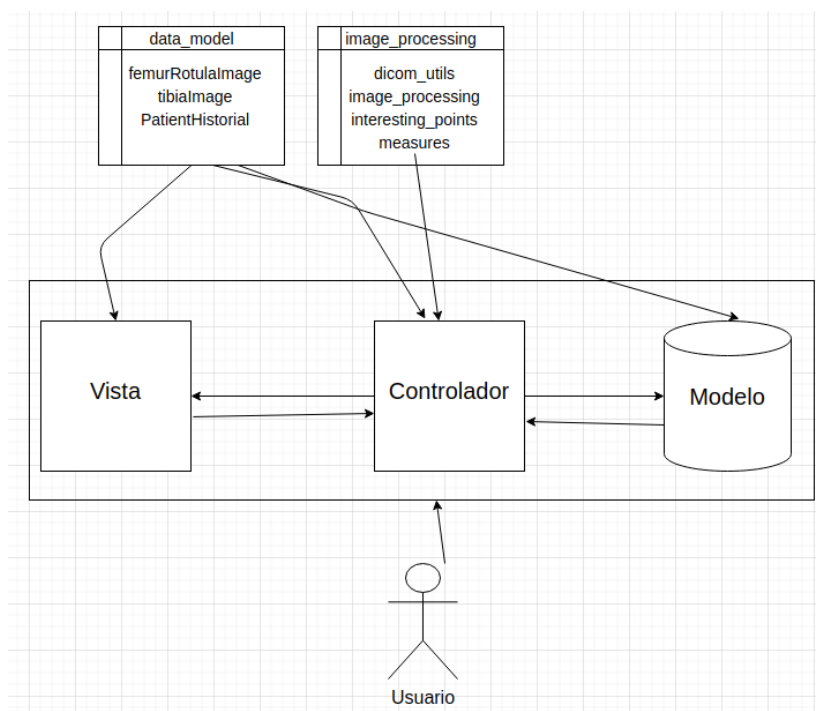


Figura 5.14: Arquitectura diseñada

El modelo de datos será accesible para los tres componentes, con la finalidad de crear y modificar instancias de la clase *PatientHistorial*. Además, los módulos de procesamiento de imágenes se ejecutarán desde el controlador, que enviará los resultados para su visualización a la Vista, y para su guardado al Modelo.

#### 5.4.2.2 Diseño de la interfaz

De manera previa a la implementación, se realizaron unos diseños prototipo que sirvieron como guía a la hora de realizar las diferentes ventanas que genere la interfaz. Tras el desarrollo de esta tarea, el resultado fueron los siguientes diseños:

Se diseñó una ventana de carga de estudios (figura 5.15), dónde se cargarán los ficheros DICOM que contienen las imágenes con las que se realizarán las medidas.

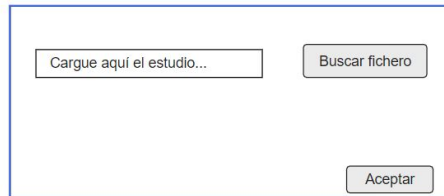


Figura 5.15: Ventana de carga de un estudio DICOM

Tal y como se recoge en la etapa de análisis, el usuario debe de indicar la ruta del estudio de ficheros DICOM, de las que el usuario escogerá una imagen como imagen de la sección del fémur, y otra imagen como imagen de la sección de la tibia.

Para la selección de las imágenes a utilizar en las medidas anatómicas, se diseñó un panel (figura 5.16) en el que el usuario podría navegar entre todas las imágenes pertenecientes al estudio cargado previamente y escoger las que considere óptimas.

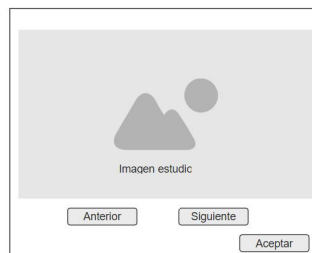


Figura 5.16: Ventana de selección de imágenes

Para la selección de opciones de las medidas, se diseñó una ventana de selección de la medida deseada, y de la región de la imagen dónde se debe realizar la medida.

Para poder escoger las opciones deseadas, se propuso la implementación de dos desplegables en los que escoger, respectivamente, la medida deseada, y la región de la imagen en la que realizar las medidas, tal y como se muestra en la figura 5.17.

Por último, para visualizar los resultados de las medidas, se desarrolla una ventana donde el usuario verá la imagen (figura 5.18) con los puntos de interés y las medidas dibujadas y se imprimirá el resultado de la medición.

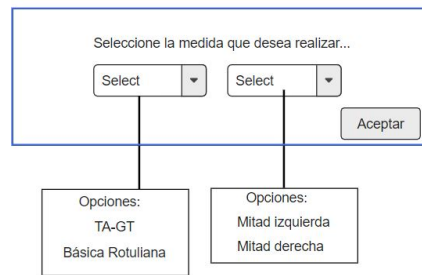


Figura 5.17: Ventana de selección de opciones para la medida a realizar



Figura 5.18: Ventana de visualización de los resultados de la medición

### 5.4.3 Implementación

Para la implementación de la interfaz gráfica, se optó por utilizar la biblioteca de componentes gráficos GTK.

Para agilizar el proceso de diseño de ventanas y conexión entre componentes de la vista, se utilizó la herramienta de diseño de interfaces gráficas Glade para crear las ventanas del sistema y conectar todos los componentes existentes en cada ventana (e.g. botones, entradas de texto).

Para seguir el diseño propuesto utilizando el patrón Modelo-Vista-Controlador, se crearon tres clases con las siguientes funcionalidades:

- **Model:** clase dónde se almacenan los **datos** del paciente.
- **View:** clase dónde se gestiona la llamada a las diferentes **ventanas**, y la gestión de **eventos** (e.g. pulsación de un botón).
- **Controller:** clase encargada de **comunicar** al modelo con la vista.

### 5.4.3.1 Modelo

En esta iteración se implementó una versión simplificada del modelo del sistema. Para la gestión de los datos del paciente, la clase *Model* contendrá como atributo una instancia de la clase *Patient* desarrollada en la segunda iteración del proyecto. De esta manera, el modelo podrá almacenar de manera ordenada los datos del paciente.

### 5.4.3.2 Vista

Este componente se encarga de la creación de ventanas (vistas) y la conexión de los componentes de estas.

Para explicar el flujo de la aplicación y representar las ventanas existentes en el sistema y las transiciones entre ventanas, se realizó el diagrama de flujo mostrado en la figura 5.19. En él, los rectángulos representan cada una de las ventanas implementadas, y las flechas las transiciones entre ventanas. El texto cercano a cada flecha representa el botón pulsado para realizar la transición en concreto.

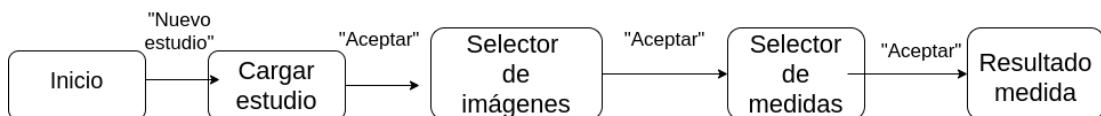


Figura 5.19: Diagrama de flujo de las ventanas

El primer paso de la implementación fue el diseño de las vistas utilizando la herramienta Glade. Una vez finalizado el diseño de las ventanas, se generó un fichero de formato *.glade* que se debe cargar desde el código escrito en Python.

En la clase *View* se carga este fichero al inicializar una instancia. Además de este fichero, se cargan también los objetos que contienen cada una de las ventanas diseñadas en Glade, para poder realizar una gestión centralizada desde la clase *View* de cada una de ellas, pudiendo abrirlas, cerrarlas o destruirlas cuando el flujo de ejecución de la aplicación lo requiera.

Entre los métodos implementados se realiza una separación explícita de los métodos utilizados para la gestión de ventanas y la gestión de eventos, con la siguiente lógica:

- **Métodos de gestión de ventanas:** se encargan de generar, ocultar, visualizar o destruir nuevas vistas (ventanas) y de insertar los datos o imágenes que contengan las vistas. Estas ventanas se generan a través del componente *GTKWindow*.
- **Métodos de gestión de eventos:** se encargan de disparar nuevos eventos tras una acción por parte del usuario (e.g. pulsar un botón) y serán los que llamen a los métodos de gestión de ventanas o recoger los datos contenidos en elementos de la vista. Tienen una firma común con la siguiente forma: `def eventfunction(self,button/widget)`. Los

eventos son lanzados tras la pulsación de botones implementados con el componente *GTKButton*.

De esta manera, cada ventana tendrá asociado de manera única un método de gestión de ventanas, y cada botón estará vinculado a un método de gestión de eventos.

La implementación de las diferentes ventanas en esta iteración y componentes de cada una de las ventanas se realizó de acuerdo con el diseño propuesto en la sección 5.4.2.2.

A continuación se explican los detalles de implementación de cada una de las ventanas implementadas.

#### 5.4.3.2.1 Ventana de inicio

Esta ventana será la primera ventana que se muestre al ejecutar el sistema. En ella, se muestra un pequeño texto de bienvenida y de guía del proceso de medidas automáticas, y dos botones: uno para salir del sistema y finalizar la ejecución, y otro para lanzar la ventana de carga de estudios.

#### 5.4.3.2.2 Carga de estudios

En esta ventana se cargará la ruta en la que se encuentra el estudio médico de imágenes DICOM. Para poder realizarlo, se implementa un botón, que al ser pulsado, lance el componente *GTK.FileChooserDialog*. Este componente lanzará un explorador de ficheros en el que el usuario podrá navegar y escoger la carpeta en la que se encuentra el estudio DICOM. Una vez escogida la ruta, esta se escribirá en un componente *GTKEntry*. En este componente también es disponible escribir la ruta por teclado, si el usuario así lo desea.

Esta ventana contiene un botón de "Aceptar", que lanzará la ventana de selección de imágenes.

#### 5.4.3.2.3 Selección de imágenes

En esta ventana se escogerán las imágenes necesarias para el cálculo de las medidas de entre todas las imágenes halladas en el estudio cargado previamente. Estas imágenes serán mostradas a través del componente *GTKImage*. Para desplazarse entre las imágenes del estudio, se implementan los botones "Anterior" y "Siguiente", que permitirán moverse a la imagen anterior y siguiente del estudio, respectivamente.

En esta ventana, se le pedirá al usuario escoger primero la imagen que mejor muestre la sección de fémur y rótula del estudio cargado. Una vez el usuario encuentre la imagen que considere oportuna, pulsará el botón "Aceptar" y realizará el mismo proceso para la imagen

de la sección de tibia. Tras pulsar nuevamente "Aceptar", se lanzará la ventana de selección de medidas.

#### 5.4.3.2.4 Selección de medidas

Esta ventana permitirá al usuario escoger la medida a realizar y las opciones de medida. Para poder realizar la selección, se implementan dos menús desplegables a través del componente *GTKComboBoxText*. En el primer menú se escogerá la medida deseada (TA-GT, Básica Rotuliana) y la región en la que se realizará la medida (Mitad izquierda, Mitad derecha).

Tras haber realizado la selección, el usuario deberá pulsar el botón "Aceptar" para lanzar la ventana de resultado de la medida.

#### 5.4.3.2.5 Resultado de la medida

Esta ventana permite la visualización por parte del usuario del resultado obtenido por el sistema de cálculo automático de las medidas.

Para visualizar la medida, se utiliza el componente *GTKImage* donde se cargará la imagen resultante del proceso de medida, con los puntos y líneas necesarios para el proceso de cálculo del sistema. Además, se escribirá en la ventana el resultado del cálculo de la medida.

En esta ventana se implementa el botón "Finalizar", que permite al usuario salir del sistema.

#### 5.4.3.3 Controlador

Este componente se encarga de la comunicación entre el modelo y la vista y de la integración de los módulos de procesado de imagen médica implementados anteriormente.

Aplicando esta idea, se implementaron métodos que llamasen al modelo para la generación, modificación y consulta de datos de instancias de la clase *Patient*, y métodos que llamen a los módulos de procesado de imagen médica.

#### 5.4.3.4 Exportación a formato .PNG de estudios DICOM

Durante la implementación de la tercera iteración se detectó un problema con la visualización de imágenes por parte de la librería GTK. El problema detectado consistía en, junto con el problema inicial, del que ya se tenía conocimiento previo, de que GTK no soportaba la visualización nativa de imágenes DICOM en formato .dcm, la visualización deficiente de imágenes cuando estas se cargaban a través de una matriz de píxeles en el componente *GTKImage*.

Para solventar ese problema y poder visualizar correctamente las imágenes se implementó un módulo de Python, con el nombre *export-serie-to-png*. Este módulo contiene dos funciones:

- **exportStudioToPNG():** permite cargar un estudio DICOM contenido en una carpeta y exportar cada una de las imágenes contenidas en el tag *pixel\_array* a un fichero .png.
- **exportImageToPNG():** permite cargar un único fichero DICOM y exportar su tag *pixel\_array* a un fichero .png.

Este módulo es utilizado para cargar las imágenes del estudio DICOM y visualizarlas en la pantalla de selección de las imágenes de fémur y tibia.

#### 5.4.4 Pruebas

Para poder validar el correcto comportamiento de los componentes Controlador y Modelo, se escribieron pruebas unitarias utilizando el módulo *unittest* de Python.

En cuanto a la Vista, se diseñaron y ejecutaron una serie de pruebas funcionales para comprobar que el sistema cumplía con los requisitos contemplados en la etapa de análisis y era capaz de ejecutar todos los casos de uso.

## 5.5 Cuarta iteración

El desarrollo de la cuarta, y última, iteración del sistema tiene como objetivo principal añadir nuevas funcionalidades a la interfaz desarrollada en la tercera iteración. Las principales funcionalidades añadidas a la interfaz son:

- **Persistencia** de datos relativos a estudios médicos
- **Refinado** manual de las **medidas** calculadas automáticamente por el sistema por parte del usuario.

### 5.5.1 Análisis

De manera previa al desarrollo de la cuarta iteración del proyecto, se realizó un análisis de requisitos. Con el resultado del análisis de requisitos se establecieron los casos de uso a desarrollar. En este apartado se explican en profundidad todos los detalles.

#### 5.5.1.1 Análisis de requisitos

Tras la realización de la recogida de requisitos, se extrajo la siguiente lista:

- Sistema de persistencia de estudios DICOM que almacene la siguiente lista de datos para cada estudio:
  - Nombre del estudio, establecido por el usuario
  - Fecha de nacimiento y sexo del paciente
  - Ruta en la que se alojan los ficheros DICOM escogidos para la realización de medidas.
  - Resultados de las diferentes medidas realizadas (TA-GT, Básica Rotuliana) en las diferentes regiones anatómicas (pierna izquierda, pierna derecha).
- Sistema de visualización del listado de estudios almacenados en disco, con la posibilidad de realizar las siguientes funciones:
  - Visualización de la lista completa de estudios almacenados.
  - Creación de nuevos estudios.
  - Eliminación de estudios almacenados.
  - Acceso en detalle a los datos de cada estudio almacenado.
- Sistema de refinado manual de las medidas realizadas automáticamente por el sistema de cálculo automático. El usuario debe tener la posibilidad de :
  - Mover los puntos y líneas calculados por el sistema a una nueva localización en la imagen.
  - Obtener el resultado del cálculo de la medida teniendo en cuenta los puntos modificados por el usuario.
  - Almacenar en memoria el nuevo resultado obtenido.

#### 5.5.1.2 Casos de uso

A continuación se detallan los casos de usos implementados en esta iteración.

- **Añadir nuevo estudio:** el usuario debe introducir los datos personales del paciente al que pertenece el estudio (fecha de nacimiento y sexo), el nombre del estudio y la ruta de la carpeta dónde se encuentra el estudio DICOM.
- **Ver listado de estudios:** se muestra una ventana donde están listados todos los estudios almacenados, identificados por los datos personales del paciente al que pertenecen.
- **Eliminar estudio:** se elimina un estudio del listado de estudios almacenados.



- **Visualizar detalles del estudio:** se muestra una ventana donde están los datos personales del paciente, las imágenes seleccionadas para el cálculo de las medidas, y el resultado de los cálculos.
- **Refinar medida:** se ejecuta la herramienta de refinado manual de las medidas, donde el usuario puede mover las líneas que conforman la medida para realizar un nuevo cálculo de las medidas.



Figura 5.20: Casos de uso para la cuarta iteración

En la figura 5.20 se muestra el diagrama de casos de uso con los nuevos casos de uso añadidos para esta iteración.

## 5.5.2 Diseño

En esta sección se detalla el diseño elegido para la implementación de los casos de uso detallados en la sección anterior.

### 5.5.2.1 Modelo de datos

Para poder adaptar el diseño del modelo de datos al diseño de la persistencia de datos, se realizaron unos ligeros cambios en el modelo de datos.

En concreto, los cambios que se realizaron fueron la adición de cuatro nuevos atributos a la clase *PatientHistorial*, para poder almacenar los resultados del cálculo TA-GT y Básica

Rotuliana en ambas mitades de la imagen. Además, se añaden atributos para almacenar el nombre del estudio, la fecha de nacimiento y el sexo del paciente.

### 5.5.2.2 Persistencia de datos

Para poder escribir en disco los datos manipulados en el sistema y poder mantenerlos una vez termine su ejecución, es necesario implementar algún sistema de persistencia de datos. Debido a que el modelo de datos es bastante sencillo y no hay ninguna relación compleja entre entidades, se descartó la posibilidad de utilizar una base de datos SQL. En su lugar, se optó por utilizar un fichero JSON dónde se almacene una lista con todos los estudios y los datos relativos a estos.

Se añaden campos también para añadir los campos relativos a la información personal del paciente, debido a que en un dominio de aplicación real estos campos vendrían contenidos en algún tag del fichero DICOM, pero debido a que todos los ficheros que se utilizaron para realizar las pruebas estaban anonimizados, los datos de nombre y apellidos de los pacientes no se almacenan en ningún caso. Sí que se almacenan los datos de fecha de nacimiento y sexo del paciente porque pueden ser de relevancia en el diagnóstico, pero su obtención será a través del usuario.

```
"patients": [  
  {  
    "patient_id": "00000001A",  
    "first_name": "Nombre",  
    "last_name": "Apellido1 Apellido2",  
    "age": "01/01/1970",  
    "sex": "Hombre",  
    "img_femur": {  
      "file_femur": "/home/pablo/serie_prueba/femur.dcm"  
    },  
    "img_tibia": {  
      "file_tibia": "/home/pablo/serie_prueba/tibia.dcm"  
    },  
    "results": {  
      "rb_left": "10",  
      "rb_right": "11",  
      "ta_gt_left": "12.3",  
      "ta_gt_right": "11.2"  
    }  
  }  
]
```

Figura 5.21: Estructura diseñada para los objetos JSON

Los objetos JSON utilizados siguen una estructura como la mostrada en el objeto de prueba de la figura 5.21. De esta manera, para las operaciones de lectura, escritura, modificación y borrado de datos en disco se utilizarán objetos como el mostrado.

### 5.5.2.3 Arquitectura diseñada

En esta iteración no se añaden nuevos componentes a la arquitectura (figura 5.22) diseñada en la iteración anterior. El principal cambio en el modelo de la arquitectura es el modelo de persistencia implementado en esta iteración. El componente encargado del acceso directo al componente de persistencia será el modelo del sistema.

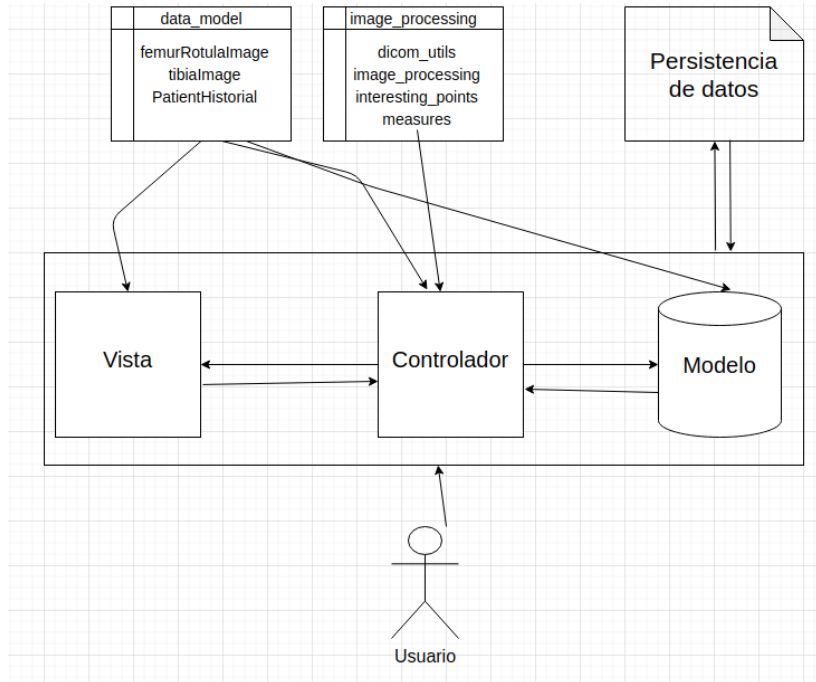


Figura 5.22: Cambios realizados en la arquitectura del sistema

### 5.5.2.4 Diseño de la interfaz

Siguiendo la idea utilizada en la sección 5.4.2.2, para la implementación de ventanas en esta iteración se realizaron nuevos diseños prototipo que sirviesen de guía. En algunos casos los diseños se corresponden a modificaciones de ventanas ya existentes en la iteración 3, y en otros casos se corresponden a nuevas ventanas que se corresponden con las funcionalidades implementadas desde cero en esta iteración.

#### 5.5.2.4.1 Añadir nuevo estudio

Esta ventana es una modificación de la ya existente ventana de carga de un estudio de ficheros DICOM recogida en la figura 5.23. En esta nueva ventana, se añade la posibilidad de introducir los datos personales del paciente al que pertenece el historial de estudio.

Nombre estudio

Fecha de nacimiento

Sexo

Figura 5.23: Ventana de introducción de datos de un nuevo estudio

Esta ventana es una modificación de la ya existente ventana de carga de un estudio de ficheros DICOM recogida en la figura 5.24. En esta nueva ventana, se añade la posibilidad de introducir los datos personales del paciente al que pertenece el historial de estudio.

#### 5.5.2.4.2 Listado de estudios

---

Nombre del estudio	Birth Date	Sexo
Estudio1-21092020	December 10, 1815	M
Estudio2TAGT	December 9, 1906	H
BR-EstudioCHUAC	August 17, 1936	H
0912-M	June 24, 1917	M

Figura 5.24: Ventana de listado de estudios

En esta ventana se visualizará un listado de todos los estudios almacenados anteriormente en el sistema. La tabla de estudios mostrará los datos recogidos en la ventana 5.5.2.4.1 y

permitirá al usuario la opción de añadir nuevos estudios, visualizar en detalle los datos del estudio o eliminar algún estudio almacenado.

#### 5.5.2.4.3 Detalles del estudio

Esta ventana (figura 5.25) mostrará en detalle los datos pertenecientes a un estudio en concreto. En concreto, mostrará todos los datos personales del paciente, las dos imágenes escogidas para el cálculo de medidas y los resultados obtenidos tras el cálculo de las medidas realizadas (si estas han sido realizadas previamente, en caso contrario, lo advertirá con un mensaje de "No realizada todavía" en el lugar que corresponde al resultado de la medida). Además, se incluyen dos botones, uno para acceder al listado de estudios, y otro para desplegar la ventana de selección de medidas.

Imagen fémur	Imagen tibia
Nombre del estudio: CHUAC-0001	TA-GT Izquierda: x mm
Fecha de nacimiento: 10/12/1885	TA-GT Derecha: x mm
Sexo: M	BR Izquierda: x °
BR Derecha: x °	
Listado pacientes	Realizar medidas

Figura 5.25: Ventana de detalles de estudio

#### 5.5.2.4.4 Resultado de la medida

Esta ventana (figura 5.26) permitirá al usuario visualizar el resultado de la medida calculada de manera automática por el sistema. La imagen que muestre tendrá los puntos y líneas necesarios para el cálculo trazados, y mostrará el resultado de la medida en las unidades correspondientes.

También mostrará los datos del paciente, y dispondrá de dos botones, uno para almacenar el resultado de la medida obtenida si el usuario que la medida es correcta, y otro para refinar la medida manual si el usuario observa anomalías en el cálculo realizado.



Figura 5.26: Ventana de resultado de la medida

#### 5.5.2.4.5 Refinado de medidas

En esta ventana (figura 5.27) se realizará el refinado de medidas, permitiendo mover a través del ratón los puntos y líneas que conforman la medida. El funcionamiento de este procedimiento será explicado en profundidad en la sección de Implementación.

Se provee de una herramienta de zoom para que el usuario pueda ampliar las dimensiones de la imagen visualizada en caso de que así lo desee.

Además, se dispone de dos botones; uno para cancelar el procedimiento de refinado, y otro para confirmar el refinado y proceder a calcular las medidas utilizando los puntos aportados por el usuario.

### 5.5.3 Implementación

En esta sección se explican en profundidad todos los detalles de implementación de las funciones y estructuras de datos desarrolladas en la cuarta iteración.

#### 5.5.3.1 Modelo

Como se comenta en la sección de Diseño, el mecanismo de persistencia de datos escogido es la utilización de un fichero JSON.

Se añade al proyecto donde se ubica el código un fichero de recursos llamado *data.json*, en el que se escribirán todos los datos pertenecientes a estudios médicos.



Figura 5.27: Ventana de refinado de la medida

El componente encargado de las tareas de lectura, escritura, borrado y modificación del fichero *data.json* será el componente Modelo.

Para las operaciones de lectura y modificación, el Modelo deberá buscar entre la lista de estudios el estudio deseado utilizando como identificador el nombre del estudio, extraer los datos contenidos en el fichero JSON y crear una instancia de *PatientHistorial* cuyos atributos sean los que contenga el fichero.

De manera análoga, para las operaciones de escritura, el modelo deberá crear un objeto JSON a partir de una instancia de *PatientHistorial* que se le pase como entrada, y añadir ese objeto a la lista de estudios del fichero.

Para ello, utilizando el módulo *json* de Python que permite la lectura/escritura de ficheros JSON, se implementan funciones *getter/setter* de los estudios médicos. Además, se implementan también funciones *getter/setter* para cada uno de los atributos de los estudios (e.g. nombre del estudio, resultado de la TA-GT izquierda).

### 5.5.3.2 Controlador

Debido a los cambios aplicados en el Modelo, es necesario también cambiar la implementación del Controlador. En concreto, se implementan funciones que llamen a las funciones de escritura/lectura de ficheros JSON que son explicadas en la sección anterior. De esta manera, se provee de un punto de acceso a la Vista para que pueda llamar a las operaciones de lectura/escritura de datos sin tener que interactuar con el Modelo, con el fin de preservar la idea principal del patrón MVC.

### 5.5.3.3 Vista

Para explicar el flujo de la aplicación y representar las ventanas existentes en el sistema y las transiciones entre ventanas, se realizó el diagrama de flujo mostrado en la figura 5.28. En él, los rectángulos representan cada una de las ventanas implementadas, y las flechas las transiciones entre ventanas. El texto cercano a cada flecha representa el botón pulsado para realizar la transición en concreto.

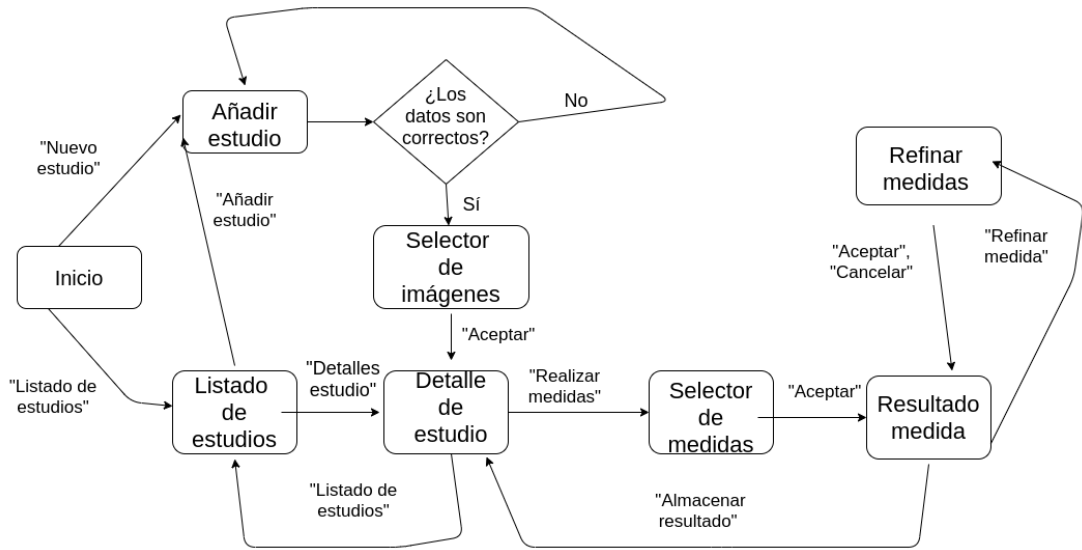


Figura 5.28: Diagrama de flujo de las ventanas del sistema

A continuación se detallan los detalles de implementación de las nuevas ventanas implementadas en esta iteración o las ventanas ya implementadas que sufrieron alguna modificación.

#### 5.5.3.3.1 Nuevo estudio médico

En esta ventana se recogerán el nombre del estudio, los datos personales del paciente (sexo, fecha de nacimiento) al que pertenece al estudio, y se cargará la carpeta que contiene los ficheros DICOM del estudio.

En un dominio de aplicación real no sería necesario introducir los datos del paciente por parte del usuario, puesto que estos datos suelen estar contenidos en los tags del fichero DICOM[25], pero debido a que las imágenes de prueba con las que se trabaja en el desarrollo están anonimizadas y no se dispone de esta información se extraen los datos de esta manera para poder obtenerlos de algún modo.

Se implementa una ventana utilizando una instancia de *GTKWindow*, e insertando una instancia de *GTKEntry* para cada uno de los datos que el usuario introducirá por teclado (nombre



del estudio, fecha de nacimiento) y una instancia de *GTKComboBoxText* para seleccionar el sexo del paciente (hombre, mujer). Para cargar el estudio, se inserta un botón *GTKButton* que, al ser pulsado, carga una instancia de *GTKFileChooserDialog*. Este componente lanza un diálogo de explorador de ficheros del sistema que permite navegar entre las carpetas existentes en el disco y escoger la carpeta donde se encuentra el estudio DICOM que el usuario debe cargar.

Para finalizar el proceso de creación un nuevo estudio, se debe pulsar el botón de "Aceptar". Este botón ejecuta la función de lanzamiento de eventos "createNewPatient()". Esta función llama al controlador para guardar en disco los datos del estudio, y generar una instancia de *PatientHistorial* con los datos aportados por el usuario. De manera previa al guardado del estudio, se comprueba que no existe ningún estudio con nombre de estudio igual al del estudio a insertar, y que todos los datos son correctos (e.g. que los campos no están vacíos, que el formato de fecha sea correcto). Si los datos son correctos, se procede a guardar en disco los datos y crear la instancia de *PatientHistorial*. En caso contrario, se muestra una ventana advirtiendo al usuario del error y pidiéndole corregirlo.

#### 5.5.3.3.2 Listado de estudios

En esta ventana se visualizará el listado de estudios guardados anteriormente.

Para implementar la zona de la ventana en la que se mostrará el listado de estudios se utiliza el componente *GTKScrolledWindow* dentro de la ventana principal, que implementa una ventana con una barra de desplazamiento (*scroll*). Dentro de esta ventana se inserta un componente *GTKTreeView*, un componente diseñado para mostrar listas junto con un componente *GTKListStore*. Para poder obtener la fila seleccionada por el usuario, se utiliza el valor almacenado en el atributo *GTKTreeSelection* de *GTKTreeView*.

En esta ventana se disponen también de tres botones, uno para acceder a la ventana de creación de un nuevo estudio, uno para acceder a la ventana de detalles del estudio del estudio seleccionado, y otro para eliminar del almacenamiento los datos del estudio seleccionado.

#### 5.5.3.3.3 Detalles del estudio

Esta ventana se mostrará tras haber finalizado el proceso de creación de un nuevo estudio, o al pulsar el botón de "Detalles" en el listado de estudios.

En esta ventana se mostrarán todos los datos del estudio y los resultados obtenidos en las medidas anatómicas. En caso de que alguna de estas medidas todavía no hayan sido realizadas, el valor de su resultado será reemplazado por el texto "Todavía no realizado". Además, se mostrarán, cargando las imágenes en instancias de *GTKImage*, las imágenes DICOM escogidas para el cálculo de medidas.

En esta ventana se disponen de dos botones: uno para realizar medidas, que hace mostrar la ventana de selección de medidas, y otro que hace mostrar la ventana de listado de estudios.

#### 5.5.3.3.4 Resultado de la medida realizada

Esta ventana se mostrará tras haber pulsado el botón "Aceptar" en la ventana de selector de medidas.

En ella, se mostrarán los datos del estudio, la imagen resultado del cálculo con todos los puntos y líneas calculados por el sistema trazados, y el resultado calculado por el sistema en las unidades correspondientes.

En esta ventana se dispondrán de dos botones: uno para almacenar el resultado, que hará mostrar la ventana de detalles del estudio, y otro para realizar un refinamiento de las medidas por parte del usuario.

#### 5.5.3.3.5 Refinar medidas manualmente

Esta ventana se mostrará tras haber pulsado el botón "Refinar medidas" en la ventana de resultado de la medida.

Para poder implementar el sistema de refinado, es necesario capturar los eventos de pulsado de botones de ratón. Para ello, se utiliza el componente *GTKEventBox* y dentro de él se inserta una imagen a través de una instancia de *GTKImage*.

Inicialmente, todas las líneas están pintadas en color azul (figura 5.29). Para poder seleccionar una línea, se utiliza el botón izquierdo del ratón. Esta acción está vinculada al *GTKEventbox*. Para poder saber qué línea es la que selecciona el usuario, se utiliza la función *getNearestPoints()*, que implementa la siguiente lógica:

- Se obtiene una lista de tuplas, en la que cada tupla contiene los dos extremos de cada una de las líneas de la medida.
- Se calcula qué extremo de línea es el que más cerca está del píxel pulsado por el usuario, y se devuelve como resultado el punto extremo de la línea que está más cerca de la línea, y la línea a la que pertenece.

Tras haber calculado cuál es la línea que selecciona el usuario, se pinta esa línea en verde, y el extremo seleccionado se pinta en rojo.

Para cancelar la selección, el usuario debe volver a pulsar el botón izquierdo del ratón. Esto provocará que todas las líneas vuelvan a estar pintadas en color azul.

Para desplazar el extremo de la línea, una vez esté el extremo seleccionado, se debe pulsar el punto destino deseado utilizando el botón derecho del ratón.

En esta ventana se inserta un selector de zoom a través de un *GTKComboBoxText*, que permite al usuario visualizar la imagen en diferentes tamaños.

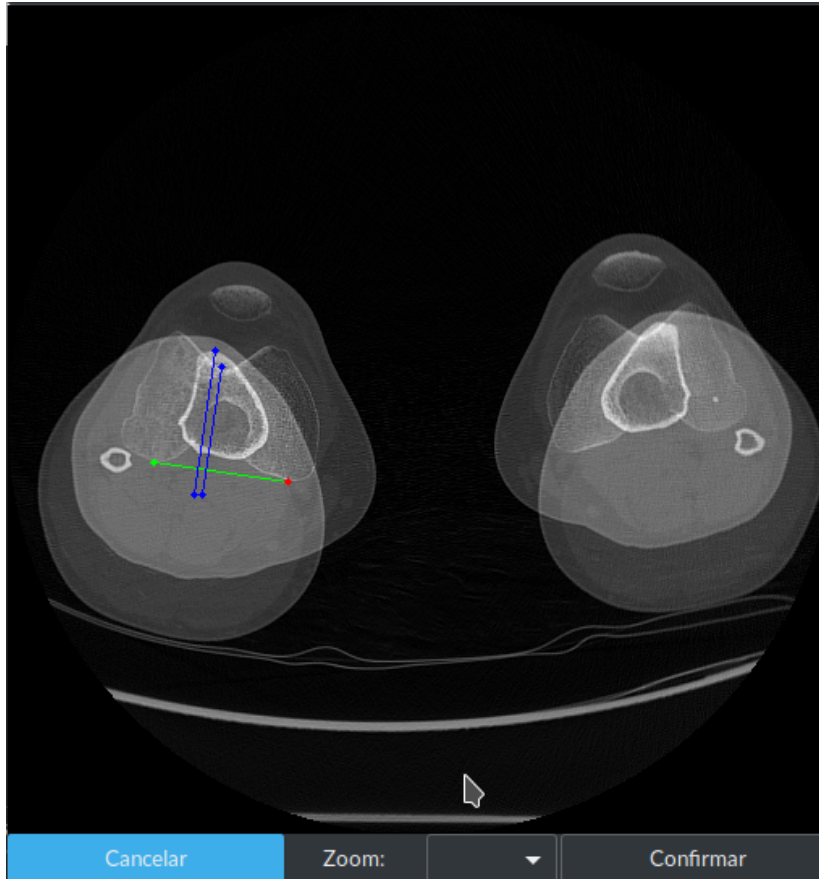


Figura 5.29: Ejemplo de selección de línea en la ventana de refinado

También dispone de un botón para cancelar la acción y retornar a la ventana de resultado de la medida, y un botón para confirmar el refinado, que retornará al usuario a la ventana de resultado de la medida con la imagen actualizada acorde con las líneas modificadas por el usuario, y el resultado obtenido con las nuevas líneas.

#### 5.5.4 Pruebas

Para verificar el correcto funcionamiento de las nuevas funcionalidades implementadas en los componentes Modelo y Controlador, se escribieron nuevas pruebas unitarias utilizando el módulo *unittest* de Python.

En cuanto a la Vista, se diseñaron y ejecutaron una serie de pruebas funcionales para comprobar que el sistema cumplía con los requisitos contemplados en la etapa de análisis y era capaz de ejecutar todos los casos de uso.

# Conclusiones y líneas futuras de trabajo

---

ÚLTIMO capítulo de la memoria, dónde se analizan todas las conclusiones obtenidas a lo largo del desarrollo del proyecto, así como se proponen unas líneas de trabajo futuras que permitan mejorar las cualidades del sistema desarrollado en un futuro desarrollo, y aportar nuevas funcionalidades.

## 6.1 Conclusiones

Con la conclusión del desarrollo de la última iteración del sistema se ha obtenido una solución tecnológica que permita automatizar las medidas anatómicas TA-GT y Básica Rotuliana, principal motivación del desarrollo. Se han podido cumplir satisfactoriamente todos los requisitos propuestos, y se ha desarrollado un sistema que podría ser fácilmente adaptable a otros tipos de medidas anatómicas.

La saturación del sistema sanitario a causa de la pandemia de COVID-19 truncó algunas ideas propuestas de manera previa al comienzo del desarrollo, como eran la obtención de más estudios DICOM que sirviesen de prueba, o la posibilidad del contacto con personal sanitario que probase el sistema desarrollado y aportase *feedback* que se hubiera tenido en cuenta en las diferentes etapas del desarrollo. Pese a esto, se llevó a cabo el desarrollo con los estudios que pudieron ser obtenidos y se logró desarrollar un sistema que aporta unos resultados satisfactorios.

El desarrollo del sistema permitió al alumno tener una experiencia de desarrollo de un proyecto total, involucrándose en todas las tareas y etapas que lo conforman, enriqueciendo su conocimiento y fortaleciendo competencias adquiridas a lo largo de la etapa universitaria.

El desarrollo de un proyecto ubicado en el ámbito de la informática biomédica permitió al alumno adquirir competencias en tecnologías del ámbito médico como el estándar de imagen

DICOM, y debido a que las heurísticas necesarias para desarrollar los mecanismos de detección requerían de un conocimiento experto ajeno a la informática fue necesaria por tanto la adquisición de ciertas competencias básicas en el dominio del problema médico que se trataba de resolver.

La experiencia técnica fue altamente satisfactoria, trabajando en un desarrollo ubicado en diferentes competencias del desarrollo software. El enfoque del proyecto permitió al alumno trabajar en el ámbito de la Visión Artificial en las primeras etapas del proyecto, y en el desarrollo de una interfaz gráfica en las iteraciones finales. Esto permitió al alumno obtener nuevos conocimientos y competencias en estas disciplinas y en las librerías utilizadas en el desarrollo, como OpenCV, Numpy y GTK, y en el lenguaje Python.

Por último, la experiencia de trabajo con el uso de Python como lenguaje de desarrollo fue altamente satisfactoria en las dos primeras iteraciones, cuando el desarrollo estaba enfocado en la herramienta de cálculo automático de medidas. La integración de OpenCV en Python y diferentes librerías como Numpy o Pydicom permitieron un desarrollo cómodo y con una curva de aprendizaje equilibrada teniendo en cuenta la dificultad del problema a resolver. Para el desarrollo de la interfaz gráfica se optó por continuar utilizando el lenguaje Python junto con la librería GTK debido al dominio del lenguaje por parte del alumno y a que permitiría una integración con el código anteriormente desarrollado de manera muy simple. Sin embargo, tras haber concluido el desarrollo, cabe la duda sobre si esta decisión fue la óptima, debido a que el funcionamiento de los componentes de GTK necesarios en la implementación del refinado de las medidas o de la visualización de imágenes no eran los más apropiados para las necesidades concretas del desarrollo, y la documentación encontrada en internet solía ser escasa y en muchas ocasiones escrita para otros lenguajes de programación.

## 6.2 Líneas futuras de trabajo

El principal objetivo que deben cumplir las líneas futuras del desarrollo debe ser la selección automática de las imágenes DICOM que el sistema utilice para el cálculo automático de las medidas. La selección automática de las imágenes óptimas es la única de las tareas del procedimiento de medida que la solución desarrollada en este proyecto no automatiza, y su implementación permitiría agilizar todavía más el proceso y evitar posibles errores por parte del personal médico. Esto podría ser desarrollado a través de un sistema basado en Visión Artificial que, a través de unos criterios morfológicos de los huesos encontrados en cada imagen, elige cuál es el que mejor se ajusta de entre toda la serie de imágenes de un estudio DICOM. También podría ser desarrollado utilizando alguna solución basada en Deep Learning si se dispusiese de un conjunto de estudios de prueba suficientemente grande como para poder llevarlo a cabo.

Una vez desarrollada esta funcionalidad, se abren dos líneas paralelas por las que se podría seguir trabajando, bien en una de ellas, o en ambas:

- La implementación de nuevas medidas anatómicas, ampliando la funcionalidad del sistema a otras regiones de la anatomía humana, o trabajando con otro tipo de imágenes médicas, como radiografías o ecografías.
- El desarrollo de una plataforma de subida de estudios DICOM a un servidor web, pudiendo almacenar de manera remota y compartir los datos y las imágenes generadas por el sistema.



# **Apéndices**





# Guía de instalación

---

## A.1 Instalación del sistema

En este apartado se detallan los pasos necesarios para instalar todos el software necesario para la ejecución del sistema.

### A.1.1 Instalación en Ubuntu

Para instalar todo el software necesario en Ubuntu, se provee de un script de instalación para realizar este proceso de manera cómoda. Este script está en la raíz del proyecto con nombre `install.sh`, y para ejecutarlo es necesario seguir los siguientes pasos:

- Abrir una consola de comandos y situarse en la carpeta raíz del proyecto.
- Dar permisos de ejecución al archivo con el siguiente comando: `chmod +x install.sh`
- Ejecutar el script con el siguiente comando: `.\script.sh`

### A.1.2 Instalación en Windows

Esta guía de instalación ha sido desarrollada y probada para su uso en Windows 10. Su uso en versiones anteriores de Windows no ha sido probado y, por tanto, no se garantiza su correcto funcionamiento.

#### A.1.2.1

Instalación de Python El primer paso para poder instalar el proyecto en nuestro equipo es la instalación de Python. Para ello, se recomienda uno de los siguientes métodos.

- Descargar el instalador en la web <https://www.python.org/downloads/> y ejecutar el instalador descargado.

- Instalarlo a través de la tienda de aplicaciones Microsoft Store.

El sistema no es compatible con versiones de Python anteriores a 3.0, por tanto, se recomienda su instalación en versiones superiores a esta.

### A.1.2.2 Configurar variables de entorno

En caso de que el instalador no haya añadido la ruta de instalación de Python a las variables de entorno del sistema, el usuario debe introducirla manualmente.

Para realizarlo, debemos buscar la ruta donde se haya instalado Python en el sistema.

Una vez obtenida esta ruta, en el Panel de Control, en la sección de Sistema, se debe ejecutar la opción "Editar las variables de entorno del sistema" como administrador. A continuación, se debe pulsar el botón "Variables de entorno", y seleccionando la variable "Path", pulsar el botón "Editar". En esta variable se debe añadir la ruta de instalación de Python, y guardar pulsando el botón "Aceptar".

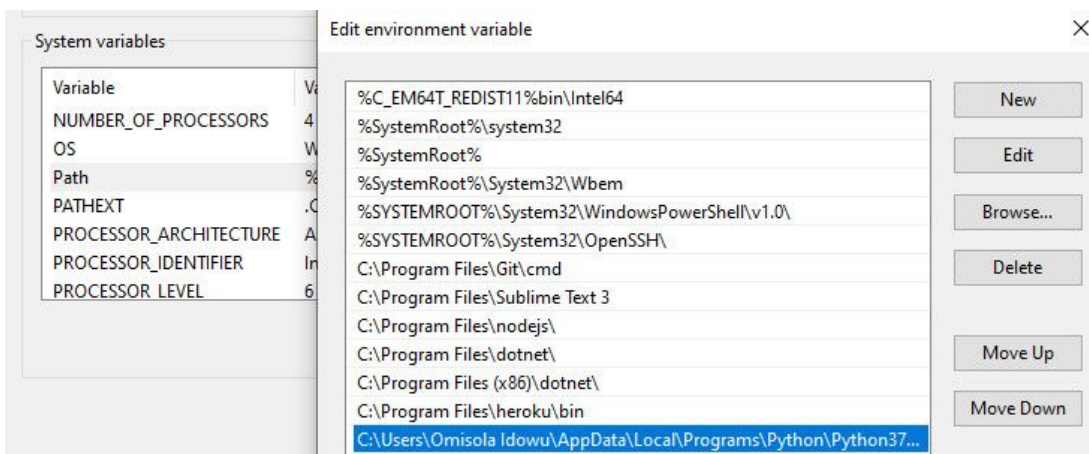


Figura A.1: Configuración de las variables de entorno

Para comprobar que la variable de entorno está bien añadida, se puede introducir el comando `python --version` en una consola de comandos cmd que deberá devolver una salida como la mostrada en la figura A.2.

### A.1.2.3 Instalación de módulos con PIP

Pip es un software de gestión de paquetes para la instalación de paquetes de software de Python. Con pip es posible instalar todos los módulos de Python necesarios para ejecutar el sistema.

La sintaxis de pip para instalar un módulo es la siguiente:

- `pip install <nombre del módulo>`

```
C:\Users\pablo.valeiropena>python --version
Python 3.8.6

C:\Users\pablo.valeiropena>
```

Figura A.2: Comprobación de la versión de Python instalada

Para ejecutar el sistema, se deben ejecutar los siguientes comandos de instalación de pip:

- `pip install numpy`
- `pip install pydicom`
- `pip install opencv-python`

## A.2 Ejecución

Para ejecutar el sistema, se debe abrir una consola de comandos y situarse en la carpeta `interface/src` y ejecutar el comando `python main.py`. Tras ejecutar este comando, se iniciará el sistema.



## Apéndice B

# Guía de uso

---

### B.1 Introducción

El sistema de cálculo automático de medidas anatómicas en imágenes TAC de rodilla es un sistema multiplataforma que permite al usuario la automatización del cálculo de medidas TA-GT y Básica Rotuliana. Entre sus principales funcionalidades se encuentran las siguientes:

- Creación, almacenamiento y listado de estudios DICOM, permitiendo al usuario mantener un historial de todos los diagnósticos realizados por el sistema.
- Cálculo automático de medidas TA-GT y Básica Rotuliana, dotando al sistema de las herramientas necesarias para la visualización de las imágenes producto del cálculo y del resultado aportado por el sistema.
- Sistema de refinado manual de las medidas, aportando la posibilidad de realizar las medidas de manera manual por parte del usuario cuando este considere que las realizadas de manera automática por el sistema no han sido correctas.

### B.2 Compatibilidad del sistema

El sistema está creado como una aplicación de escritorio compatible con los principales sistemas operativos, como Windows, Linux o MacOS.

### B.3 Flujo de trabajo

En esta sección se presentan las directrices para ayudar al usuario en el uso del sistema, siguiendo el flujo de trabajo ilustrado con ejemplos.

### B.3.1 Ventana de bienvenida

Esta ventana (figura B.1) es la ventana que el sistema carga cuando comienza la ejecución del sistema. En ella, encontramos tres botones:

- **"Nuevo estudio"**: el sistema carga la ventana de creación de nuevos estudios.
- **"Listado de estudios"**: el sistema carga la ventana de listado de estudios almacenados.
- **"Salir"**: el sistema finaliza su ejecución y se cierra la ventana de bienvenida.



Figura B.1: Ventana de bienvenida

### B.3.2 Añadir nuevo estudio

En esta ventana se introducen los datos personales del paciente al que pertenece el estudio, y la ruta de la carpeta donde está almacenado el estudio DICOM.

En esta ventana (figura B.2) tendremos que añadir los siguientes datos del paciente:

- **Nombre del estudio:** este valor será el identificador único para cada estudio. Este campo ha de ser único para el estudio, y en caso de que el valor introducido coincida con algún valor almacenado en otro estudio, se mostrará una ventana advirtiéndolo al usuario y pidiendo que revise el valor introducido.
- **Fecha de nacimiento:** debe introducirse la fecha de nacimiento del paciente en el formato "día/mes/año". En caso de introducir un valor inválido, se mostrará una ventana advirtiéndolo al usuario y pidiendo que revise el valor introducido.
- **Sexo:** se dispone de un menú desplegable para seleccionar entre las opciones el sexo del paciente. Pulsando con el botón izquierdo del ratón, se mostrará el menú desplegable y para escoger la opción deseada, el usuario debe pulsar con el botón izquierdo del ratón en esta opción.
- **Carga del estudio:** para poder cargar la ruta de la carpeta que contenga el estudio DICOM, se dispone de un botón de "Explorar". Al pulsar este botón, se mostrará una ventana de explorador de ficheros. En esta ventana se debe buscar entre los ficheros del sistema la carpeta que se debe cargar. Los ficheros del estudio deben estar en formato **.dcm**. En caso de que el usuario prefiera introducir por teclado la ruta, también existe esa posibilidad. En caso de introducir una carpeta que no exista, se mostrará una ventana advirtiéndolo al usuario y pidiendo que revise el valor introducido.

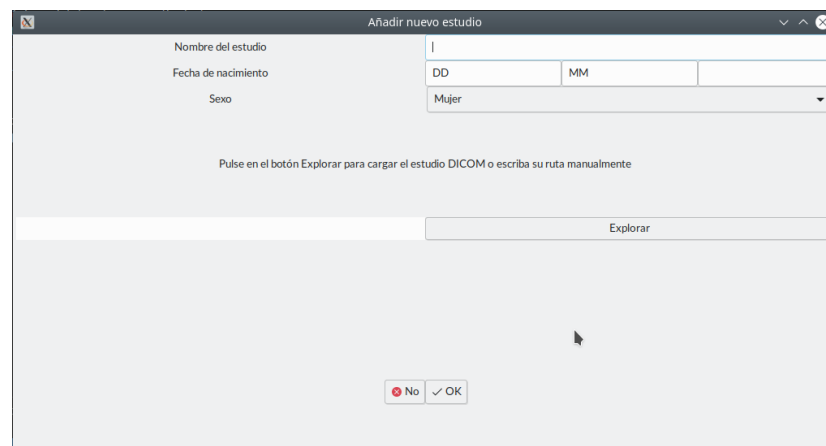


Figura B.2: Ventana de nuevo estudio

En el borde inferior de la ventana se encuentran dos botones:

- "Cancelar": cancela la operación y vuelve al usuario a la ventana de Bienvenida.
- "Aceptar": comprueba que todos los datos sean correctos y almacena el estudio. En caso



de que alguno de los datos introducidos no sean correctos, se muestra la ventana de advertencia mencionada en los campos a introducir.

Si los datos introducidos por el usuario son correctos y ha pulsado el botón aceptar, se cargará la ventana de selección de imágenes.

### B.3.3 Selección de imágenes

Esta ventana se mostrará tras haber confirmado la creación de un nuevo estudio en la pantalla de añadir un nuevo estudio. En esta ventana se mostrarán todas las imágenes DICOM .dcm que el sistema haya encontrado en la carpeta introducida por el usuario. Para poder navegar entre las imágenes, se disponen un botón en cada lateral de la ventana, "Anterior" y "Siguiente", que permiten al usuario visualizar la anterior y siguiente imagen del estudio, respectivamente.

En esta ventana se deberán escoger las dos imágenes necesarias para el cálculo de las medidas. Primero ha de escogerse la imagen que mejor muestre la sección fémur-rótula, y tras haber escogido esta imagen, se debe escoger la imagen de la sección tibia.

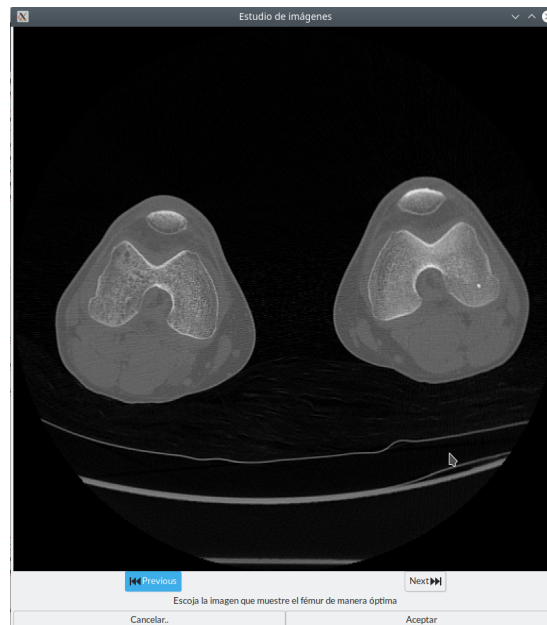


Figura B.3: Ventana de selección de imágenes

Para poder confirmar o cancelar la selección se disponen de dos botones en el extremo inferior de la ventana:

- **"Cancelar"**: en caso de que el usuario haya confirmado por error los datos del estudio introducido anteriormente, dispone de esta opción para cancelar la operación y volver

a la ventana de añadir nuevo paciente. También, en caso de haber confirmado erróneamente la imagen de femur, puede pulsar este botón, que le permitirá cancelar la selección y comenzar de nuevo.

- **"Aceptar"**: cuando el usuario considere que la imagen que está visualizando es la adecuada para la selección de las medidas, en cada una de las dos imágenes que debe seleccionar, deberá pulsar este botón para confirmar la selección.

Una vez se hayan seleccionado las dos imágenes sobre las que realizar las medidas, se mostrará la imagen de detalles del estudio, mostrando los datos introducidos en pasos anteriores.

### B.3.4 Detalles del estudio

En esta ventana (figura B.4) se mostrarán los datos personales del paciente al que pertenece el estudio, y las dos imágenes escogidas para realizar las medidas. Además, se muestran los resultados obtenidos en las medidas realizadas. En caso de que alguna de estas medidas no haya sido realizada todavía, en lugar del resultado obtenido, se mostrará el texto "No realizada todavía".

En esta ventana se dispone de dos botones:

- **"Realizar medidas"**: en caso de que el usuario quiera realizar alguna medida en este estudio, podrá pulsar este botón. Tras haber pulsado el botón, se abrirá la ventana de selección de medidas, para escoger la medida que el usuario desee realizar.
- **"Listado de pacientes"**: cuando el usuario no quiera realizar ninguna medida, podrá pulsar este botón, que hará mostrar la ventana de listado de pacientes.

### B.3.5 Selección de medidas

Esta ventana (figura B.5) se abrirá tras haber pulsado el botón "Realizar medidas" por parte del usuario en la ventana detalles del estudio. En esta ventana se disponen de dos menús desplegables, que contienen los siguientes campos:

- El desplegable de la izquierda contiene las medidas posibles de realizar por el sistema. Pulsando con el botón izquierdo del ratón se muestran las siguientes opciones:
  - **"TA-GT"**: en caso de querer realizar esta medida, el usuario debe pulsar en esta opción.
  - **"Básica Rotuliana"**: ídem para esta opción.
- El desplegable de la derecha contiene las opciones posibles de realizar en cada medida. Pulsando con el botón izquierdo se muestran las siguientes opciones:

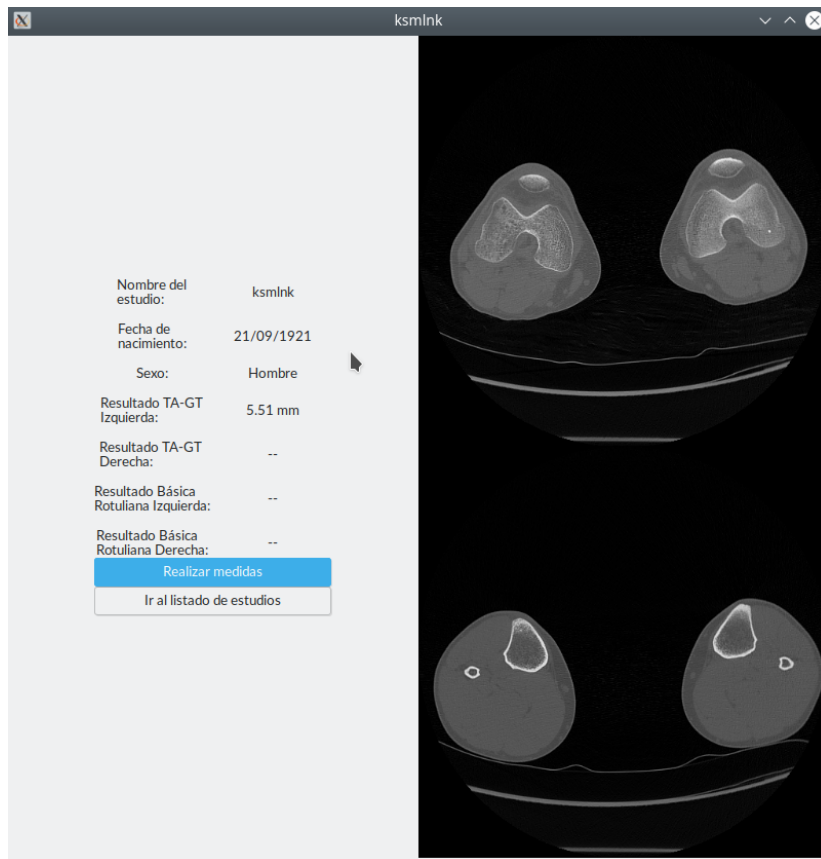


Figura B.4: Ventana de detalles del estudio

- **"Mitad izquierda"**: en caso de querer realizar la medida escogida en la pierna izquierda del paciente, se debe escoger esta opción, que realizará las medidas en la mitad izquierda de la imagen.
- **"Mitad derecha"**: de manera análoga a la anterior opción, en caso de querer realizar la medida escogida en la mitad derecha del paciente, se debe escoger esta opción.

Esta ventana dispone de dos botones en el borde inferior:

- **"Cancelar"**: al pulsar este botón, se cancelará la selección realizada y se cerrará la ventana, devolviendo al usuario a la ventana de detalles del paciente.
- **"Aceptar"**: al pulsar este botón, se confirmará la operación y se ejecutará el proceso de cálculo de la medida seleccionada. Tras haber finalizado el cálculo, se mostrará la ventana de resultado de la medida, que permitirá al usuario visualizar el resultado.

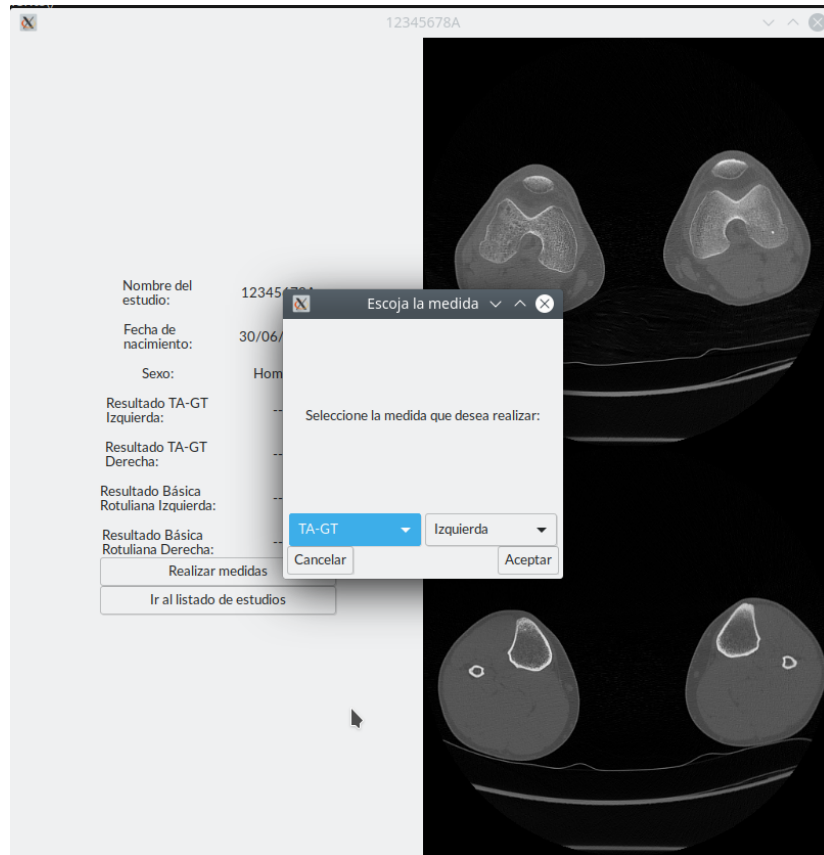


Figura B.5: Ventana de selección de detalles de la medida

### B.3.6 Resultado de la medida

Esta ventana (figura B.6) es mostrada tras haber confirmado la operación de medida en la ventana de selección de medidas. En ella, se muestran los datos personales del paciente en el lateral izquierdo, y en el lateral derecho se muestra la imagen resultante del proceso de cálculo. Debajo de esta imagen se muestra el resultado del cálculo realizado automáticamente.

Debajo de los datos personales del pacientes, en el lateral izquierdo se dispone de dos botones:

- **”Refinar medidas”**: en caso de que el usuario considere que las medidas realizadas no son correctas, y desee corregirlas manualmente, deberá pulsar este botón, que abrirá la ventana de refinar medidas.
- **”Almacenar resultado”**: si el usuario considera que la medida realizada es correcta y desea almacenarla en el sistema, debe pulsar este botón. Tras haber pulsado este botón, se mostrará la ventana de detalles del paciente.

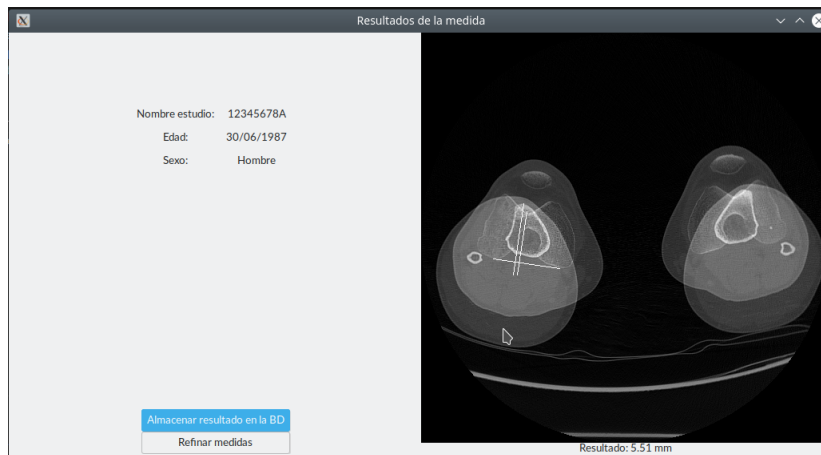


Figura B.6: Ventana de resultado de la medida

### B.3.7 Refinar medidas

Esta ventana (figura B.7) es mostrada tras haber pulsado el botón de "Refinar medidas" en la ventana de resultado de la medida. En esta ventana se muestra la imagen sobre la que se va a realizar la medida de manera manual.

Con el fin de visualizar de manera más nítida las líneas dibujadas, estas líneas se pintan de color azul sobre la imagen.

Para corregir alguna línea, el usuario debe pulsar la línea escogida con el botón izquierdo del ratón. Tras haber pulsado la línea, esta se pintará de color verde (figura B.8), y el extremo que será permitido moverse se pintará de color rojo.

En caso de querer seleccionar el otro extremo de la línea, se debe volver a pulsar sobre el otro extremo de la línea, cancelando la selección anterior. Tras haber pulsado, el extremo pulsado anteriormente se pintará en verde, y el nuevo extremo pulsado se pintará en rojo.

En caso de querer cancelar la selección y no seleccionar ninguna línea, el usuario debe pulsar con el botón izquierdo del ratón sobre algún punto de la imagen lejano con respecto al punto seleccionado.

Para poder mover algún punto, se debe seleccionar un punto siguiendo el procedimiento descrito anteriormente. Una vez seleccionado, se debe situar el cursor del ratón sobre el píxel de la imagen al que se quiere desplazar el punto seleccionado. Una vez situado el cursor sobre el píxel deseado, se debe pulsar el botón derecho del ratón, que desplazará el punto seleccionado al píxel dónde se ubique el cursor. Tras haber pulsado, se verá el punto seleccionado en su nueva ubicación.

En esta ventana se dispone también de dos botones en la región inferior:

- **"Cancelar"**: si el usuario pulsa este botón, cancelará la operación y se le devolverá a la pantalla de resultado de la medida.

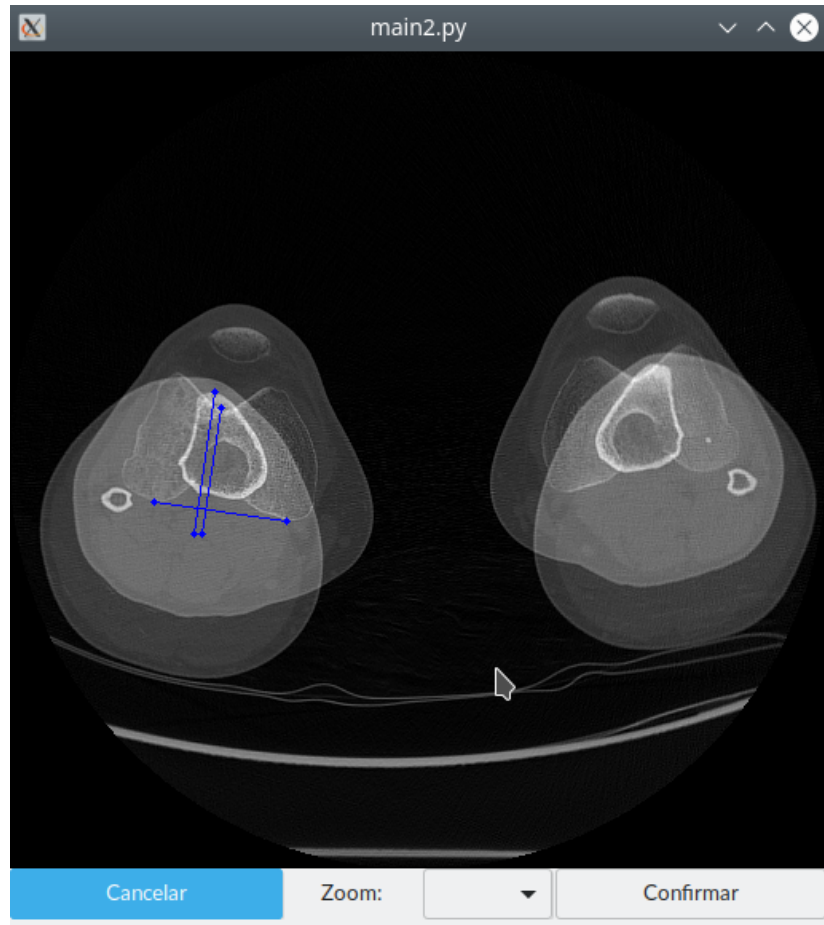


Figura B.7: Ventana de refinado de la medida

- **"Aceptar"**: cuando el usuario haya terminado el proceso de refinamiento de las medidas y quiera calcular el nuevo resultado, pulsará este botón, que le devolverá a la ventana de resultado de la medida. En esta ventana se actualizará la imagen mostrada con los cambios realizados, y se mostrará el nuevo resultado obtenido.

Además, entre estos dos botones, la ventana cuenta con un menú desplegable para seleccionar la escala de tamaño de imagen deseada, en caso de que el usuario desee visualizar la imagen en unas dimensiones diferentes a las mostradas por defecto.

### B.3.8 Listado de estudios

Esta ventana (figura B.9) mostrará un listado de todos los estudios almacenados anteriormente por el usuario. En ella, se muestra una lista con cinco columnas, donde en cada fila se mostrarán los campos nombre del estudio, fecha de nacimiento y sexo. Para poder realizar alguna operación sobre alguno de los estudios listados, se debe seleccionar el estudio en cues-

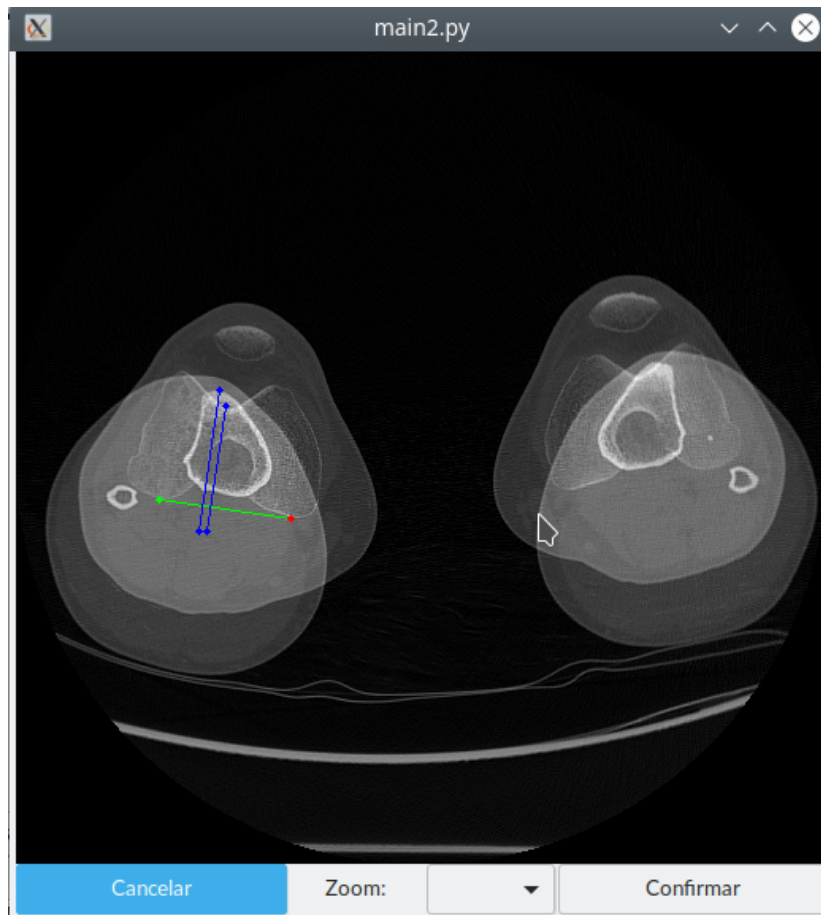


Figura B.8: Ventana de refinado de la medida con una línea seleccionada

ción. La selección se realiza pulsando sobre la fila correspondiente usando el botón izquierdo del ratón. Una vez seleccionado el estudio, se coloreará la fila seleccionada (figura B.10) (el color utilizado dependerá de las opciones de tema del sistema operativo utilizado).

Una vez seleccionado el estudio sobre el que se desea realizar una operación, se pueden pulsar los siguientes botones, ubicados en el extremo superior de la ventana, para realizar las siguientes operaciones:

- **"Detalles del estudio"**: pulsando este botón, se abrirá la ventana de detalles del estudio mostrando los detalles del estudio seleccionado.
- **"Eliminar estudio"**: pulsando este botón es posible eliminar el estudio seleccionado. Para evitar errores, tras pulsar el botón se abrirá una ventana (figura B.11) que obligue al usuario a confirmar la acción.

En caso de querer añadir un nuevo estudio, se dispone del botón "Añadir estudio" en la esquina superior izquierda, que lleva al usuario a la ventana de añadir nuevo estudio.

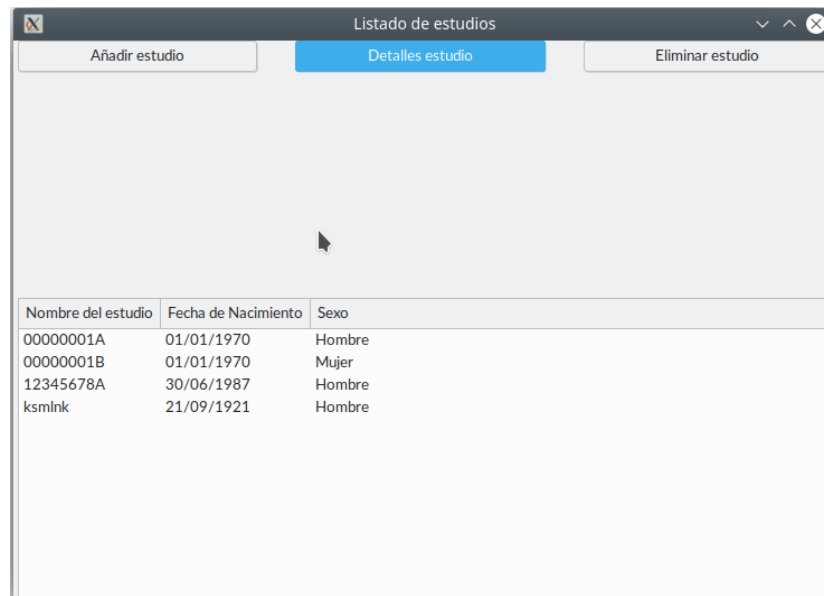


Figura B.9: Ventana de listado de estudios

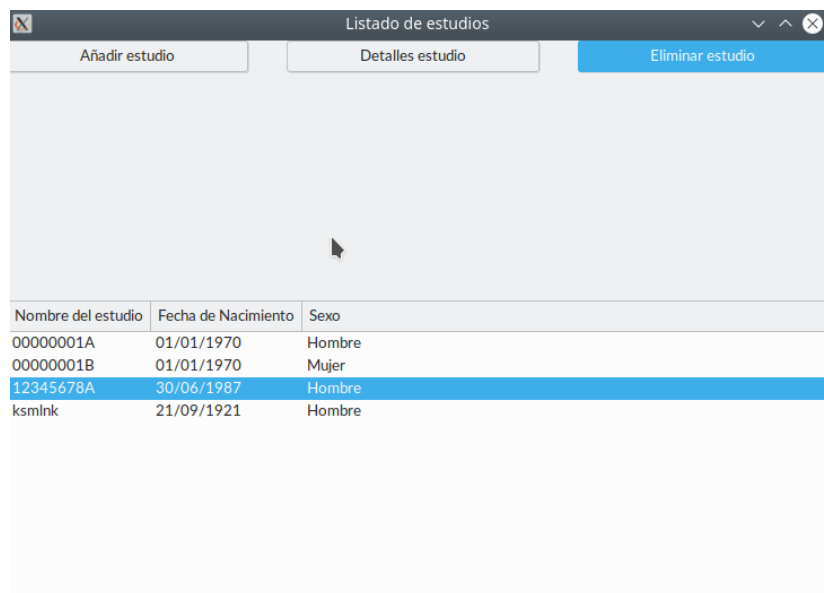


Figura B.10: Ventana de listado de estudios tras haber seleccionado un elemento de la lista





Figura B.11: Ventana de confirmación de eliminación de un estudio





# Glosario

---

**CUDA** Software de computación paralela destinado a utilizar la GPU con propósitos generales desarrollado por la compañía NVIDIA.. 7

**ndarray** Contenedor universal de elementos del mismo tipo y tamaño.. 7

**OpenCL** Alternativa código libre a CUDA.. 7

**PACS** Sistema encargado del almacenamiento digital, de la transmisión y de la descarga de imágenes radiológicas.. 15

**Umbralización** método de segmentación consistente en la separación de los objetos de interés del resto de la imagen. . 31



# Bibliografía

---

- [1] D. Standard, “About dicom: Overview.” [En línea]. Disponible en: <https://www.dicomstandard.org/about-home/>
- [2] DamienD, “A very basic dicom introduction,” 2015. [En línea]. Disponible en: [https://dcm4che.atlassian.net/wiki/spaces/d2/pages/1835038/A+Very+Basic+DICOM+Introduction#:~:text=DICOM%20files%20contain%20a%20file,use%20it%20for%20proprietary%20data\).](https://dcm4che.atlassian.net/wiki/spaces/d2/pages/1835038/A+Very+Basic+DICOM+Introduction#:~:text=DICOM%20files%20contain%20a%20file,use%20it%20for%20proprietary%20data).)
- [3] D. Library, “Dicom tags.” [En línea]. Disponible en: <https://www.dicomlibrary.com/dicom/dicom-tags/>
- [4] Wikipedia, “Hounsfield scale.” [En línea]. Disponible en: [https://en.wikipedia.org/wiki/Hounsfield\\_scale](https://en.wikipedia.org/wiki/Hounsfield_scale)
- [5] JSON, “Introducción a json.” [En línea]. Disponible en: <https://www.json.org/json-es.html>
- [6] P. S. Foundation, “What is python? executive summary.” [En línea]. Disponible en: <https://www.python.org/doc/essays/blurb/>
- [7] T. S. community, “What is numpy?” [En línea]. Disponible en: <https://numpy.org/doc/stable/user/whatisnumpy.html>
- [8] O. team, “About.” [En línea]. Disponible en: <https://opencv.org/about/>
- [9] Wikipedia, “Gtk.” [En línea]. Disponible en: <https://es.wikipedia.org/wiki/GTK>
- [10] T. G. Project, “Gtk.” [En línea]. Disponible en: <https://www.gtk.org/docs/language-bindings/index>
- [11] RadiologyInfo, “Tac.” [En línea]. Disponible en: <https://www.radiologyinfo.org/sp/info.cfm?pg=bodyct>

- [12] wikipedia, “Cóndilo.” [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Condilo>
- [13] fisioterapia online, “troclea.” [En línea]. Disponible en: <https://www.fisioterapia-online.com/glosario/troclea-humeral-y-femoral>
- [14] Wikipedia, “Tat.” [En línea]. Disponible en: [https://en.wikipedia.org/wiki/Tuberosity\\_of\\_the\\_tibia](https://en.wikipedia.org/wiki/Tuberosity_of_the_tibia)
- [15] —, “Patellar translation or tt tg distance.” [En línea]. Disponible en: <https://boneandspine.com/patellar-translation/>
- [16] F. López Ortiz, R. Valverde Cámara, “Valoración radiológica de la inestabilidad rotuliana,” p. 97.
- [17] H. C. M. Ibiza, “Radiología hcm - hospital can misses - ibiza,” 2013. [En línea]. Disponible en: <https://radiologiahcm.wordpress.com/2013/03/06/mediciones-basicas-de-rodilla-en-tac/>
- [18] Medixant, “Radiant dicom viewer.” [En línea]. Disponible en: <https://www.radiantviewer.com/es/>
- [19] PostDICOM, “Top 25 free dicom viewers for doctors, medical students, and health professionals.” [En línea]. Disponible en: <https://www.radiantviewer.com/es/>
- [20] MedDream, “Meddream html5 viewer.” [En línea]. Disponible en: <https://www.softneta.com/products/meddream-dicom-viewer/#downloads>
- [21] D. Rivera, “Cómo implementar test-driven development (tdd) sin morir en el intento,” 2019. [En línea]. Disponible en: <https://medium.com/nursoft/implementar-tdd-facilmente-4d2cfaa9172>
- [22] Github, “Pydicom.” [En línea]. Disponible en: <https://pydicom.github.io/>
- [23] S. Pearls, “Hounsfield unit.” [En línea]. Disponible en: <https://www.ncbi.nlm.nih.gov/books/NBK547721/>
- [24] T. A. Learner, “Tag archives: Angle of rotation by cv2.minareact().” [En línea]. Disponible en: <https://thealearner.com/tag/angle-of-rotation-by-cv2-minareact/>
- [25] D. S. Browser, “Patient module.” [En línea]. Disponible en: <https://dicom.innolitics.com/ciods/rt-plan/patient>