



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
CENTRO DE INFORMÁTICA MÉDICA(CESIM), FACULTAD DE TECNOLOGÍAS INTERACTIVAS

FUNCIONALIDAD PARA EL MÓDULO DE ORTOPEDIA EN EL SISTEMA XAVIA PACSVIEWER.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Javier Alejandro Romero Petrova

Tutores: M.Sc. Arianne Méndez Mederos, Profesor Asistente

Ing. Ianko Zamora Abrahante

La Habana, 2024

"Lo que con mucho trabajo se obtiene, más se ama."

Aristóteles.

Dedicatoria

De Mi:

Para toda mi familia y personas a las que quiero.

Agradecimientos

Me gustaría primero que nada agradecer a toda mi familia en especial a mis padres y a mi pareja en primer lugar por todo su amor y su incondicional apoyo durante todo este tiempo. A mis seres queridos que ya no están pero que fueron mi motivación desde el minuto uno y mis fuerzas para seguir adelante, por eso les debo la vida. Siempre siento que estaré en deuda con ustedes. De igual forma tengo que estar agradecido con la familia y la pareja que tengo, ya que en los momentos más duros y días malos siempre me animan y me ayudan a levantarme. No bastan las palabras para expresar lo mucho que los quiero. Mi vida entera. Los amo mucho: mamá, papá, Amanda y Yulita.

También quiero dar las gracias a todas mis amistades y hermanos que he hecho aquí y que estuvieron siempre para mí, son unas personas excepcionales, con un gran corazón, y yo nunca los voy a olvidar, siempre los voy a llevar aquí... Me estuvieron apoyando todo el tiempo en los momentos difíciles y buenos. En gran parte de no haber sido por ustedes no hubiese conseguido llegar hasta donde estoy ahora. Les deseo lo mejor en su nueva vida como profesionales. Gracias por todo: Javi, Kevin, Orli, Adrian, Kleyris, el Fumeee, el Migue, El rojo, Jose, y muchos más!!!!

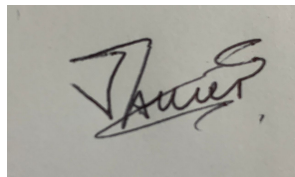
Igualmente me encantaría agradecer y destacar el trabajo y aporte que me brindaron mi tutora y co-tutor, que siempre me atendieron y aclararon todas mis dudas y sin ellos la realización de esta tesis no hubiese sido posible. Muchas Gracias Arianne e Janko!!!

Por último pero no menos importante quiero reconocer y agradecer a muchos profesores por todo el trabajo y el sacrificio que han hecho y han demostrado, pero en particular quiero felicitar a 3 profesores. Una de ellos es la profesora Adriana que ya no se encuentra en la escuela que me impartió la asignatura de álgebra, es de esas pocas personas que te enseñan tan bien y terminas cogiéndole mucho cariño a la asignatura. El otro profe que quiero felicitar inmensamente es al profe Alejandro Mendaro el profe de calculo 2 y 3 que tampoco está en la escuela debido a que ya culimo su labor del servicio social pero me tengo que rendir a él de verdad porque era tan buen profesor que sin dudas calculo era mi asignatura favorita, jamás había estudiado una asignatura con tanto entusiasmo, por eso Ale te estaré eternamente agradecido. Y la ultima persona que quiero de verdad felicitar y hacerle un monumento es a mi jefa de año!!! la profe Saylin!!! la mejor de todas, siempre sonriente , ayudandome siempre en lo que estaba a su alcance y en lo que no también. La profe que siempre estuvo apoyandome! le tengo mucho aprecio y cariño! jamás me olvidaré de ella.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los 10 días del mes de diciembre del año 2024.



Javier Alejandro Romero Petrova

Autor



Digitally signed
by Arianne
Méndez
Mederos

M.Sc. Arianne Méndez Mederos, Profesor Asistente
Tutora

Ianko
Abrahante
Zamora

Digitally signed
by Ianko
Abrahante
Zamora
Date:
2024.12.09
18:13:15 -0500

Ing. Ianko Zamora Abrahante
Co-Tutor

XAVIA PACSViewer es un sistema desarrollado por el Centro de Informática Médica (CESIM) para la visualización y procesamiento de imágenes médicas digitales. Además, cumple con los estándares internacionales DICOM para la gestión de imágenes médicas y sus informes de diagnóstico. En ocasiones, cuando el especialista realiza mediciones, estas se salen del plano de visualización de la imagen, lo que dificulta el cálculo de ángulos con las herramientas actuales del sistema. En ausencia de estas herramientas, el especialista debe recurrir a mediciones manuales en placas impresas, lo que puede afectar la precisión del diagnóstico. Ahora se va a desarrollar una nueva funcionalidad: el módulo de Ortopedia que se integrará al sistema XAVIA PACSViewer. Dentro de la Ortopedia, que es una rama de la medicina, se diagnostican varias enfermedades como la escoliosis, que se diagnostica a través del método de Cobb, y también existe una condición biomecánica llamada Balance Sagital, que se diagnostica a través de los parámetros espino pélvicos. El objetivo de la investigación es desarrollar una funcionalidad para el cálculo de la amplitud angular para el diagnóstico de la escoliosis y el balance sagital en el sistema XAVIA PACSViewer que permita medir correctamente y de manera cómoda, mejorando así la precisión diagnóstica. Para ello, se emplearon métodos teóricos, métodos empíricos y técnicas de investigación. Las herramientas utilizadas en el desarrollo incluyen el lenguaje C#, .NET Framework, el entorno de desarrollo integrado Visual Studio 2022 y Visual Paradigm como herramienta CASE. El aporte práctico de esta aplicación es la capacidad de realizar mediciones angulares precisas y eficientes, lo que asegura diagnósticos más exactos y confiables, beneficiando tanto a los especialistas como a los pacientes en el ámbito ortopédico.

Palabras clave: Ortopedia, método de Cobb, diagnóstico, parámetros espino pélvicos, XAVIA PACSViewer.

XAVIA PACSViewer is a system developed by the Center for Medical Informatics (CESIM) for the visualization and processing of digital medical images. In addition, it complies with international DICOM standards for the management of medical images and their diagnostic reports. Sometimes, when the specialist takes measurements, these go out of the image visualization plane, making it difficult to calculate angles with the system's current tools. In the absence of these tools, the specialist must resort to manual measurements on printed plates, which can affect the accuracy of the diagnosis. A new functionality is now being developed: the Orthopedics module that will be integrated into the XAVIA PACSViewer system. Within Orthopedics, which is a branch of medicine, several diseases are diagnosed, such as scoliosis, which is diagnosed using the Cobb method, and there is also a biomechanical condition called Sagittal Balance, which is diagnosed through spinopelvic parameters. The objective of the research is to develop a functionality for calculating the angular amplitude for the diagnosis of scoliosis and sagittal balance in the XAVIA PACSViewer system that allows correct and comfortable measurement, thus improving diagnostic accuracy. Theoretical methods, empirical methods, and research techniques were used for this. The tools used in the development include the C# language, .NET Framework, the integrated development environment Visual Studio 2022, and Visual Paradigm as a CASE tool. The practical contribution of this application is the ability to perform precise and efficient angular measurements, ensuring more accurate and reliable diagnoses, benefiting both specialists and patients in the orthopedic field.

Keywords: *Orthopedics, Cobb method, diagnosis, spinopelvic parameters, XAVIA PACSViewer.*

Introducción	1
1 FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	5
1.1 Diagnóstico y Evaluación de las enfermedades de la columna vertebral	5
1.2 Diagnóstico y Evaluación del Balance Sagital	9
1.3 Sistemas homólogos.	11
1.4 Tecnologías utilizadas para el desarrollo	14
2 PROPUESTA DE SOLUCIÓN	18
2.1 Modelado de los procesos del negocio	19
2.2 Requisitos del sistema	20
2.3 Arquitectura de Software	34
2.4 Modelo de Datos	35
2.5 Diagramas de secuencia	37
2.6 Patrones de diseño	40
3 IMPLEMENTACIÓN Y PRUEBAS	42
3.1 Estándares de codificación	42

3.2	Tratamiento de excepciones	44
3.3	Pruebas de Software	45
3.4	Tipo de Prueba Funcional	45
3.5	Pruebas No Funcionales	56
3.6	Listas de verificación	57
3.6.1	Listas de verificación basadas en Usabilidad	58
3.6.2	Listas de verificación basadas en Portabilidad	60
	Conclusiones	63
	Recomendaciones	64
	Referencias bibliográficas	68
	Apéndices	71
.1	RF 5. Cargar Imagen	72
.2	RF 6. Invertir Perpendicular de PI	72
.3	RF 7. Eliminar trazo	73
.4	Diagrama del proceso de negocio sin sistema	74

Índice de figuras

1	Ángulo de Cobb	7
2	Curvaturas anormales de la columna vertebral en el plano sagital.	8
3	Método de Ferguson. Fuente: (García Fontecha, 2014).	8
4	Método de Incidencia Pélvica.	10
5	Método de Versión Pélvica.	10
6	Método de Inclinación de Sacro.	11
7	Diagrama del proceso de negocio	19
8	Requisitos No Funcionales.	27
9	Patrón Arquitectónico: Modelo-Vista-VistaModelo	35
10	Diagrama de clases.	37
11	Diagrama de Secuencia Cobb.	38
12	Diagrama de Secuencia Incidencia Pélvica.	38
13	Diagrama de Secuencia Versión Pélvica.	39
14	Diagrama de Secuencia Inclinación de Sacro.	40
15	Set de pruebas realizadas al control de usuario que pertenece al parámetro espino pélvico de Incidencia Pélvica.	47

16	Set de pruebas realizadas al control de usuario que pertenece al método de Cobb.	48
17	Set de pruebas realizadas al control de usuario que pertenece al parámetro espino pélvico de Inclinación de Sacro.	49
18	Set de pruebas realizadas al control de usuario que pertenece al parámetro espino pélvico de Versión Pélvica.	50
19	Set de pruebas restantes realizadas al control de usuario que pertenece al parámetro espino pélvico de Versión Pélvica.	51
20	Resultados de pruebas basadas en tiempo de CPU	57
21	Diagrama del proceso de negocio sin sistema	74

Índice de tablas

1	Análisis de los sistemas de tratamiento de imágenes médicas.	13
2	Descripción del RF 1 Medir Ángulo de Cobb.	20
3	Descripción del RF2 Medir Ángulo de Incidencia Pélvica (PI).	22
4	Descripción del RF3 Medir Ángulo de Versión Pélvica (PT).	23
5	Descripción del RF4 Medir Ángulo de Inclinación de Sacro (SS).	25
6	Descripción del RNF 1 de usabilidad	28
7	Descripción del RNF 2 de usabilidad	29
8	Descripción del RNF 3 de usabilidad	30
9	Descripción del RNF 4 de usabilidad	31
10	Descripción del RNF de eficiencia de desempeño	31
11	Descripción del RNF 1 de mantenibilidad	33
12	Descripción del RNF 2 de mantenibilidad	34
13	Descripción del RNF de portabilidad	34
15	Resultados de la prueba de caja negra.	53
14	Listado de no conformidades detectadas por iteración	54
16	Características del entorno de prueba.	57

17	Lista de Chequeo de Usabilidad	58
18	Lista de Chequeo - Portabilidad	60
19	Descripción del RF 5 Cargar Imagen.	72
20	Descripción del RF 6 Invertir Perpendicular de PI.	72
21	Descripción del RF 7 Eliminar trazo	73

En la era moderna, la intersección entre la medicina y las ciencias informáticas ha promovido avances significativos en la precisión y eficacia del diagnóstico de diversas condiciones médicas. La ortopedia, como rama de la medicina, evalúa y trata lesiones y enfermedades en diferentes regiones del cuerpo, incluyendo las extremidades y la columna vertebral (J. e. a. Smith, 2020). Entre las afecciones más comunes se encuentra la escoliosis, una deformidad lateral de la columna vertebral que se diagnostica mediante radiografías y la medición de la amplitud angular por el método de Cobb y de Ferguson (J. Smith, 2023). Además, en la ortopedia se evalúa una condición biomecánica llamada balance sagital, que es crucial para mantener una postura erguida y se refiere a la alineación de la columna vertebral con respecto a la pelvis. Este balance se diagnostica a través de los parámetros espino-pélvicos, que incluyen la incidencia pélvica (PI), la versión pélvica (PT) y la inclinación del sacro (SS) (Miller, 2018). La evaluación de la escoliosis no solo se limita a la medición del ángulo de Cobb, sino que también implica considerar los parámetros espino-pélvicos, que son fundamentales para comprender la alineación y la mecánica de la columna vertebral en relación con la pelvis, lo cual es esencial para un diagnóstico preciso y un plan de tratamiento adecuado (Davis, 2019). La inclusión de estos parámetros en la evaluación de la escoliosis permite una comprensión más completa de la condición y puede mejorar la precisión de las mediciones, así como la planificación del tratamiento (Miller, 2018). Al abordar estos tres parámetros junto con el método de Cobb, los profesionales de la salud pueden obtener una visión integral de la deformidad espinal y su impacto en la biomecánica del cuerpo (T. e. a. Williams, 2021).

Cuando el radiólogo especialista en ortopedia evalúa una imagen radiográfica digital en el visor XAVIA PACSViewer tiene que utilizar la herramienta de medición angular. Esta herramienta traza el ángulo a partir de 3 puntos en el que el punto intermedio va a constituir el vértice del ángulo. El especialista debe marcar en el plano de la imagen estos 3 puntos. Sin embargo, cuando se evalúa la escoliosis en una radiografía, para el cálculo de la amplitud angular de la curvatura, el trazo sale del plano. Para medirlo es necesario comenzar trazando rectas tangentes, por lo que no es posible utilizar desde un inicio la medición angular. En su lugar, se debe utilizar la herramienta de línea para realizar estos trazos. Una vez que las líneas se cruzan, es necesario utilizar la herramienta de medición de la amplitud angular marcando los puntos exactamente sobre

las líneas y el vértice en el punto exacto del cruce. Cuando la curvatura de la columna es muy pronunciada, es posible que el punto donde las líneas tangentes se cortan se mantenga en el plano de la imagen, pero si esto no sucede, el cruce de las líneas queda fuera del plano y no se visualiza en la imagen.

Cuando esto sucede, el especialista tiene dos alternativas. Una de ellas, y la más común, es utilizar otro visor de imágenes médicas que implemente el cálculo de esta amplitud angular por el método de Cobb. Sin embargo, esta opción puede resultar en la pérdida de clientes para el sistema XAVIA PACS, ya que otros visores pueden no tener soporte técnico ni certificación oficial como equipo médico para el diagnóstico. Además, el proceso de diagnóstico puede tener demoras, y el especialista puede optar por no registrar cambios o no utilizar el sistema XAVIA debido al agotamiento causado por cambiar constantemente de sistema. La otra alternativa es aplicar el método de Cobb utilizando la herramienta de trazar líneas. En este caso, el especialista debe trazar perpendiculares a cada tangente y alargar el trazo hasta que se corten las dos perpendiculares, aplicar las propiedades de la trigonometría para localizar el ángulo, y luego utilizar la herramienta del ángulo para marcar los 3 puntos sobre las líneas, asegurándose de que el segundo punto esté exactamente en el cruce. Este proceso implica más trazos y requiere una estimación visual precisa, lo que puede generar fatiga y agotamiento visual. Además, cualquier error en la aplicación de las propiedades trigonométricas o en la selección de los puntos puede llevar a un diagnóstico erróneo. Si se realiza un trazo incorrecto o se selecciona un punto incorrecto, se debe comenzar el proceso nuevamente, lo que elimina el progreso alcanzado. Esta situación afecta directamente la precisión del diagnóstico, pudiendo llevar a errores en las mediciones y, en consecuencia, a diagnósticos erróneos o imprecisos.

La introducción de herramientas informáticas ha mejorado significativamente la precisión y reproducibilidad de las mediciones, reduciendo los errores asociados con las mediciones manuales (Martínez, 2022). Esta evolución tecnológica ha permitido una evaluación más precisa de la escoliosis, contribuyendo a decisiones clínicas más informadas y personalizadas (Lopez, 2023).

La inclusión de los parámetros espino-pélvicos en la evaluación de la escoliosis permite una comprensión más completa de la condición y puede mejorar la precisión de las mediciones, así como la planificación del tratamiento (Miller, 2018). Al abordar estos tres parámetros junto con el método de Cobb, los profesionales de la salud pueden obtener una visión integral de la deformidad espinal y su impacto en la biomecánica del cuerpo (T. e. a. Williams, 2021).

Dada la situación actual, se plantea el siguiente **problema científico**: ¿Cómo facilitar la medición angular para el diagnóstico de la escoliosis y el balance sagital?

El **objeto de estudio** se enmarca en el diagnóstico de la escoliosis y el balance sagital. El **campo de acción** se centra en la medición de la amplitud angular para el diagnóstico de la escoliosis y el balance sagital en el sistema XAVIA PACSViewer.

Para dar solución al problema planteado se define como **objetivo general**: Desarrollar las funcionalidades para calcular la amplitud angular para el diagnóstico de la escoliosis y el balance sagital en el sistema XAVIA PACSViewer.

El cumplimiento del objetivo general se desglosa en las siguientes **tareas de investigación**:

- Elaboración de los fundamentos teóricos y metodológicos de la investigación asociados al diagnóstico de la escoliosis y el balance sagital.
- Análisis de los métodos para el cálculo de la amplitud angular para el diagnóstico de la escoliosis y el balance sagital.
- Desarrollo de las funcionalidades para el cálculo de la amplitud angular de los métodos para el sistema XAVIA PACSViewer.
- Validación del resultado obtenido a partir de las observaciones del especialista, así como su aceptación y uso por parte del mismo, las pruebas propuestas para la investigación y las validaciones de la herramienta con datos reales de pacientes para asegurar su fiabilidad y precisión, comparando los resultados obtenidos con los métodos tradicionales manuales.

Para el desarrollo de la investigación se emplearon los siguientes métodos y técnicas:

Histórico?lógico: Permite trazar la evolución histórica de un fenómeno y comprender su lógica interna, facilitando una aproximación a los referentes teóricos del tema. Empleado para profundizar en el estudio de la escoliosis y la implementación de herramientas informáticas en su diagnóstico y evaluación. Se utilizó para el análisis documental del expediente del proyecto, evaluando su evolución en el tiempo y los métodos de diagnóstico de la escoliosis y el balance sagital.

Inductivo-deductivo: Implica derivar principios generales a partir de observaciones específicas y aplicar principios generales a casos específicos. Este método es crucial para la formación de hipótesis y la verificación de marcos teóricos dentro del estudio, mostrando cómo se necesita desarrollar la nueva funcionalidad.

Analítico-sintético: Se estudiaron las formas de calcular los métodos de diagnóstico de la escoliosis y el balance sagital mediante análisis y síntesis de la información para llegar a un resultado.

Modelación: Se modelaron diagramas de clases, de secuencia y la descripción del proceso de negocio para estructurar y entender mejor las funcionalidades del sistema.

Análisis documental: Se centró en el estudio de documentos y textos para extraer las relaciones esenciales del objeto investigado. Permite concretar la bibliografía relevante, determinando el sistema de conceptos y categorías clave para el estudio, incluyendo la clasificación de enfermedades de la columna vertebral y el

balance sagital.

Entrevista: Mediante entrevistas con especialistas en ortopedia y el equipo del sistema XAVIA PACS-Viewer, se obtuvo información crucial acerca de las necesidades del sistema y las mejoras necesarias para el diagnóstico de la escoliosis y el balance sagital.

Estas metodologías y técnicas se emplearon para asegurar la precisión y efectividad del sistema, basándose en fundamentos científicos sólidos y en una comprensión integral de la escoliosis y sus implicaciones clínicas.

El documento se estructura en tres capítulos que abordan de manera integral el desarrollo del submódulo de las Unidades de Cuidados Intensivos:

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Se realiza el marco conceptual necesario para comprender los aspectos fundamentales relacionados con el diagnóstico de la escoliosis y el balance sagital. Se analizan los principales conceptos del dominio, así como el proceso de diagnóstico en esta área. Se realiza un estado del arte que caracteriza diversos sistemas informáticos utilizados para gestionar la información relacionada con el tratamiento de imágenes médicas. Además, se mencionan las metodologías de desarrollo, herramientas, tecnologías y lenguajes de programación que se implementarán para la propuesta de solución, asegurando así una base sólida para el desarrollo de la funcionalidad de mediciones específicas en el sistema XAVIA PACSViewer.

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO ORTOPEDIA

Se realiza la propuesta de solución para el desarrollo de un módulo específico de ortopedia en el sistema XAVIA PACSViewer. Primero, se modelarán los procesos de negocio mediante diagramas BPMN, seguidos de la definición de los requisitos funcionales, y se asumirán los requisitos no funcionales del visor XAVIA PACSViewer. Luego, se definirá el diagrama de clases y se desarrollarán diagramas de secuencia para cada método principal. También se establecerá la arquitectura del sistema basada en el patrón Modelo-Vista-VistaModelo (MVVM), y finalmente se describirán los patrones de diseño utilizados.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS

Se describen los estándares de codificación utilizados para obtener un código limpio y legible. Así como se hace referencia a las tareas ingenieriles que rigen el desarrollo de la herramienta y las pruebas realizadas a la misma. Con los resultados obtenidos se demuestra la eficiencia y el correcto funcionamiento de la solución que se desarrolló para dar solución al problema planteado. Se describen las pruebas funcionales y no funcionales a las que se somete el software.

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

En el presente capítulo se abordan los referentes teóricos y metodológicos en el diagnóstico de las enfermedades de la columna vertebral. Se realiza una caracterización de la escoliosis y el balance sagital abordando diagnóstico, afecciones que presentan quienes lo padecen, cómo identificarlos y diagnosticarlo. Además se caracterizan y comparan sistemas similares con el objetivo de extraer características y evaluar la posible reutilización del código. Por último se realiza una caracterización de las tecnologías utilizadas para el desarrollo de la solución.

1.1. Diagnóstico y Evaluación de las enfermedades de la columna vertebral

Las enfermedades y padecimientos de la columna vertebral son afecciones comunes que afectan a una gran parte de la población en algún momento de sus vidas. Entre las más frecuentes se encuentran la escoliosis, la lordosis, la cifosis y las hernias discales. Estas condiciones pueden causar dolor, deformidades y limitaciones en la movilidad, impactando significativamente la calidad de vida de las personas afectadas. Las causas de estos trastornos son variadas, incluyendo factores genéticos, traumas, enfermedades degenerativas y malas posturas mantenidas durante largos periodos.

Escoliosis

La escoliosis es una desviación lateral de la columna vertebral que se manifiesta principalmente en el plano frontal y afecta a una amplia gama de pacientes, incluyendo niños, adolescentes y adultos (Martínez, 2020). A pesar de los esfuerzos realizados desde su descubrimiento, el tratamiento y diagnóstico de esta condición siguen presentando desafíos significativos (Smith, 2018).

Su diagnóstico incluye una combinación de evaluaciones clínicas y pruebas de imagen. Estos estudios de imágenes se realiza con radiografías simples de la columna para confirmar el diagnóstico y evaluar la gravedad de la curvatura. En algunos casos, se pueden recomendar pruebas adicionales como resonancias magnéticas o tomografías computarizadas para descartar causas subyacentes (A. e. a. Smith, 2021).

Para medir la curvatura de la columna en la escoliosis, se emplean varios métodos, entre ellos el método de Ferguson y el método de Cobb . El método de Cobb consiste en identificar las vértebras superior e inferior de la curva, dibujar líneas paralelas a la parte superior de la vértebra superior y la parte inferior de la vértebra inferior, y luego trazar perpendiculares a estas líneas. El ángulo donde se cruzan las perpendiculares representa la gravedad de la curvatura (Jones, 2022).

El método de Ferguson, aunque es menos común, se utiliza en casos específicos y consiste en medir la curvatura a lo largo de la columna utilizando diferentes puntos de referencia y técnicas de medición específicas (Brown, 2023). Para calcular la curvatura mediante el este método, primero se identifican los vértices de la curvatura, que son las vértebras que marcan el inicio y el final de la curva. Luego, se marcan tres puntos específicos en las radiografías: uno en el centro del cuerpo vertebral de la vértebra superior al vértice de la curvatura, otro en el centro del cuerpo vertebral de la vértebra inferior al vértice de la curvatura, y el tercero en el cuerpo vertebral de la vértebra que está en el ápice de la curvatura. A continuación, se trazan líneas desde el punto superior al vértice y desde el vértice al punto inferior. El ángulo de la escoliosis se mide en el punto donde se cruzan las líneas trazadas, obteniendo así la amplitud angular de la curvatura.

Además de estos métodos, existen otros menos utilizados como el método de Risser y el método de Lenke, que también evalúan la curvatura de la columna en la escoliosis (R. e. a. Williams, 2019). Estos métodos proporcionan una evaluación precisa de la escoliosis, contribuyendo a diagnósticos más exactos y tratamientos más adecuados.

Método de Cobb

El método del ángulo de Cobb fue descubierto por el cirujano John Robert Cobb, quien lo describió por primera vez en el año 1948. Es el más utilizado y efectivo para validar la escoliosis, permite cuantificar la magnitud de la deformación y la gravedad de la columna vertebral, al obtener los resultados se podría evitar una cirugía innecesaria y elegir el mejor tratamiento a seguir (Lisintuna, 2015).

Para medir el ángulo de Cobb, primero se identifican las vértebras más inclinadas en la parte superior e inferior de la curva escoliótica. La vértebra superior es aquella cuya placa superior está inclinada más hacia la curva, mientras que la vértebra inferior es aquella cuya placa inferior está inclinada más hacia la curva. Una vez identificadas estas vértebras, se trazan líneas a lo largo de sus placas superiores e inferiores. Después, se dibujan líneas perpendiculares a estas primeras líneas. El ángulo formado en la intersección de estas líneas perpendiculares es el ángulo de Cobb, que representa la severidad de la curvatura (A. e. a. Smith,

2021) (Figura 1). Este método proporciona una medida objetiva de la deformidad espinal, permitiendo a los especialistas realizar un diagnóstico preciso y planificar el tratamiento adecuado (Brown; 2019).

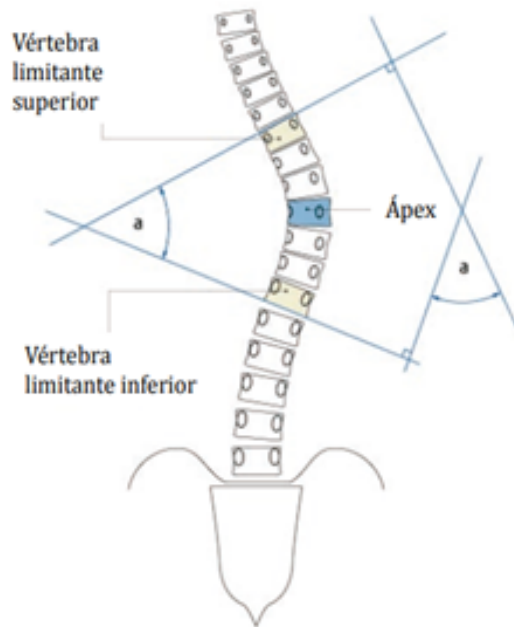


Figura 1. Ángulo de Cobb

La precisión del método Cobb depende de factores como la identificación de las vértebras límite, la variabilidad inter e intraobservador, la influencia de la rotación vertebral y el tipo de instrumento utilizado para medir el ángulo. Algunos datos relevantes sobre su precisión incluyen que el error en la medición del ángulo de forma manual suele estar entre 2° y 7° de curvatura de la columna (Rong, 2019). La medición del ángulo de forma digital ofrece la misma fiabilidad respecto a las mediciones manuales estandarizadas por expertos (Cheung, 2016). Se definen como escolióticas aquellas curvaturas superiores a los 10° (Hresko, 2013). La clasificación de la severidad de la escoliosis en cuanto a su ángulo varía significativamente según los estudios, siendo los más utilizados: leve (10-24°), moderada (24-50°) y grave (>50°) (Negrini, 2012).

Fundamentos matemáticos:

- Paralelismo entre líneas: plantea que dos líneas son paralelas si no se cortan nunca, es decir, si tienen la misma dirección. Se trazan dos líneas paralelas a las superficies de los cuerpos vertebrales donde comienza y termina la deformidad, que son las vértebras más inclinadas de la curva (Cobb, 1948).
- Ángulo entre líneas: consiste en la medida de la abertura que forman dos líneas que se cortan en un punto. Se mide el ángulo que forman las líneas paralelas trazadas anteriormente, denominado ángulo de Cobb (Lonstein, 1994).

- Teorema de Tales: establece que, si dos rectas secantes cortan a otras rectas paralelas, entonces los segmentos determinados en las secantes son proporcionales. Este teorema se utiliza para calcular el ángulo a partir de la medida de los segmentos que forman las líneas paralelas con la línea que une las vértebras límite de la curva (Rigo, 2010).

Método de Ferguson

El método Ferguson se emplea para evaluar la cifosis (curvatura de la columna vertebral que produce un arqueamiento o redondeo de la espalda) y la lordosis (curvatura exagerada y hacia adentro de la columna lumbar), que son las curvaturas anormales de la columna vertebral en el plano sagital (Figura 2). Fue descubierto por el cirujano ortopédico John Ferguson en el año 1930 (Romero Echegaray; Bobadilla Minaya; Castro Rojas; Paredes Campos y Zuñiga Osorio, 2020). Consiste en trazar una línea que une los centros de los cuerpos vertebrales desde el sacro hasta el occipucio, y medir el ángulo que forma esta línea con la horizontal (Figura 3). También se puede aplicar a otras partes del cuerpo, como la rodilla o la cadera, para medir la flexión o la extensión de estas articulaciones.

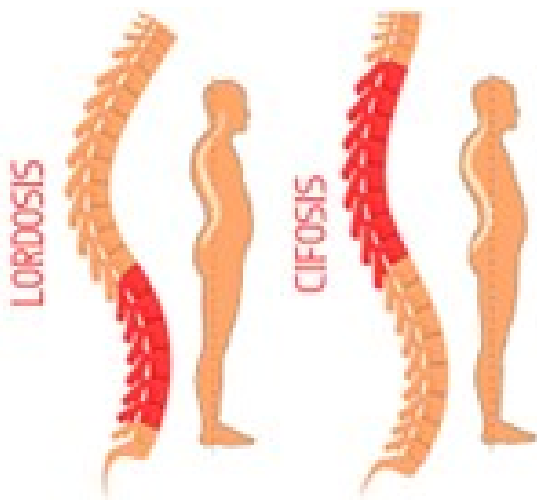


Figura 2. Curvaturas anormales de la columna vertebral en el plano sagital.

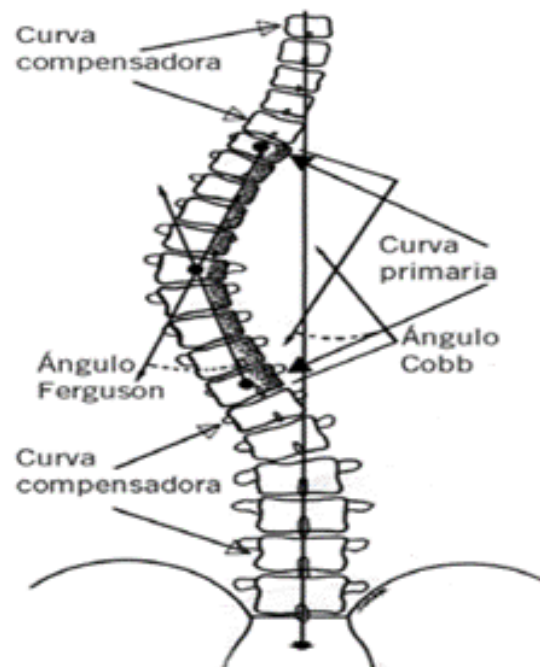


Figura 3. Método de Ferguson. Fuente: (García Fontecha, 2014).

La precisión del método Ferguson depende de varios factores, incluyendo la posición del paciente, la inclinación pélvica, la identificación de los centroides vertebrales y el tipo de instrumento utilizado para

medir el ángulo. Manualmente, el error en la medición del ángulo suele estar entre 3° y 5° de curvatura de la columna. La medición digital ofrece una mayor precisión y reproducibilidad, reduciendo el error a menos de 1°. La inclinación pélvica puede modificar el ángulo de la línea de Ferguson con la horizontal, por lo que se recomienda medir la inclinación pélvica y corregir el ángulo de Ferguson en función de ella.

1.2. Diagnóstico y Evaluación del Balance Sagital

El balance sagital se refiere a la alineación de la columna vertebral en el plano sagital (de adelante hacia atrás) y es crucial para mantener una postura erguida y eficiente. Un buen balance sagital permite una distribución equilibrada de las cargas a lo largo de la columna vertebral, lo que minimiza el esfuerzo muscular y reduce el riesgo de dolor o lesiones (Legaye, 1998). Esta alineación se mantiene gracias a la combinación óptima de las curvaturas de la columna vertebral y la estructura de la pelvis. Los parámetros espino-pélvicos son esenciales para evaluar y mantener este balance porque determinan cómo se distribuye la curvatura de la columna y cómo la pelvis se posiciona en relación con la columna vertebral. Cualquier desequilibrio en estos parámetros puede resultar en una distribución incorrecta de las cargas, llevando a dolor crónico y aumentando el riesgo de lesiones.

Parámetros Espino Pélvicos

Los parámetros espino-pélvicos son medidas utilizadas para evaluar la alineación y el balance de la columna vertebral en relación con la pelvis. Están definidos por: la incidencia pélvica (PI), la versión pélvica (PT) y la inclinación sacra (SS).

Incidencia Pélvica (PI)

La incidencia pélvica es independiente de la posición del cuerpo y se considera un valor anatómico fijo para cada individuo. Varía entre individuos y puede influir en la lordosis lumbar, la cifosis torácica y la lordosis cervical. La incidencia pélvica es crucial porque su valor puede predecir la alineación espinal y la distribución de cargas a lo largo de la columna vertebral (Boulay, 2006).

Método de medición

Este parámetro es el ángulo formado por una línea perpendicular a la plataforma del sacro, que parte de la mitad del sacro, y una segunda línea que va del centro de las cabezas femorales a la mitad de la plataforma del sacro (Duval-Beaupère 1992) como se puede presenciar en la Figura 4.

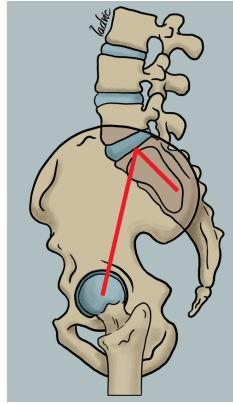


Figura 4. Método de Incidencia Pélvica.

Versión Pélvica (PT)

La versión pélvica, también conocida como inclinación pélvica, refleja la inclinación de la pelvis en el plano sagital y puede variar en respuesta a diferentes posturas y compensaciones biomecánicas. La versión pélvica es crucial porque proporciona información sobre la orientación de la pelvis en relación con el eje vertical del cuerpo, lo que influye en la alineación y biomecánica de la columna vertebral.

Método de medición

Este parámetro es el ángulo formado entre la línea que conecta el centro de las cabezas femorales con el punto medio del platillo sacro y una línea vertical que pasa por el punto medio de las cabezas femorales (Legaye, 1998) como se puede presenciar en la Figura 5.

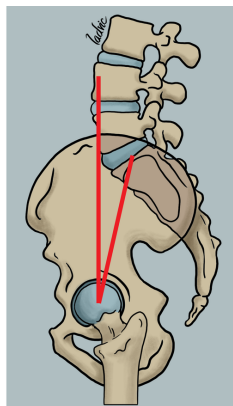


Figura 5. Método de Versión Pélvica.

Inclinación de Sacro (SS)

La inclinación del sacro es un valor variable que puede cambiar con la postura y las compensaciones biomecánicas. Es esencial para comprender cómo la orientación del sacro influye en la alineación espinal y la biomecánica del cuerpo.

Método de medición

Este parámetro es el ángulo formado entre la línea horizontal y la línea tangente al platillo superior de la primera vértebra sacra (S1). Este parámetro refleja la inclinación del sacro en relación con la horizontal y proporciona información crucial sobre la posición del sacro (Boulay, 2006) como se puede presenciar en la Figura 6.

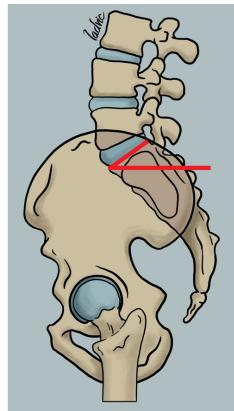


Figura 6. Método de Inclinación de Sacro.

1.3. Sistemas homólogos.

A partir de una revisión bibliográfica se identificaron sistemas para la visualización de imágenes médicas. En esta revisión se tuvieron en cuenta los siguientes criterios:

- medición de Cobb, Ferguson y los parámetros espino-pélvicos.
- lenguaje de desarrollo
- sistema de código abierto
- tipo de licencia
- plataforma sobre la que se ejecuta.

Los sistemas identificados son:

- **AMIDE**: Sistema de código abierto desarrollado por Andy Loening, que permite la visualización y análisis de imágenes DICOM. Este es ampliamente utilizado en el ámbito académico debido a su accesibilidad y características robustas. La herramienta proporciona capacidades básicas y avanzadas para la visualización de imágenes y la realización de medidas cuantitativas, incluidas las mediciones del grado de escoliosis mediante el método de Cobb. Sin embargo, AMIDE no incluye herramientas específicas para medir los parámetros espino-pélvicos, lo que limita su aplicabilidad en estudios ortopédicos complejos. Entre sus ventajas destaca su gratuidad y su interfaz intuitiva que facilita su uso incluso para usuarios con poca experiencia técnica. Como desventaja, se encuentra la falta de herramientas avanzadas que son necesarias para ciertos tipos de análisis clínicos especializados (Loening, 2022).
- **3D Slicer**: es una plataforma de código abierto desarrollada por el equipo de Surgical Planning Laboratory en el Brigham and Women's Hospital. Se utiliza para el análisis y la comprensión de datos de imágenes médicas. Este sistema ofrece una amplia gama de herramientas para el análisis de imágenes, incluyendo la medición del grado de escoliosis mediante el método de Cobb, así como herramientas para medir los tres parámetros espino-pélvicos: inclinación del sacro, versión pélvica e incidencia pélvica. La flexibilidad de 3D Slicer permite su uso en una variedad de aplicaciones médicas, desde la planificación quirúrgica hasta la investigación biomédica. Las ventajas de este sistema incluyen sus capacidades avanzadas de análisis y la posibilidad de personalización a través de su código abierto. Sin embargo, su complejidad puede representar una curva de aprendizaje empinada para usuarios sin experiencia técnica, lo que puede ser una desventaja significativa (3D Slicer2022).
- **Horos**: es un visor de imágenes DICOM para macOS, basado en la plataforma OsiriX. Este software ofrece herramientas para la visualización avanzada de imágenes y la realización de mediciones, incluyendo el método de Cobb para la escoliosis. Aunque Horos es ideal para usuarios de sistemas Apple y permite la integración con Radiopedia para compartir casos clínicos, carece de herramientas para medir los parámetros espino-pélvicos. Una de las principales ventajas de Horos es su compatibilidad con macOS y su interfaz de usuario amigable, lo que facilita su adopción entre profesionales de la salud que utilizan este sistema operativo. No obstante, su falta de funcionalidades avanzadas para la medición espino-pélvica es una limitación importante para su uso en estudios ortopédicos detallados (H. P. Team, 2022).
- **MicroDicom**: es un visor de imágenes médicas gratuito diseñado para la visualización y el análisis de imágenes DICOM. Ofrece herramientas para medir el grado de escoliosis mediante el método de Cobb, pero no incluye herramientas para medir los parámetros espino-pélvicos. MicroDicom es conocido por ser fácil de instalar y usar, con una interfaz sencilla que facilita la visualización básica de imágenes médicas. Entre sus ventajas se encuentra su gratuidad y accesibilidad, lo que lo hace

ideal para usuarios con necesidades básicas de visualización. Sin embargo, la falta de funcionalidades avanzadas limita su aplicabilidad en estudios médicos que requieren análisis detallados de parámetros espino-pélvicos (MicroDicom, 2022).

- **WeAsis:** es un visor DICOM de código abierto y multiplataforma desarrollado por the Weasis Team, que permite la integración flexible con sistemas de información hospitalarios (PACS, RIS, HIS, PHR). Este visor ofrece herramientas avanzadas para medir el grado de escoliosis mediante el método de Cobb y también permite la medición de los tres parámetros espino-pélvicos. WeAsis se destaca por su capacidad de funcionar en diferentes sistemas operativos y su flexibilidad de integración, lo que lo convierte en una opción robusta para instituciones médicas que requieren soluciones interoperables. Sin embargo, su complejidad puede requerir conocimientos técnicos avanzados para su configuración y uso óptimo, lo que puede ser una desventaja para algunos usuarios (W. Team, 2022).

En la Tabla 1 se realiza un resumen comparativo entre los sistemas antes mencionados en cuanto a las características definidas.

Tabla 1. Análisis de los sistemas de tratamiento de imágenes médicas.

Sistema	Implementa el Método de Cobb	Implementa el Método de Ferguson	Implementa Inclinação de Sacro	Implementa Versión Pélvica	Implementa Incidencia Pélvica
AMIDE	SI	NO	NO	NO	NO
3D Slicer	SI	NO	SI	SI	SI
Horos	SI	NO	NO	NO	NO
WeAsis	SI	NO	SI	SI	SI
MicroDicom	SI	NO	NO	NO	NO

El resultado del análisis arrojó que:

- El método de Ferguson para calcular el grado de la escoliosis no es implementado en ninguno de los sistemas, criterio contrastado con especialistas de la rama de ortopedia que confirmó la obsolescencia del método para este fin.
- Todos los sistemas ofrecen el cálculo del ángulo de Cobb, demostrando que es el método más imprescindible para el diagnóstico de la escoliosis.
- Aunque la mayoría de los sistemas analizados implementan el método de Cobb, dos de ellos en su descripción comercial menciona que presentan herramientas para el cálculo de los parámetros espino pélvicos no se pudo corroborar el funcionamiento de esta característica.
- Los resultados obtenidos tras la investigación de los sistemas homólogos concluyen que No fue posi-

ble obtener el código que implementa esta funcionalidad debido a que los sistemas que lo implementan son de código cerrado o no lo tienen implementado en lenguaje de programación C#.

- No pudo ser posible reutilizar código para las nuevas funcionalidades del sistema XAVIA PACSViewer por lo que se respalda el desarrollo de nuevas funcionalidades del sistema XAVIA PACSViewer para la especialidad de ortopedia.

1.4. Tecnologías utilizadas para el desarrollo

Para el desarrollo de una herramienta que implemente funcionalidades para el sistema XAVIA PACSViewer se tuvo en cuenta la continuidad en el uso de las que el proyecto de desarrollo tiene definidas para garantizar la integridad tecnológica.

Metodología de desarrollo de software

La metodología AUP-UCI es empleada para guiar y documentar el proceso de desarrollo. El escenario en el que se ejecuta esta metodología es el escenario 3 que describe procesos de negocio y describe los requisitos por procesos. Esto facilita la documentación del negocio con procesos bien definidos como los servicios médicos y en específico en el servicio de imagenología de las instituciones de salud.

Para modelar el flujo de procesos se utiliza la notación BPMN (Business Process Model and Notation) utilizando el lenguaje de modelado BPMN v2.0 y además se modelan las clases del sistema y el diagrama de secuencia con el lenguaje UML v3.0 utilizando la herramienta VisualParadigm v17.0

Las metodologías de desarrollo de software son métodos sistemáticos que guían el ciclo de vida del software, desde su concepción hasta el mantenimiento, garantizando calidad, eficiencia y satisfacción del cliente (Gabriel, 2020). La Metodología AUP-UCI, realizada por la Universidad de las Ciencias Informáticas (UCI), integra principios ágiles con prácticas de mejora continua del Modelo de Madurez de Capacidad - Integración (CMMI) (AUP Metodología 2019). Esta metodología mejora la calidad del software combinando flexibilidad y control, adecuada para proyectos académicos y de investigación (Variación de AUP para la UCI (AUP-UCI) – Metodología tradicionales 2020). Esta adaptación fusiona las fases de Inicio, Elaboración, Construcción y Transición en dos fases principales: Inicio y Ejecución, además de una fase de Cierre. En la fase de Inicio se planifica detalladamente el proyecto, en la fase de Ejecución se desarrollan las actividades necesarias ajustando los planes según los requisitos y la arquitectura del sistema, y en la fase de Cierre se evalúan los resultados del proyecto, documentando lecciones aprendidas.

La metodología AUP-UCI propone cuatro escenarios posibles según el contexto y las necesidades del proyecto:

- **Escenario 1:** sugiere emplear casos de uso del sistema (CUS) para abordar los requisitos funcionales del sistema desde una perspectiva técnica, destacando cómo el sistema opera con otros sistemas y elementos técnicos. Esta metodología se prefiere porque se centra en la visión técnica del sistema.
- **Escenario 2:** propone el uso del modelo conceptual (MC) para capturar los elementos clave del negocio y sus conexiones. El MC se orienta hacia la organización y la lógica empresarial, sin profundizar en los detalles de ejecución del sistema. Mientras tanto, los CUS sirven para definir los requisitos funcionales del sistema desde su punto de vista técnico, explicando cómo interactúa con otros sistemas y componentes técnicos.
- **Escenario 3:** se describen los procesos de negocio (DPN) para representar los procedimientos comerciales y las normativas correspondientes. La DPN se enfoca en el funcionamiento del negocio sin considerar a las personas que lo administran. Para los requisitos funcionales del sistema desde una perspectiva técnica, se recomienda el uso de la descripción de requisitos de procesos (DRP), que detalla cómo el sistema integra dichos procesos comerciales.
- **Escenario 4:** indica el uso de historias de usuario (HU) para diseñar las características del sistema desde la perspectiva del usuario, describiendo las interacciones del usuario con el sistema y las capacidades que debe tener para cumplir con las expectativas del usuario. Las HU reemplazan a los casos de uso del negocio y del sistema, debido a que se centran en las necesidades del usuario sin entrar en los detalles técnicos del sistema.

El **escenario 3** es adecuado para proyectos que representan el negocio con el objetivo de facilitar el desarrollo de la nueva funcionalidad para medir el grado de escoliosis y el balance sagital en el visor de imágenes médicas en el sistema Xavia PACSViewer debido a que modela el negocio con descripción de proceso de negocio (DPN) y se modela el sistema con descripción de requisitos de procesos (DRP), adecuado para proyectos que buscan representar el negocio y sus procesos de manera clara y concisa. La descripción de requisitos de procesos (DRP) facilita la integración de procesos de negocio, asegurando un desarrollo de alta calidad adherente a los estándares del CMMI-DEV v1.3, esencial para herramientas de precisión y confiabilidad (Rodríguez, 2020).

Metodología BPMN

La Business Process Model and Notation (BPMN) es una metodología y notación gráfica para modelar procesos de negocio. BPMN facilita la comunicación y comprensión de los procesos internos de una organización, permitiendo a los usuarios representar flujos de trabajo y transacciones de manera estandarizada (*Business Process Model and Notation* 2024). Esta metodología es útil para coordinar procesos y mensajes

entre diferentes participantes en un conjunto de actividades.

UML 3.0

La Unified Modeling Language (UML) versión 3.0 es una notación estándar para modelar sistemas de software y procesos de negocio. Desarrollada por el Object Management Group (OMG), UML 3.0 permite a los desarrolladores y analistas crear diagramas que representan la estructura, el comportamiento y la arquitectura de un sistema de manera visual y comprensible (*Object Management Group: UML 3.0* 2024).

Herramienta CASE Visual Paradigm 17.0

Visual Paradigm versión 17.0 es una herramienta CASE (Computer-Aided Software Engineering) que ofrece una amplia gama de funcionalidades para el diseño, administración y organización de proyectos. Esta herramienta es multiplataforma y soporta UML, BPMN, y otras metodologías de modelado, facilitando la creación de diagramas y la gestión de proyectos de manera eficiente (*Visual Paradigm 17.0* 2022).

Entorno de Desarrollo Integrado (IDE)

El sistema para el cual se desarrolla está implementado en el IDE Visual Studio 2022. Se empleó el marco de trabajo .NET Framework en su versión 4.8. El lenguaje de programación empleado fue C# que se integra perfectamente con el .NET Framework y proporciona una sintaxis simplificada y robusta para el desarrollo de aplicaciones escalables y seguras.

Visual Studio 2022

Visual Studio 2022 es un entorno de desarrollo integrado (IDE) de Microsoft que ofrece herramientas avanzadas para desarrolladores. Este IDE permite crear aplicaciones modernas para diversas plataformas, incluyendo Windows, Android, iOS, y aplicaciones web2. Entre sus características destacan el soporte para múltiples lenguajes de programación (como C Sharp, Visual Basic, C++, HTML, JavaScript, TypeScript, Python, etc.), herramientas de depuración, perfiles, y Git integrado para el control de versiones.

Framework

El .NET Framework 4.8 es una versión del .NET Framework, una plataforma de desarrollo de Microsoft que permite crear aplicaciones para Windows. Esta versión incluye mejoras en rendimiento, seguridad y compatibilidad con versiones anteriores. Es compatible con aplicaciones que utilizan .NET Standard y .NET Core, y proporciona bibliotecas y herramientas para desarrollar aplicaciones de escritorio, web y servicios en la nube.

Lenguaje de Programación

C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft. Es parte del .NET Framework y se utiliza para crear aplicaciones para Windows, web, y dispositivos móviles. C# es conocido por su sintaxis clara y legible, y ofrece características avanzadas como la gestión de memoria automática, excepciones, y la programación asincrónica.

Arquitectura de Software

La norma ISO/IEC/IEEE 42010:2011 define la arquitectura de software como la estructura esencial de un sistema de software, incluyendo componentes, propiedades visibles externamente y las relaciones entre ellos, proporcionando el marco para el desarrollo, implementación, operación y mantenimiento del sistema (ISO/IEC/IEEE 42010:2011 - *Systems and Software Engineering. Architecture Description* 2011). El patrón arquitectónico Modelo-Vista-VistaModelo (MVVM). Este patrón se utiliza en aplicaciones de software, especialmente en interfaces de usuario basadas en tecnologías de enlace de datos como WPF (Windows Presentation Foundation). MVVM deriva del patrón Model-View-Controller (MVC) y se enfoca en la separación de preocupaciones, mejorando la modularidad, facilidad de prueba y mantenimiento del código. Se divide en tres componentes principales:

- **Modelo** (Model): Representa la lógica de negocio y los datos de la aplicación. No contiene lógica específica de la interfaz de usuario y puede consistir en objetos de negocio, servicios y acceso a datos, gestionando el estado de la aplicación y ejecutando operaciones para satisfacer los requisitos de negocio.
- **Vista** (View): Es la representación visual de la información del modelo, compuesta por elementos que los usuarios interactúan, como botones, cuadros de texto y gráficos. La vista presenta datos y captura la interacción del usuario sin contener lógica de negocio.
- **VistaModelo** (ViewModel): Actúa como intermediario entre el modelo y la vista. Contiene la lógica de presentación y maneja las interacciones del usuario, exponiendo datos y comandos a la vista mediante enlaces de datos (bindings) y actualizando la vista según sea necesario (Gabriel, 2020).

El MVVM fue elegido para este propósito debido a sus numerosas ventajas en el desarrollo de aplicaciones orientadas a datos y su capacidad para gestionar complejidades en la interfaz de usuario (Gabriel, 2020).

Conclusiones parciales

- El análisis de los métodos para el diagnóstico de la escoliosis y el balance sagital permitió identificar las mediciones angulares que se realizan en cada uno.
- La caracterización de los sistemas homólogos confirmó el empleo del método de Cobb para el diagnóstico de la escoliosis y desechando el método de Ferguson para este propósito.
- La caracterización de las tecnologías para el desarrollo de software permitió identificar los artefactos que se generan para la documentación de las nuevas funcionalidades.

PROPUESTA DE SOLUCIÓN

En el presente capítulo se hace un análisis del negocio a partir del cual se realiza una descripción de la propuesta de solución. En esta se describen los requisitos funcionales y no funcionales identificados para la solución y se modelan las clases y la secuencia de transacciones por cada requisito.

Características y descripción de la propuesta de solución

El sistema XAVIA PACSViewer permite la visualización de imágenes médicas digitales a los especialistas, facilitando herramientas para realizar diversas mediciones y análisis. El sistema proporciona herramientas y funcionalidades de medición de carácter general, que no están especializadas en áreas específicas como la ortopedia. La interfaz de usuario de XAVIA PACSViewer está diseñada para ser intuitiva y accesible, facilitando la visualización y manipulación de imágenes médicas. Sin embargo, se ha identificado una necesidad de incorporar mediciones específicas que no están disponibles en la versión actual del sistema, como el método de Cobb y los parámetros espino-pélvicos.

Para abordar esta necesidad, se propone el diseño de un módulo específico para ortopedia que se integrará en el sistema XAVIA PACSViewer. Este nuevo módulo, denominado ".ortopedia", incluirá funcionalidades específicas para medir el grado de escoliosis utilizando el método de Cobb y el balance sagital mediante la inclusión de los parámetros espino-pélvicos. Inicialmente, este módulo se presentará como una única ventana llamada ".ortopedia", desde la cual los especialistas podrán acceder a las mediciones especializadas necesarias para el diagnóstico y tratamiento de las deformidades espinales.

La integración de este módulo permitirá a los radiólogos ortopédicos disponer de herramientas específicas que actualmente no están disponibles en sistemas de código abierto y gratuitos, mejorando así la precisión y eficacia de sus diagnósticos. Esta propuesta de solución tiene como objetivo satisfacer las ne-

cesidades clínicas de los especialistas, proporcionando una funcionalidad robusta y fácil de usar dentro del entorno familiar de XAVIA PACSViewer.

2.1. Modelado de los procesos del negocio

Se define el modelado de negocios como una herramienta conceptual que contiene un conjunto de objetos, conceptos y sus relaciones, con el objetivo de expresar la lógica del negocio de una empresa. Proporciona una vista simplificada de la estructura de negocios que actúa como la base para la comunicación, mejoras o innovación, y define los requisitos de los sistemas de información que apoyan la empresa (Sommerville, 2005).

Los procesos en el sector de la salud, todos están bien definidos. En el área de la imagenología, cuando un paciente se realiza un estudio por imágenes médicas en un equipo de adquisición de imágenes, es enviado desde el equipo hasta el servidor de imágenes médicas. Una vez almacenado puede ser consultado por un especialista en un visor de imágenes médicas (éste interpreta el estudio como se pone de manifiesto en la actividad 19 del diagrama del proceso de negocio sin sistema). Con el sistema XAVIA PACSViewer el especialista debe descargar la imagen y una vez abierto el visor puede realizar mediciones angulares con la funcionalidad de ángulo (Figura 7). Esta funcionalidad no es la ideal para aplicar debido a que las líneas a trazar para el ángulo pueden no cortarse en el plano de la imagen y por el método de Cobb es necesario trazar líneas perpendiculares que se corten para luego calcular el ángulo marcando 3 puntos exactamente sobre las líneas trazadas coincidiendo el segundo con el vértice del ángulo sobre el punto donde se cortan las perpendiculares.

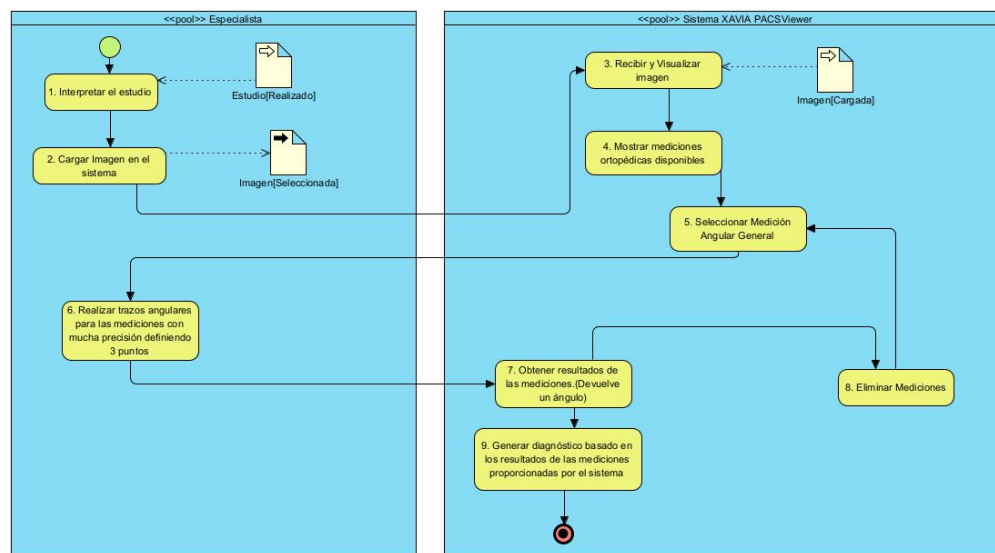


Figura 7. Diagrama del proceso de negocio

2.2. Requisitos del sistema

Requisitos Funcionales

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información (Somerville, 2005). Un requisito funcional es una declaración de cómo debe comportarse un sistema. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. Los requisitos funcionales se pueden considerar como características que el usuario detecta. Son diferentes de los requisitos no funcionales, que definen cómo debe funcionar internamente el sistema (*What are Functional Requirements?* 2023). A continuación, se muestran los requisitos definidos y agregados necesarios para incorporar la funcionalidad definida. Para cumplir con el objetivo de la investigación de desarrollar una funcionalidad específica para la específicas en el sistema XAVIA PACS Viewer, se identificaron los siguientes requisitos funcionales:

- RF1. Medir ángulo Cobb
- RF2. Medir PI
- RF3. Medir PT
- RF4. Medir SS

Otros requisitos auxiliares que se identificaron y se agregaron a la lista:

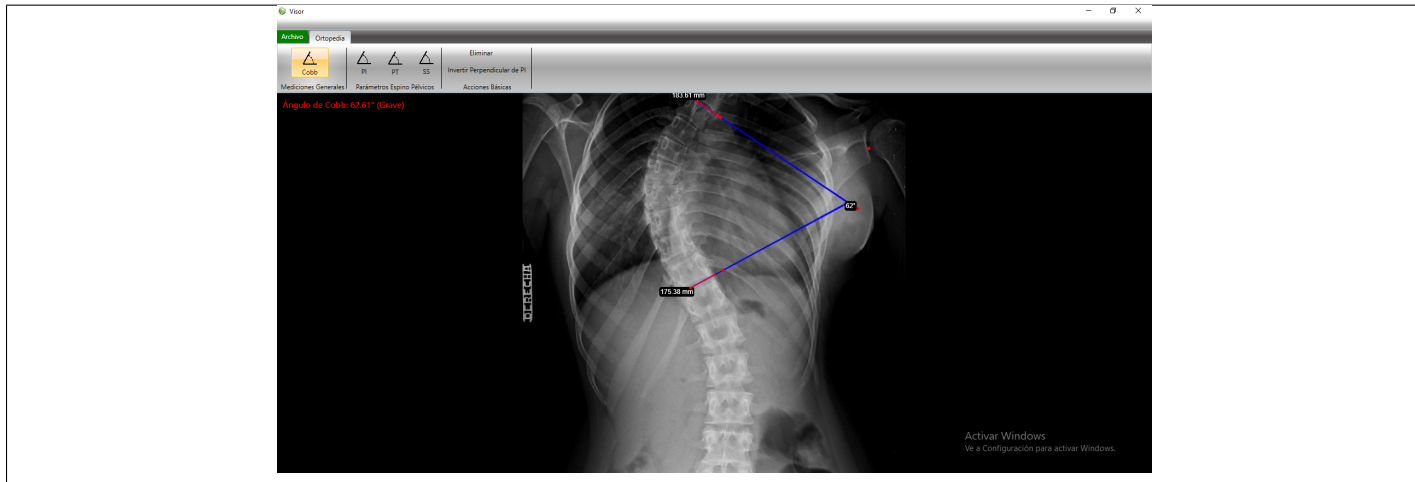
- RF 5. Cargar imagen
- RF 6. Invertir perpendicular de PI
- RF 7. Eliminar trazo

RF1. Medir Ángulo de Cobb

Tabla 2. Descripción del RF 1 Medir Ángulo de Cobb.

Descripción textual	Permite medir el grado de escoliosis por el método de Cobb mediante el trazado de líneas.
Actor	Especialista (inicia).
Precondiciones	Seleccionar una medición.
Flujo de eventos	
Flujo básico Medir Ángulo de Cobb	
1.	El Especialista selecciona la opción Cobb en la sección Mediciones Generales de la pestaña Ortopedia del visor.

2.	El Sistema carga y visualiza una imagen de una radiografía en el plano frontal. Ver RF 5. Cargar imagen.
3.	El Sistema resalta la medición seleccionada.
4.	El Especialista da clic sobre la imagen para indicar el punto de inicio de la primera línea.
5.	El Especialista mueve el mouse hasta la posición en la que colocará el segundo punto de fin de la primera línea y da clic en dicha posición.
6.	El Especialista da clic de nuevo sobre la imagen para indicar el punto de inicio de la segunda línea.
7.	El Especialista mueve el mouse hasta la posición en la que colocará el segundo punto de fin de la segunda línea y da clic en dicha posición.
8.	El Sistema dibuja la medición.
9.	El Sistema muestra el resultado de la medición en grados.
Pos-condiciones	
1.	Se calcula la medición y se mostró el resultado de la misma en grados.
Flujos alternativos	
Flujo alternativo 1 'Trazar una sola línea'	
1.	El Especialista traza una sola línea.
2.	El Sistema no mostrará ningún ángulo hasta que se tracen dos líneas.
Flujo alternativo 2 'Trazar líneas fuera del panel y del espacio establecido'	
1.	El Sistema no mostrará ningún ángulo hasta que no se tracen dos líneas dentro del panel de forma correcta.
2.	El Especialista revisa y corrige el trazado de las líneas dentro del espacio establecido.
Flujo alternativo 3 'Trazar una tercera línea después de haber trazado dos'	
1.	El Especialista al trazar una tercera línea, las dos anteriores desaparecerán, actualizando así el panel.
2.	Regresar al paso 5 del flujo básico.
Flujo alternativo 4 'Trazar una línea y presionar eliminar'	
1.	El Especialista después de trazar una línea, toca el botón eliminar
2.	El Sistema borra la línea trazada.
Requisitos especiales	N/A
Asuntos pendientes	N/A
Prototipo elemental de interfaz gráfica de usuario	



RF2. Medir Ángulo de Incidencia Pélvica (PI)

Tabla 3. Descripción del RF2 Medir Ángulo de Incidencia Pélvica (PI).

Descripción textual	Permite medir el ángulo de Incidencia Pélvica mediante el trazado de una línea y un clic.
Actor	Especialista (inicia).
Precondiciones	Seleccionar una medición.
Flujo de eventos	
Flujo básico Medir Ángulo de Incidencia Pélvica	
1.	El Especialista selecciona la opción PI en la sección Parámetros Espino Pélvicos.
2.	El Sistema carga y visualiza una imagen de una radiografía en el plano sagital. Ver RF 5. Cargar imagen.
3.	El Sistema resalta la medición seleccionada.
4.	El Especialista da clic sobre la imagen para indicar el punto de inicio de la primera línea.
5.	El Especialista mueve el mouse hasta la posición en la que colocará el segundo punto de fin de la primera línea y da clic en dicha posición.
6.	El Sistema dibuja la perpendicular de la línea trazada.
7.	El Especialista da clic de nuevo sobre la imagen para indicar el punto medio de las cabezas femorales.
8.	El Sistema dibuja un punto donde el especialista hizo clic.
9.	El Sistema dibuja una línea que va desde ese punto hasta el punto medio de la primera línea trazada.
10.	El Sistema dibuja la medición.
11.	El Sistema muestra el resultado de la medición en grados.
Pos-condiciones	
1.	Se calcula la medición y se mostró el resultado de la misma en grados.
Flujos alternativos	

Flujo alternativo 1 'Trazar una línea y luego no hacer clic'	
1.	El Especialista traza una sola línea.
2.	El Sistema no mostrará ningún ángulo hasta que se trace una línea y luego se haga clic.
Flujo alternativo 2 'Traza una línea y hacer clic fuera del panel y del espacio establecido'	
1.	El Sistema no mostrará ningún ángulo hasta que no se trace una línea y después se haga clic dentro del panel.
2.	El Especialista revisa y corrige el trazado de la línea y el clic dentro del espacio establecido.
Flujo alternativo 3 'Hacer un segundo clic después de haber trazado una línea y haber hecho un clic'	
1.	El Especialista al hacer otro clic, actualizará la medición y el dibujo en el panel.
2.	Regresa al paso 9 del flujo básico.
Flujo alternativo 4 'Trazar una línea y presionar eliminar'	
1.	El Especialista después de trazar una línea, toca el botón eliminar
2.	El Sistema borra la línea trazada.
Requisitos especiales	N/A
Asuntos pendientes	N/A
Prototipo elemental de interfaz gráfica de usuario	
	

RF3. Medir Ángulo de Versión Pélvica (PT)

Tabla 4. Descripción del RF3 Medir Ángulo de Versión Pélvica (PT).

Descripción textual	Permite medir el ángulo de Versión Pélvica mediante el trazado de una línea y un clic.
Actor	Especialista (inicia).
Precondiciones	Seleccionar una medición.
Flujo de eventos	
Flujo básico Medir Ángulo de Versión Pélvica	
1.	El Especialista selecciona la opción PT en la sección Parámetros Espino Pélvicos.

2.	El Sistema carga y visualiza una imagen de una radiografía en el plano sagital.Ver RF 5. Cargar imagen.
3.	El Sistema resalta la medición seleccionada.
4.	El Especialista da clic sobre la imagen para indicar el punto de inicio de la primera línea.
5.	El Especialista mueve el mouse hasta la posición en la que colocará el segundo punto de fin de la primera línea y da clic en dicha posición.
6.	El Sistema dibuja el punto medio de la línea trazada.
7.	El Especialista da clic de nuevo sobre la imagen para indicar el punto medio de las cabezas femorales.
8.	El Sistema dibuja un punto donde el especialista hizo clic.
9.	El Sistema dibuja una línea que va desde ese punto hasta el punto medio de la primera línea trazada.
10.	El Sistema dibuja una línea vertical que comienza donde el especialista hizo clic hacia arriba.
11.	El Sistema dibuja la medición.
12.	El Sistema muestra el resultado de la medición en grados.
Pos-condiciones	
1.	Se calcula la medición y se mostró el resultado de la misma en grados.
Flujos alternativos	
Flujo alternativo 1 'Trazar una línea y luego no hacer clic'	
1.	El Especialista traza una sola línea.
2.	El Sistema no mostrará ningún ángulo hasta que se trace una línea y luego se haga clic.
Flujo alternativo 2 'Traza una línea y hacer clic fuera del panel y del espacio establecido'	
1.	El Sistema no mostrará ningún ángulo hasta que no se trace una línea y después se haga clic dentro del panel.
2.	El Especialista revisa y corrige el trazado de la línea y el clic dentro del espacio establecido.
Flujo alternativo 3 'Hacer un segundo clic después de haber trazado una línea y haber hecho un clic'	
1.	El Especialista al hacer otro clic, actualizará la medición y el dibujo en el panel.
2.	Regresa al paso 9 del flujo básico.
Flujo alternativo 4 'Trazar una línea y presionar eliminar'	
1.	El Especialista después de trazar una línea, toca el botón eliminar
2.	El Sistema borra la línea trazada.
Requisitos especiales	N/A
Asuntos pendientes	N/A
Prototipo elemental de interfaz gráfica de usuario	



RF4. Medir Ángulo de Inclinación de Sacro (SS)

Tabla 5. Descripción del RF4 Medir Ángulo de Inclinación de Sacro (SS).

Descripción textual	Permite medir el ángulo de Inclinación de Sacro mediante el trazado de una línea.
Actor	Especialista (inicia).
Precondiciones	Seleccionar una medición.
Flujo de eventos	
Flujo básico Medir Ángulo de Inclinación de Sacro	
1.	El Especialista selecciona la opción SS en la sección Parámetros Espino Pélvicos.
2.	El Sistema carga y visualiza una imagen de una radiografía en el plano sagital. Ver RF 5. Cargar imagen.
3.	El Sistema resalta la medición seleccionada.
4.	El Especialista da clic sobre la imagen para indicar el punto de inicio de la primera línea.
5.	El Especialista mueve el mouse hasta la posición en la que colocará el segundo punto de fin de la primera línea y da clic en dicha posición.
6.	El Sistema dibuja una línea horizontal hacia la derecha que comienza en el punto más inferior de la 1ra línea.
7.	El Sistema dibuja la medición.
8.	El Sistema muestra el resultado de la medición en grados.
Pos-condiciones	
1.	Se calcula la medición y se mostró el resultado de la misma en grados.
Flujos alternativos	
Flujo alternativo 1 'Trazar una segunda línea después de haber trazado una'	
1.	El Especialista al trazar una segunda línea, la anterior desaparecerá, actualizando así el panel.
2.	Regresar al paso 5 del flujo básico.
Flujo alternativo 2 'Trazar una línea y hacer clic fuera del panel y del espacio establecido'	
1.	El sistema no mostrará ningún ángulo hasta que no se trace una línea dentro del panel de forma correcta.

2.	El Especialista revisa y corrige el trazado de la línea dentro del espacio establecido.
Flujo alternativo 4 'Trazar una línea y presionar eliminar'	
1.	El Especialista después de trazar una línea, toca el botón eliminar
2.	El Sistema borra la línea trazada.
Requisitos especiales	N/A
Asuntos pendientes	N/A
Prototipo elemental de interfaz gráfica de usuario	
	

Requisitos Funcionales Agregados

Los requisitos funcionales agregados se refieren a los nuevos requerimientos que se añaden al sistema después de su definición inicial. Estos pueden incluir nuevas funcionalidades, mejoras en las existentes o adaptaciones necesarias para que el sistema siga siendo relevante y útil para los usuarios.

Incorporar estos requisitos funcionales agregados de manera efectiva es crucial para el éxito continuo del proyecto. Requiere un proceso de gestión de cambios robusto, que incluya la evaluación del impacto, la planificación de la implementación y la comunicación clara con todas las partes interesadas. Se definieron 3 requisitos funcionales agregados: Cargar Imagen, Invertir Perpendicular de PI y Eliminar Trazo, la descripción de estos se puede encontrar en la sección de Apéndices.

Requisitos No Funcionales

Los requisitos no funcionales son características del sistema que definen criterios de calidad y restricciones, tales como rendimiento, seguridad, usabilidad y mantenibilidad, que no están directamente relacionados con las funciones específicas del software (Sommerville, 2011). Estos requisitos son fundamentales para asegurar que el sistema funcione de manera eficiente, segura y fiable bajo diferentes condiciones de

uso.

En el contexto del desarrollo del módulo de Ortopedia para el sistema XAVIA PACSViewer, se han asumido los requisitos no funcionales ya establecidos por el sistema(Figura 8), por lo que es imperativo que el nuevo módulo siga estos mismos criterios para garantizar la coherencia y la compatibilidad. Esto asegura que las nuevas funcionalidades se integren adecuadamente dentro del entorno existente sin comprometer la calidad o el rendimiento general del sistema.

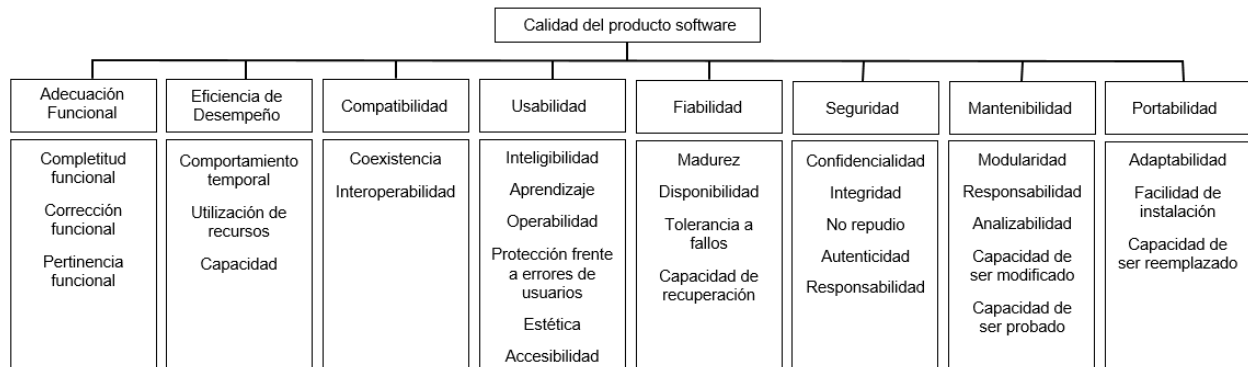


Figura 8. Requisitos No Funcionales.

Descripción de Requisitos No Funcionales

Tabla 6. Descripción del RNF 1 de usabilidad

Atributo de Calidad	Usabilidad
Sub-atributos/Sub-características	Inteligibilidad
Objetivo	Elaborar materiales de apoyo que ilustren el funcionamiento del sistema XAVIA PACSViewer, mediante los cuales los usuarios puedan reconocer si este es apropiado para satisfacer sus necesidades.
Origen	Usuario
Artefacto	Manuales de usuario y Videos.
Entorno	El sistema se encuentra funcionando correctamente y el usuario cuenta con el material de apoyo suficiente.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Acceso a los manuales de usuario	
El usuario accede a los manuales de usuario.	Queda identificado el funcionamiento del sistema, mediante las imágenes y descripciones de las funcionalidades descritas en el manual.
1.b Acceso a los videos	
El usuario accede a los videos tutoriales.	Queda identificado el funcionamiento del sistema, mediante los flujos de información y descripciones de las funcionalidades descritas en el manual.
Medida de respuesta	
El sistema cuenta con manuales de usuario y videos tutoriales.	

Tabla 7. Descripción del RNF 2 de usabilidad

Atributo de Calidad	Usabilidad
Sub-atributos/Sub-características	Inteligibilidad
Objetivo	Identificar las funcionalidades por un nombre, ícono y figura ilustrativa que brinde información al usuario.
Origen	Sistema
Artefacto	Sistema
Entorno	El sistema funcionando correctamente.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Acceso a una de las funcionalidades del sistema XAVIA PACSViewer	
El usuario accede a una de las funcionalidades del sistema.	Al acceder a una de las funcionalidades del sistema XAVIA PACSViewer, se muestra una breve descripción de esta (al pasar el mouse por encima del ícono de dicha funcionalidad).
Medida de respuesta	
Se visualiza una breve descripción de la funcionalidad. Cada funcionalidad se encuentra identificada por un ícono representativo y su nombre ilustra su objetivo en el sistema.	

Tabla 8. Descripción del RNF 3 de usabilidad

Atributo de Calidad	Usabilidad
Sub-atributos/Sub-características	Operabilidad
Objetivo	Facilitar el uso y navegación en el sistema por parte del usuario, mediante el uso de pautas de diseño que estandaricen las funcionalidades y flujos de información en el sistema.
Origen	Sistema
Artefacto	Interfaz de usuario
Entorno	El sistema funcionando correctamente.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Interacción con diferentes funcionalidades del sistema	
Los usuarios interactúan con diferentes funcionalidades del sistema XAVIA PACSViewer.	El sistema permite una fácil navegación por las funcionalidades con diseño o comportamientos similares.
Medida de respuesta	
Las funcionalidades del sistema presentan un diseño o comportamiento estándar.	

Tabla 9. Descripción del RNF 4 de usabilidad

Atributo de Calidad	Usabilidad
Sub-atributos/Sub-características	Protección frente a errores de usuarios
Objetivo	Notificar en el sistema la existencia de datos incorrectos o incompletos, así como confirmaciones de acciones que así lo requieran.
Origen	Sistema
Artefacto	Sistema
Entorno	El sistema funcionando correctamente.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Acceso a una de las funcionalidades del sistema XAVIA PACSViewer	
El usuario accede a una de las funcionalidades del sistema y realiza una acción.	Una vez completada la acción el sistema notifica la existencia de datos incorrectos o incompletos y/o realiza las confirmaciones sobre las acciones que así lo requieran.
Medida de respuesta	
El sistema notifica al usuario la existencia de datos incorrectos o incompletos cuando accede a una funcionalidad. Para las funcionalidades que así lo requieran, el sistema muestra mensajes de confirmación.	

Tabla 10. Descripción del RNF de eficiencia de desempeño

Atributo de Calidad	Eficiencia de desempeño
Sub-atributos/Sub-características	Utilización de recursos
Objetivo	Ejecutar el sistema XAVIA PACSViewer con los requerimientos necesarios.
Origen	Cliente
Artefacto	El sistema
Entorno	El sistema no se encuentra en ejecución.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Ejecutar el sistema con los requerimientos necesarios para su funcionamiento	Se ejecuta el sistema XAVIA PACSViewer con los requerimientos necesarios para su funcionamiento.

• Características de Software: ◦ Computadora con Sistema Operativo (SO) Windows XP sp3 y Marco de trabajo .NET Framework 4.0. • Características del Hardware: ◦ Para la consulta y visualización de imágenes, así como, quemado de estudios imagenológicos, transcripción y/o impresión de reportes diagnósticos: ▪ HDD: 80 GB (la capacidad de almacenamiento de las estaciones de trabajo tiene relación directa a la cantidad de estudios que permitirá almacenar de manera local y temporal para su visualización.) ▪ TR: 100 Mbps ▪ RAM: 1 GB ▪ CPU: Pentium 4 3.0 Ghz ◦ Para el diagnóstico mediante el empleo de la reconstrucción multiplanar y por referencia: ▪ HDD: 80 GB ▪ TR: 100 Mbps ▪ RAM: 2 GB ▪ CPU: Pentium 4 3.0 Ghz ◦ Para diagnóstico mediante el empleo de la reconstrucción 3D: ▪ HDD: 80 GB ▪ TR: 100 Mbps ▪ RAM: 4 GB ▪ CPU: Dual Core 2.5 Ghz ▪ TV: 512 MB • Leyenda: ◦ TV: Tarjeta de video. ◦ TR: Tarjeta de red. ◦ RAM: Memoria de acceso aleatorio. ◦ HDD: Disco duro. ◦ CPU: Procesador. ◦ SO: Sistema operativo. ◦ FW: Marco de trabajo (Framework).	
Javier Alejandro Romero Petrova	32

Medida de respuesta
El sistema se encuentra disponible y funciona correctamente. En caso que no se cumplan estos requerimientos el sistema no funcionará correctamente.

Tabla 11. Descripción del RNF 1 de mantenibilidad

Atributo de Calidad	Mantenibilidad
Sub-atributos/Sub-características	Capacidad de ser probado
Objetivo	Establecer criterios de prueba para verificar la calidad del sistema.
Origen	Probador
Artefacto	El sistema XAVIA PACSViewer
Entorno	El sistema XAVIA PACSViewer se encuentra funcionando correctamente.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Ejecución de los casos de prueba	
El probador ejecuta las pruebas a partir del diseño de los casos de prueba definidos para cada requisito.	La respuesta del sistema se encuentra en correspondencia con el diseño de caso de prueba definido para cada requisito.
Medida de respuesta	
Existe un diseño de caso de prueba por cada requisito.	

Tabla 12. Descripción del RNF 2 de mantenibilidad

Atributo de Calidad	Mantenibilidad
Sub-atributos/Sub-características	Reusabilidad
Objetivo	Permitir la reutilización de elementos comunes en el desarrollo del sistema.
Origen	Desarrolladores
Artefacto	El sistema
Entorno	El sistema se encuentra en desarrollo.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Reutilización	
El desarrollador reutiliza componentes y plantillas.	El sistema funciona correctamente.
Medida de respuesta	
El sistema funciona correctamente y los componentes y/o plantillas reutilizadas se encuentran acorde a las pautas de diseño definidas.	

Tabla 13. Descripción del RNF de portabilidad

Atributo de Calidad	Portabilidad
Sub-atributos/Sub-características	Capacidad de ser reemplazado (reemplazabilidad)
Objetivo	Permitir la reemplazabilidad del sistema por una nueva versión del propio sistema.
Origen	Usuario
Artefacto	Todo el sistema
Entorno	El sistema funcionando correctamente.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Actualizar versión del sistema XAVIA PACSViewer	
El usuario actualiza la versión del sistema XAVIA PACSViewer.	El sistema funciona correctamente.
Medida de respuesta	
Sistema funcionando correctamente con las nuevas actualizaciones.	

2.3. Arquitectura de Software

La arquitectura Modelo-Vista-VistaModelo (MVVM) se utilizó en el diseño del programa para separar claramente las responsabilidades entre la interfaz de usuario, la lógica de presentación y los datos subyacentes. En este programa, la Vista se encarga de la representación visual y la interacción del usuario, im-

plementada a través de archivos XAML como MainWindow.xaml, AnguloIncidenciaControl.xaml, CobbControl.xaml, InclinacionSacroControl.xaml y VersionPelvicaControl.xaml. La VistaModelo actúa como un intermediario que mantiene la lógica de presentación y maneja los comandos, como se ve en las clases MainViewModel, AnguloIncidenciaViewModel, CobbViewModel, InclinacionSacroViewModel y VersionPelvicaViewModel. Estas clases exponen propiedades y comandos que la Vista puede enlazar a través de mecanismos de binding, como el comando LimpiarCommand y la propiedad AnguloTexto. Aunque no hay un Modelo explícito en este caso debido a la ausencia de una capa de datos compleja, la estructura MVVM facilita la extensión futura para incluir modelos de datos si es necesario. El uso de RelayCommand encapsula la lógica de los comandos, asegurando que las interacciones del usuario en la Vista se traduzcan en acciones ejecutadas por la VistaModelo. Esta separación de responsabilidades permite que el código sea más mantenible, testeable y escalable, siguiendo los principios de diseño de MVVM.

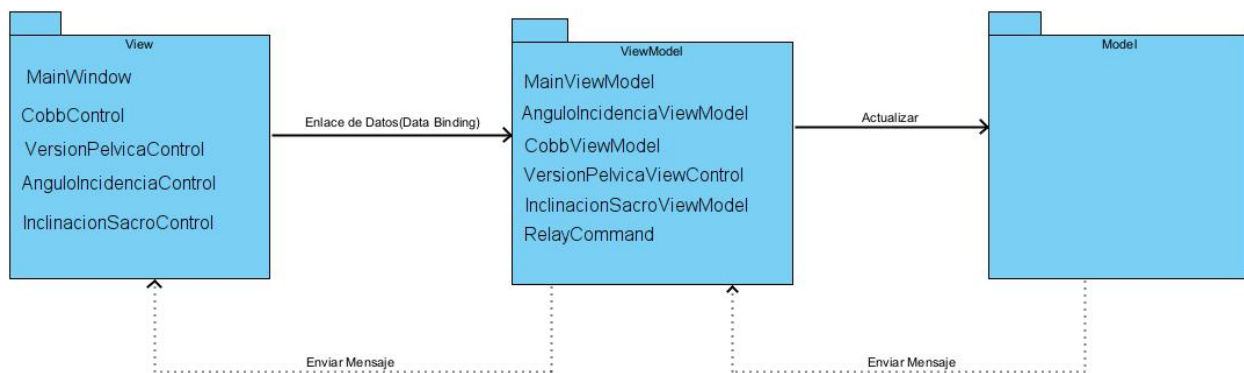


Figura 9. Patrón Arquitectónico: Modelo-Vista-VistaModelo

2.4. Modelo de Datos

El modelo de datos es una parte fundamental de muchas aplicaciones que sigue la arquitectura Modelo-Vista-VistaModelo (MVVM). Un modelo de datos define cómo se estructuran y gestionan los datos en una aplicación. Esta capa es responsable de interactuar con las fuentes de datos, ya sean bases de datos, servicios web u otros medios de almacenamiento. En el contexto de MVVM, el modelo encapsula la lógica de negocio y los datos, proporcionando una interfaz clara para que los ViewModels interactúen con ellos sin preocuparse por los detalles de implementación de la fuente de datos.

En esta aplicación específica, no se ha implementado un modelo explícito debido a la ausencia de una capa de datos compleja o una fuente de datos subyacente, como bases de datos o servicios web. Las necesidades de datos en esta aplicación son mínimas y se manejan directamente en los ViewModels, lo que elimina la necesidad de una capa de modelo separada. Esta decisión se basa en la simplicidad del proyecto y la lógica de negocio limitada que no requiere una estructura de datos sofisticada.

A pesar de la ausencia de un modelo de datos explícito, la aplicación sigue cumpliendo con los principios de la arquitectura MVVM. Los ViewModels actúan como intermediarios entre las vistas y la lógica de presentación, gestionando los comandos y las propiedades que la interfaz de usuario puede enlazar mediante el binding. Esta separación clara de responsabilidades asegura que el código sea mantenible y escalable, permitiendo que las vistas se mantengan libres de lógica de negocio y que los ViewModels se encarguen de la manipulación de datos y la lógica de presentación. En un futuro, si la aplicación necesita manejar datos más complejos, se puede integrar fácilmente una capa de modelo sin afectar la estructura existente, gracias a la flexibilidad que ofrece la arquitectura MVVM.

Diagrama de clases

El diagrama de clases de este proyecto sigue la arquitectura Modelo-Vista-VistaModelo (MVVM) y define claramente las relaciones entre las diferentes clases de la aplicación, facilitando la separación de responsabilidades y la mantenibilidad del código. En este diseño, MainWindow actúa como la vista principal y se asocia con MainViewModel, que se establece como su DataContext. Esta asociación se representa mediante una línea continua, indicando que MainWindow depende de MainViewModel para la lógica de presentación (J. e. a. Smith, 2020).

MainWindow se asocia con MainViewModel, que actúa como su DataContext. MainViewModel contiene referencias a varios RelayCommand para manejar comandos, y se asocia con varias vistas específicas (AnguloIncidenciaControl, CobbControl, InclinationSacroControl, VersionPelvicaControl). Estas vistas están fuertemente acopladas a sus respectivos ViewModels (AnguloIncidenciaViewModel, CobbViewModel, InclinationSacroViewModel, VersionPelvicaViewModel) mediante composición, lo que asegura que la vida del ViewModel esté completamente ligada a la vida de la vista correspondiente. Este diseño facilita una clara separación de responsabilidades, gestión del ciclo de vida y mantenibilidad del código, cumpliendo con los principios de la arquitectura MVVM.



Los diagramas de secuencia son un tipo de diagrama de interacción utilizado en UML (Lenguaje Unificado de Modelado) para ilustrar cómo los objetos interactúan en un sistema a lo largo del tiempo. Estos diagramas muestran la secuencia de mensajes que se envían entre los objetos para llevar a cabo una función o proceso específico. Son particularmente útiles para entender el flujo de control y la colaboración entre diferentes componentes de un sistema (**Jacobson2005**). En este proyecto, se realizaron cuatro diagramas de secuencia, cada uno correspondiente a un requisito funcional clave: medir Ángulo de Cobb, medir Ángulo de Incidencia Pélvica, medir Ángulo de Versión Pélvica y medir Ángulo de Inclinación de Sacro. Cada uno de estos diagramas proporciona una visión clara y detallada de cómo el sistema cumple con los requisitos funcionales específicos, asegurando una comprensión completa y precisa de los procesos involucrados.

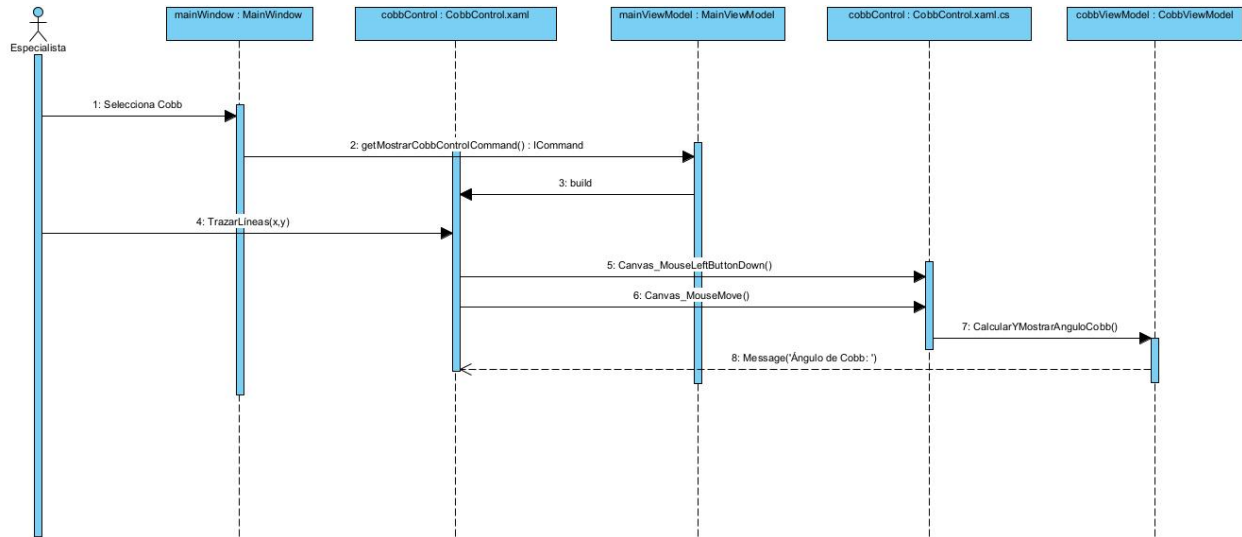


Figura 11. Diagrama de Secuencia Cobb.

En el diagrama de secuencia para medir el Ángulo de Cobb, el actor selecciona esta medición en la ventana MainWindow. La solicitud se envía al MainViewModel, que muestra la vista correspondiente en CobbControl.xaml. El especialista traza las líneas de medición en CobbControl.xaml, ejecutando los métodos CanvasMouseLeftButtonDown() y CanvasMouseMove() en CobbControl.xaml.cs. Posteriormente, el control se transfiere a CobbViewModel donde se ejecuta el método CalcularYMostrarAngulo(). Finalmente, el CobbViewModel envía el ángulo calculado de regreso a CobbControl.xaml para su visualización.

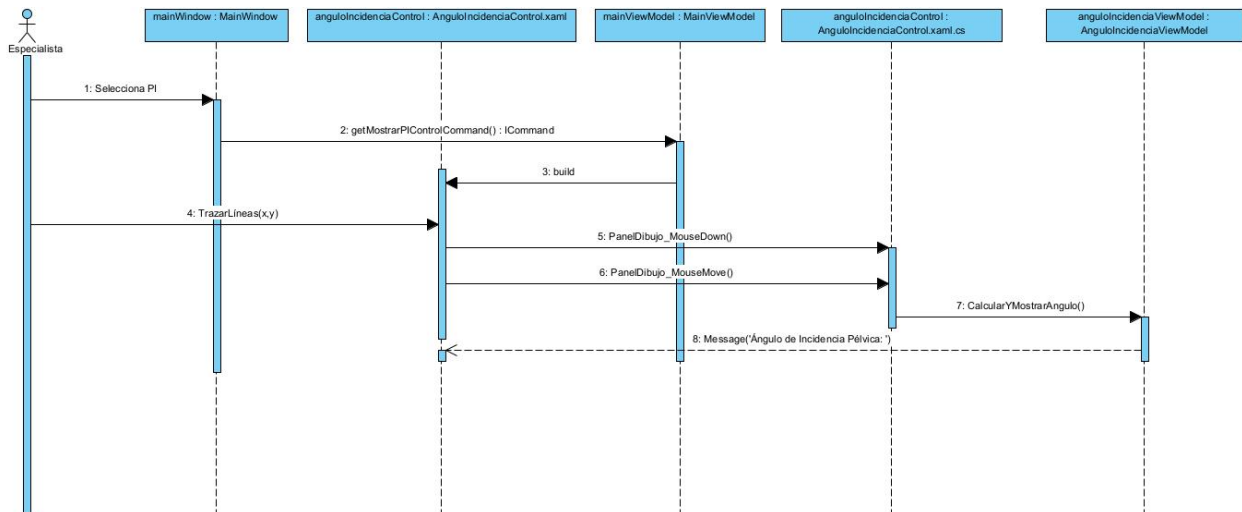


Figura 12. Diagrama de Secuencia Incidencia Pélvica.

Para medir el Ángulo de Incidencia Pélvica (PI), el proceso es similar. El actor selecciona la medición

en MainWindow, enviando la solicitud al MainViewModel, que muestra la vista en AnguloIncidenciaControl.xaml. El especialista traza las líneas y ejecuta los métodos PanelDibujoMouseDown() y PanelDibujoMouseMove() en AnguloIncidenciaControl.xaml.cs. Luego, el control se transfiere a AnguloIncidenciaViewModel para ejecutar CalcularYMostrarAngulo(), devolviendo el ángulo calculado a AnguloIncidenciaControl.xaml.

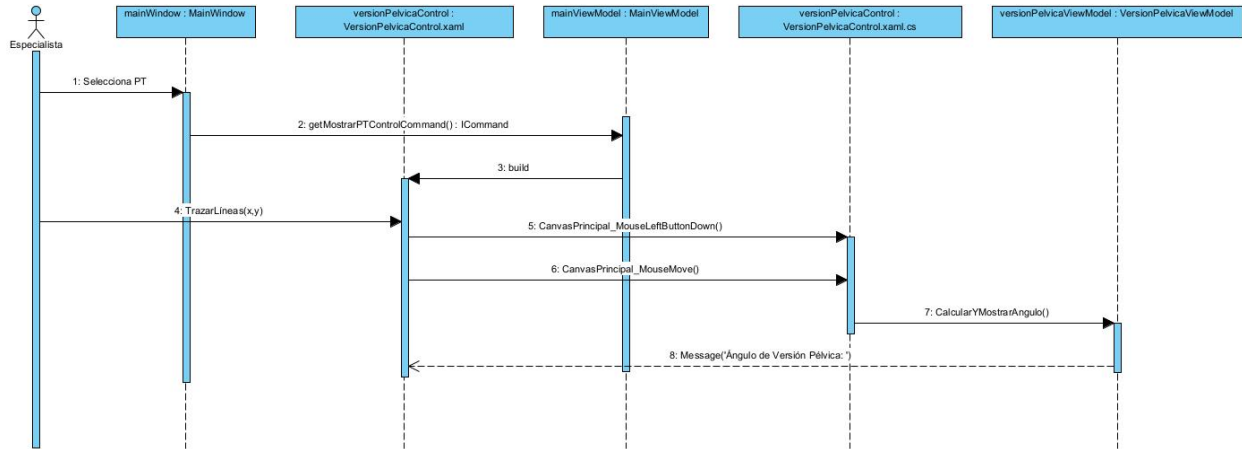


Figura 13. Diagrama de Secuencia Versión Pélvica.

Para medir el Ángulo de Versión Pélvica (PT), el actor selecciona esta opción en MainWindow, y la solicitud se maneja de manera similar a través de MainViewModel, que muestra la vista en VersionPelvicaControl.xaml. El especialista realiza las mediciones y traza las líneas ejecutando CanvasPrincipalMouseDown() y CanvasPrincipalMouseMove() en VersionPelvicaControl.xaml.cs. A continuación, VersionPelvicaViewModel ejecuta CalcularYMostrarAngulo(), enviando el ángulo calculado de vuelta a VersionPelvicaControl.xaml.

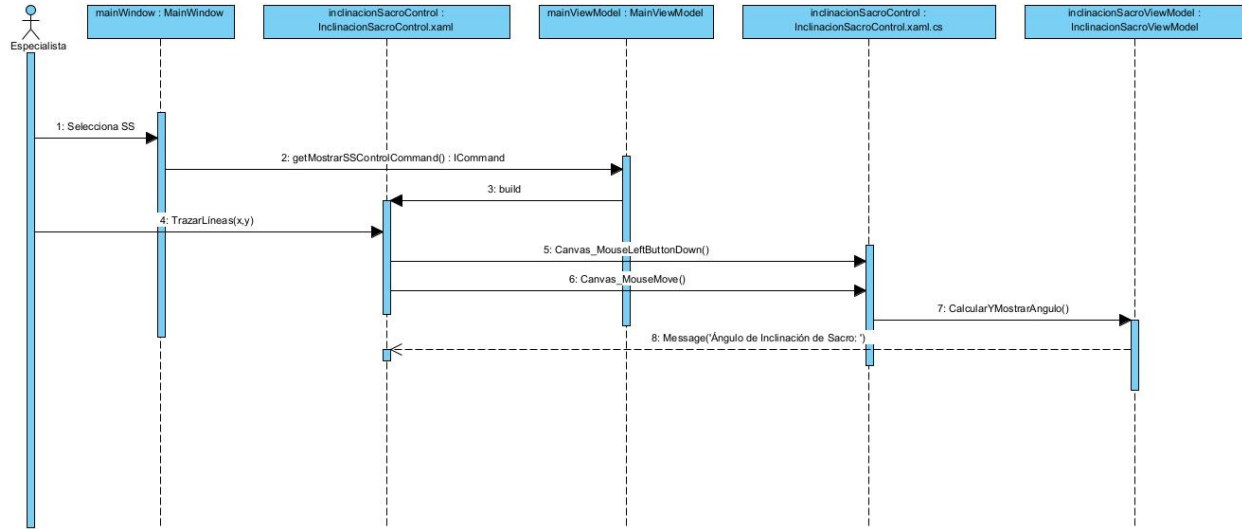


Figura 14. Diagrama de Secuencia Inclinação de Sacro.

Por último, para medir el Ángulo de Inclinação de Sacro (SS), el actor selecciona la medición en Main-Window, enviando la solicitud al MainViewModel, que muestra la vista en InclinacionSacroControl.xaml. El especialista traza las líneas necesarias y ejecuta CanvasMouseLeftButtonDown() y CanvasMouseMove() en InclinacionSacroControl.xaml.cs. Posteriormente, el método CalcularYMostrarAngulo() en InclinacionSacroViewModel se ejecuta, y el ángulo calculado se devuelve a InclinacionSacroControl.xaml para su visualización.

2.6. Patrones de diseño

Los patrones de diseño son soluciones reutilizables y probadas para problemas comunes que surgen en el diseño del software. Son descripciones o plantillas para solucionar problemas que pueden ser utilizados en muchas situaciones diferentes (Vlissides1994).

En la solución desarrollada se utilizaron varios patrones de diseño GRASP (General Responsibility Assignment Software Patterns).

Controlador: Este patrón se refiere a una clase que maneja eventos del sistema generados por la interacción del usuario, como los eventos de la interfaz de usuario. En este caso, MainViewModel y los diferentes controles (CobbControl.xaml, AnguloIncidenciaControl.xaml, VersionPelvicaControl.xaml e InclinacionSacroControl.xaml) actúan como controladores, gestionando la interacción del usuario y manejando eventos como clics del mouse y cambios en la interfaz.

Creador: Una clase es responsable de crear instancias de otras clases. MainWindow actúa como creador al instanciar el MainViewModel y asignarlo como su DataContext. Además, cada control puede ser considerado un creador de su respectivo ViewModel.

Experto: Este patrón asigna la responsabilidad de manejar información a la clase que tiene la información necesaria para cumplir con la tarea. Cada ViewModel (AnguloIncidenciaViewModel, CobbViewModel, InclinacionSacroViewModel, VersionPelvicaViewModel) maneja la lógica de negocio y los datos específicos necesarios para las mediciones que realizan, actuando como expertos en la información de su dominio.

Además, se han identificado varios patrones GoF (Gang of Four), definidos como patrones que proporcionan soluciones elegantes y reutilizables para problemas comunes en el diseño de software orientado a objetos (Noorullah; 2019).

Comando: Este patrón encapsula una solicitud como un objeto, permitiendo parametrizar a los clientes con diferentes solicitudes, colas o registros de solicitudes y operaciones que se pueden deshacer. RelayCommand es una implementación de este patrón que permite encapsular acciones como comandos, facilitando la gestión de la ejecución de comandos desde la vista a los ViewModels.

Observador: Define una dependencia de uno a muchos entre objetos para que cuando un objeto cambie su estado, todos sus dependientes sean notificados y actualizados automáticamente. El enlace de datos (Data Binding) en la arquitectura MVVM utiliza este patrón, permitiendo que las vistas observen los cambios en los ViewModels y se actualicen automáticamente cuando hay cambios en las propiedades.

Estrategia: Este patrón se observa en el uso de diferentes ViewModels para distintas mediciones, donde cada ViewModel encapsula la lógica de cálculo específica para su tipo de medición, permitiendo que el algoritmo varíe independientemente de los clientes que lo usan.

Conclusiones parciales

- La modelación del negocio permitió comprender el proceso de diagnóstico por imágenes médicas en las instituciones de salud.
- La descripción de los requisitos permitió definir las acciones del sistema ante cada acción del especialista.
- La modelación de las clases y las transacciones entre ellas permitió diseñar el sistema y sus acciones principales en una arquitectura MVVM que de respuesta a las peticiones del usuario.

IMPLEMENTACIÓN Y PRUEBAS

En el presente capítulo se realiza la descripción de algunas buenas prácticas utilizadas en la implementación de la solución y se planten algunos resultados de la estrategia de pruebas aplicada para garantizar la calidad de la solución desarrollada.

3.1. Estándares de codificación

Según Guido van Rossum (científico de la computación, conocido por ser el autor del lenguaje de programación Python), el código es leído muchas más veces de lo que es escrito. Los estándares de programación son reglas que se aplican para lograr uniformidad en el código producido por un grupo de desarrollo de un sistema. Estos reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, evitan más bien que los errores ocurran, lo que permite obtener un código de alta calidad.

Los estándares de codificación son un conjunto de pautas y normas que los desarrolladores siguen para escribir código de manera clara, eficiente y sostenible. Son cruciales para asegurar la calidad del código, facilitar su entendimiento y permitir un mantenimiento eficaz a lo largo del tiempo (Martin, 2021). Al adoptar estos estándares, se obtienen varios beneficios (Manuel, 2023):

- Promueven un estilo de codificación homogéneo, lo que facilita la lectura y comprensión del código entre diversos desarrolladores.
- Un código bien estructurado y documentado es más sencillo de modificar y ampliar, reduciendo los

costos y el tiempo de mantenimiento.

- Ayudan a prevenir errores comunes mediante buenas prácticas, como el uso de nombres descriptivos y la organización adecuada del código.
- Fomentan la colaboración en equipo, ya que todos los miembros pueden entender el código sin importar quién lo escribió.
- Promueven una documentación adecuada del código, útil para la capacitación de nuevos desarrolladores y la comprensión del sistema a lo largo del tiempo.

En la implementación de la solución se utilizó como base estándares de codificación en C establecidos por el equipo de proyecto en el desarrollo del sistema XAVIA PACS, entre ellos se identifican:

- El idioma utilizado debe ser español, y las palabras no se acentuarán.

Las líneas en blanco mejoran la legibilidad separando secciones de código lógicamente relacionadas. Siempre se deben usar dos líneas en blanco en las siguientes situaciones:

- Entre las secciones de un fichero fuente.
- Entre las definiciones de clases e interfaces.

Se debe usar siempre una línea en blanco en estas situaciones:

- Entre métodos.
- Entre las variables locales de un método y su primera sentencia.
- Antes de un comentario de bloque o de una sola línea.
- Entre las distintas secciones lógicas de un método para facilitar la lectura.

Se debe dejar un espacio en blanco en la siguiente situación:

- Entre una palabra clave del lenguaje y un paréntesis.

En cuanto a la inicialización, declaración y ubicación de variables, constantes, clases y métodos:

- Todas las instancias y variables de clases o métodos comenzarán con minúscula. Las palabras internas en nombres compuestos comenzarán con su primera letra en mayúscula. Los nombres de variables no deben comenzar con guiones bajos o signos de peso, aunque están permitidos por el lenguaje.

- Los nombres de las clases deben ser sustantivos; en nombres compuestos, cada palabra comenzará con mayúscula. Mantener los nombres de las clases simples y descriptivos, utilizando palabras completas y evitando acrónimos y abreviaturas.

Sobre la indentación y longitud de la línea:

- Usar cuatro espacios como unidad de indentación. Los tabuladores deben ser exactamente cada 8 espacios.
- Evitar líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

3.2. Tratamiento de excepciones

El tratamiento de excepciones es una técnica utilizada durante el tiempo de ejecución de un sistema para gestionar errores o fallos que pueden ocurrir en diversas rutinas. Estos fallos, conocidos como excepciones, son situaciones inesperadas que pueden interrumpir el flujo normal de ejecución. El objetivo principal del tratamiento de excepciones es restablecer el sistema a un estado en el que las rutinas puedan continuar su ejecución, garantizando así la robustez y fiabilidad del sistema.

En el contexto del módulo de Ortopedia desarrollado, no se ha implementado un manejo exhaustivo de excepciones. Esto se debe a que, dado el diseño y las funcionalidades del módulo, como la entrada de datos son coordenadas o posiciones que el usuario entra con el uso del mouse, existe una baja probabilidad de que el programa en ejecución se interrumpa abruptamente debido a un error, e independientemente de las acciones que se realicen.

Durante el tiempo de ejecución de un sistema, es posible que diversas rutinas experimenten fallos, comúnmente conocidos como excepciones. El tratamiento de excepciones tiene como objetivo restablecer el sistema a un estado en el que las rutinas puedan continuar su ejecución, asegurando así la robustez y fiabilidad del sistema.

En la herramienta desarrollada, no se ha implementado un control exhaustivo de las excepciones en todas las secciones de código, dado que las posibilidades de fallos en el módulo de Ortopedia son mínimas. Este módulo ha sido diseñado de manera que las operaciones sean robustas y se manejen adecuadamente en la mayoría de los casos sin necesidad de un manejo complejo de excepciones. Aunque el uso de bloques try y catch es una buena práctica para detectar y gestionar excepciones, y aunque mostrar mensajes de error puede facilitar la identificación y resolución de problemas, en este caso específico, la ausencia de estos mecanismos no compromete significativamente la estabilidad y fiabilidad del sistema. Donde único fue necesario aplicar

estas excepciones fue para que no se hiciese un trazo de línea invertido, que el programa no devolviese un ángulo negativo y que calculara el mismo ángulo cuando se trazaba la línea de izquierda a derecha y de derecha a izquierda, empleando condicionales y bloques try catch que verifican que lo mencionado anteriormente funcione adecuadamente.

3.3. Pruebas de Software

Las pruebas de software son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto. Cuando se realiza el proceso de pruebas el programa es ejecutado con datos de ejemplo cumpliendo con los tipos de datos de cada campo. Es necesario verificar los resultados de la prueba que se ejecuta para buscar errores, anomalías o información de atributos no funcionales del programa. El proceso de prueba tiene dos metas distintas: Demostrar al desarrollador y al cliente que el software cumple con los requerimientos y encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no sea el esperado según la descripción del requisito (Pressman, 2010).

Niveles de pruebas

Los niveles de prueba son etapas fundamentales en el proceso de pruebas de software, diseñadas para identificar y corregir errores en diferentes fases del desarrollo. Estos niveles incluyen las pruebas unitarias, del sistema y de aceptación. Las pruebas unitarias se enfocan en verificar que componentes individuales del software, como funciones o módulos, funcionen correctamente por sí solos. Las pruebas del sistema examinan el sistema completo para garantizar que cumpla con los requisitos especificados y funcione adecuadamente en un entorno real. Finalmente, las pruebas de aceptación son realizadas por el cliente o usuario final para confirmar que el sistema cumple con sus expectativas y requisitos antes de su implementación definitiva (Pressman, 2020).

3.4. Tipo de Prueba Funcional

Las pruebas funcionales son un tipo de evaluación en el desarrollo de software que se enfoca en verificar que las funcionalidades del sistema cumplan con los requisitos especificados por el usuario o cliente. Estas pruebas evalúan la entrada y salida del software, asegurando que cada función opere según lo esperado. Se incluyen actividades como la validación de interfaces de usuario, la prueba de APIs y la verificación de flujos de trabajo. Son cruciales para detectar y corregir defectos antes de que el software llegue al usuario final, garantizando así la calidad y satisfacción del cliente (Baeldung, 2023).

Pruebas unitarias

Las pruebas unitarias son cruciales en el desarrollo de software. Estas consisten en evaluar de manera aislada las unidades más pequeñas del código, como funciones o métodos individuales, para asegurar que operan correctamente. Probar cada unidad por separado permite detectar errores lógicos, bucles infinitos y posibles vulnerabilidades de seguridad desde las primeras fases del desarrollo. Al abordar estos problemas de forma temprana, se mejora significativamente la calidad del software, facilitando además el mantenimiento y la comprensión del código a largo plazo. Las pruebas unitarias también fomentan una programación más disciplinada y modular, lo que a su vez contribuye a un desarrollo más eficiente y ordenado(*Unit Testing Best Practices* 2024).

En cuanto a la técnica de prueba de caja blanca, este enfoque se centra en el análisis exhaustivo del código fuente y la lógica interna del programa. A diferencia de las pruebas de caja negra, esta es otra técnica de prueba donde los evaluadores no tienen acceso al código y solo se preocupan por las entradas y salidas, las pruebas de caja blanca permiten una inspección detallada de las estructuras internas. Esto incluye la evaluación de flujos de control, rutas lógicas y condiciones específicas del código, asegurando que cada parte del programa funcione correctamente bajo diferentes escenarios. Este enfoque no solo verifica que el software cumple con sus especificaciones, sino que también garantiza que el código esté optimizado y libre de defectos que podrían causar problemas en el futuro(*White Box Testing Techniques* 2024).

Las pruebas unitarias se llevaron a cabo con una metodología meticulosa y detallada, garantizando el correcto funcionamiento de cada componente del software. A través de la creación de casos de prueba completos y la ejecución de tres iteraciones, se lograron identificar y corregir errores antes del lanzamiento final. En la primera iteración, se alcanzó un nivel de satisfacción del 65 %. Durante la segunda iteración, este porcentaje mejoró a un 80 %, y en la tercera iteración, se alcanzó el 100 % de satisfacción. La utilización de la herramienta automatizada permitió una evaluación precisa y eficiente, lo que incrementó significativamente la confianza en la calidad del producto final. En resumen, el proceso de pruebas unitarias fue exitoso y fundamental para asegurar la robustez y fiabilidad del software desarrollado.

Como resultado de las pruebas unitarias se utilizó la librería NUnit en su versión 4.1.0, lanzada en febrero de 2024. Esta herramienta facilitó la implementación y ejecución de las pruebas unitarias, permitiendo la verificación automática de la funcionalidad de diversas unidades del código. Con NUnit, se lograron identificar y corregir errores lógicos, asegurar el correcto funcionamiento de métodos individuales, y garantizar la calidad del software desde las primeras etapas del desarrollo. Además, NUnit proporcionó un entorno robusto y flexible para escribir pruebas, estructurarlas y mantenerlas, lo que contribuyó significativamente a mejorar la eficiencia del proceso de desarrollo y la estabilidad del producto final.(Tudose, 2020). A continuación, se muestran algunos de los set de pruebas realizados.

```

[Test]
✓ | 0 references
public void CalcularAngulo_DeberiaRetornarAnguloCorrecto()
{
    // Arrange
    Point puntoMedio = new Point(0, 0);
    Point puntoPerpendicular = new Point(0, 10);
    Point puntoClic = new Point(10, 0);

    // Act
    double angulo = _anguloIncidenciaControl.CalcularAngulo(puntoMedio, puntoPerpendicular, puntoClic);

    // Assert
    Assert.That(angulo, Is.EqualTo(90).Within(0.1));
}

[Test]
✓ | 0 references
public void CalcularAngulo_PerpendicularInvertida_DeberiaRetornarAnguloCorrecto()
{
    // Arrange
    Point puntoMedio = new Point(0, 0);
    Point puntoPerpendicular = new Point(0, -10);
    Point puntoClic = new Point(10, 0);

    // Act
    double angulo = _anguloIncidenciaControl.CalcularAngulo(puntoMedio, puntoPerpendicular, puntoClic);

    // Assert
    Assert.That(angulo, Is.EqualTo(90).Within(0.1));
}

[Test]
✓ | 0 references
public void CalcularAngulo_EjemploAdicional_DeberiaRetornarAnguloCorrecto()
{
    // Arrange
    Point puntoMedio = new Point(0, 0);
    Point puntoPerpendicular = new Point(10, 0);
    Point puntoClic = new Point(10, 10);

    // Act
    double angulo = _anguloIncidenciaControl.CalcularAngulo(puntoMedio, puntoPerpendicular, puntoClic);

    // Assert
    Assert.That(angulo, Is.EqualTo(45).Within(0.1));
}

```

Figura 15. Set de pruebas realizadas al control de usuario que pertenece al parámetro espino pélvico de Incidencia Pélvica.

```

public class CobbControlTests
{
    private CobbControl _cobbControl;

    [SetUp]
    0 references
    public void Setup()
    {
        _cobbControl = new CobbControl();
    }

    [Test]
    0 references
    public void CalcularAngulo_DeberiaRetornarAnguloCorrecto()
    {
        // Arrange
        Point p1 = new Point(0, 0);
        Point p2 = new Point(0, 10);

        // Act
        float angulo = _cobbControl.CalcularAngulo(p1, p2);

        // Assert
        Assert.That(angulo, Is.EqualTo(90).Within(0.1)); // Valor esperado: 90 grados con margen de error
    }

    [Test]
    0 references
    public void CalcularAnguloInterseccion_DeberiaRetornarAnguloCorrecto()
    {
        // Arrange
        Point p1 = new Point(0, 0); // Punto inicial de la primera línea
        Point p2 = new Point(10, 0); // Punto final de la primera línea (horizontal)
        Point p3 = new Point(0, 0); // Punto inicial de la segunda línea (mismo que el punto inicial de la primera línea)
        Point p4 = new Point(0, 10); // Punto final de la segunda línea (vertical)

        // Act
        float anguloInterseccion = _cobbControl.CalcularAnguloInterseccion(p1, p2, p3, p4);

        // Assert
        Assert.That(anguloInterseccion, Is.EqualTo(90).Within(0.1)); // Ajusta según la lógica del cálculo
    }
}

```

Figura 16. Set de pruebas realizadas al control de usuario que pertenece al método de Cobb.

```
namespace ProyectoPruebas
{
    [TestFixture]
    [Apartment(ApartmentState.STA)]
    0 references
    public class InclinacionSacroControlTests
    {
        private InclinacionSacroControl _inclinacionSacroControl;

        [SetUp]
        0 references
        public void Setup()
        {
            _inclinacionSacroControl = new InclinacionSacroControl();
        }

        [Test]
        ✓ | 0 references
        public void LimpiarCanvas_DebeLimpiarElTextoDelViewModel()
        {
            // Arrange
            _inclinacionSacroControl.viewModel.AnguloTexto = "Texto de prueba";

            // Act
            _inclinacionSacroControl.LimpiarCanvas();

            // Assert
            Assert.That(_inclinacionSacroControl.viewModel.AnguloTexto, Is.EqualTo(string.Empty)); // Verifica que el texto se haya limpiado
        }
    }
}
```

Figura 17. Set de pruebas realizadas al control de usuario que pertenece al parámetro espino pélvico de Inclinación de Sacro.

```

[Test]
0 references
public void DibujarPuntoAzul_DebeAñadirPuntoAzulAlCanvas()
{
    // Arrange
    Point punto = new Point(5, 5);

    // Act
    _versionPelvicaControl.DibujarPuntoAzul(punto);

    // Assert
    var puntosAzules = _versionPelvicaControl.CanvasPrincipal.Children.OfType<Ellipse>().Where(e => ((SolidColorBrush)e.Fill).Color == Colors.Blue).Count();
    Assert.That(puntosAzules, Is.EqualTo(1)); // Verifica que se añadió un punto azul al Canvas
}

[Test]
0 references
public void DibujarPuntoVerde_DebeAñadirPuntoVerdeAlCanvas()
{
    // Arrange
    Point punto = new Point(5, 5);

    // Act
    _versionPelvicaControl.DibujarPuntoVerde(punto);

    // Assert
    var puntosVerdes = _versionPelvicaControl.CanvasPrincipal.Children.OfType<Ellipse>().Where(e => ((SolidColorBrush)e.Fill).Color == Colors.LightGreen).Count();
    Assert.That(puntosVerdes, Is.EqualTo(1)); // Verifica que se añadió un punto verde al Canvas
}

[Test]
0 references
public void DibujarLineasRojasYCalcularAngulo_DebeDibujarLineasYCalcularAnguloCorrectamente()
{
    // Arrange
    Point puntoClic = new Point(10, 10);
    _versionPelvicaControl.puntoMedio = new Point(20, 20);

    // Act
    _versionPelvicaControl.DibujarLineasRojasYCalcularAngulo(puntoClic);

    // Assert
    var lineasRojas = _versionPelvicaControl.CanvasPrincipal.Children.OfType<Line>().Where(l => l.Stroke == Brushes.Red).Count();
    Assert.That(lineasRojas, Is.EqualTo(2)); // Verifica que se dibujaron dos líneas rojas
    Assert.That(_versionPelvicaControl.viewModel.AnguloTexto, Does.Contain("Ángulo de versión pélvica:")); // Verifica que el ángulo se haya calculado y mostrado en el ViewModel
}

```

Figura 18. Set de pruebas realizadas al control de usuario que pertenece al parámetro espino pélvico de Versión Pélvica.

```

[Test]
✓ | 0 references
public void LimpiarCanvas_DebeLimpiarTodoCorrectamente()
{
    // Arrange
    _versionPelvicaControl.DibujarPuntoAzul(new Point(5, 5));
    _versionPelvicaControl.viewModel.AnguloTexto = "Texto de prueba";

    // Act
    _versionPelvicaControl.LimpiarCanvas(null, null);

    // Assert
    Assert.That(_versionPelvicaControl.CanvasPrincipal.Children.Count, Is.EqualTo(0)); // \
    Assert.That(_versionPelvicaControl.viewModel.AnguloTexto, Is.EqualTo(string.Empty)); // \
}

[Test]
✓ | 0 references
public void MouseLeftButtonDown_DebeIniciarYDetenerDibujo()
{
    // Arrange
    var eventoMouseDown = new MouseButtonEventArgs(Mouse.PrimaryDevice, 0, MouseButton.Left)
    {
        RoutedEvent = UIElement.MouseLeftButtonDownEvent,
        Source = _versionPelvicaControl
    };

    // Act
    _versionPelvicaControl.RaiseEvent(eventoMouseDown);
    _versionPelvicaControl.RaiseEvent(eventoMouseDown);

    // Assert
    Assert.That(_versionPelvicaControl.estaDibujando, Is.False);
    Assert.That(_versionPelvicaControl.primerPunto, Is.Null);
}

```

Figura 19. Set de pruebas restantes realizadas al control de usuario que pertenece al parámetro espino pélvico de Versión Pélvica.

Pruebas de sistema

Las pruebas de sistema son una fase crucial en el ciclo de desarrollo de software que se realiza para asegurar que el sistema completo cumple con los requisitos funcionales y no funcionales especificados. Estas pruebas abarcan la evaluación del sistema en su totalidad, integrando todos los módulos y componentes individuales para verificar su funcionamiento como un todo (Marin, 2020).

El método de prueba de caja negra son un enfoque de evaluación de software que se centra en la funcionalidad del sistema sin considerar su estructura interna o el código fuente. Este método se realiza desde

la perspectiva del usuario final, lo que significa que los evaluadores interactúan con el software tal como lo haría un usuario típico, ingresando datos y verificando las salidas generadas siendo es especialmente útil para identificar errores funcionales, problemas de interfaz y vulnerabilidades de seguridad(Spillner, 2021).

Una técnica fundamental en el método de pruebas de caja negra es la partición de equivalencia. Esta técnica implica dividir el conjunto de posibles entradas en clases o particiones que se consideran equivalentes, lo que significa que cualquier entrada dentro de una misma clase debería generar resultados similares. Al seleccionar únicamente algunos casos de prueba representativos de cada partición, se puede disminuir considerablemente la cantidad total de pruebas necesarias sin comprometer la cobertura(Spillner, 2021).

Se ejecutaron 4 casos de pruebas para los requisitos definidos. Para el primer caso de prueba, que corresponde al requisito Método de Cobbs, se diseñaron un total de 4 escenarios. Para el segundo caso de prueba, correspondiente al requisito Método de Inclinación de Sacro, se definieron 5 escenarios, para el tercer caso de prueba, correspondiente al requisito Método de Incidencia Pélvica, se definió un solo escenario y para cuarto caso de prueba, perteneciente al requisito Método de Versión Pélvica se diseñaron 2 escenarios.

Requisito Funcional 1: Método de Cobbs. En el caso de prueba ID: CF01, se verifica que el método de Cobbs calcula el ángulo correctamente al trazar dos líneas en el panel de imágenes. Las entradas son las coordenadas de las líneas trazadas en el panel, y el resultado esperado es que el ángulo calculado se muestre correctamente en la interfaz. En el primer escenario, el usuario traza una línea desde el punto (10,10) hasta (50,50) y otra línea desde (50,50) hasta (90,90), esperando que el ángulo calculado sea el valor esperado según los cálculos geométricos. En el segundo escenario, el usuario traza las líneas en secuencia inversa (de derecha a izquierda), y el ángulo calculado debe ser el mismo que en el primer escenario. En el tercer escenario, el usuario traza líneas con ángulos muy agudos, esperando que el ángulo calculado sea preciso y no negativo. En el cuarto escenario, el usuario traza líneas con ángulos muy obtusos, y el ángulo calculado debe reflejar correctamente el ángulo obtuso.

Requisito Funcional 2: Método de Inclinación de Sacro. En el caso de prueba ID: IS01, se verifica que el método de inclinación de sacro devuelve el ángulo correcto al trazar las líneas correspondientes en el panel. Las entradas son las coordenadas de las líneas trazadas en el panel, y el resultado esperado es que el ángulo se calcula y se muestra correctamente. En el primer escenario, el usuario traza una línea desde (20,20) hasta (70,70) y otra desde (70,70) hasta (120,120), esperando que el ángulo calculado se muestre correctamente en la interfaz. En el segundo escenario, el usuario traza líneas que forman un ángulo recto, esperando que el ángulo calculado sea de 90 grados. En el tercer escenario, el usuario traza líneas con diferentes longitudes, esperando que el ángulo calculado sea preciso independientemente de la longitud de las líneas. En el cuarto escenario, el usuario traza las líneas en diferentes áreas del panel, esperando que el ángulo calculado sea correcto en todas las áreas. En el quinto escenario, el usuario traza líneas con pequeños movimientos erráticos, esperando que el ángulo calculado sea estable y correcto a pesar de las imperfecciones en el trazo.

Requisito Funcional 3: Método de Incidencia Pélvica. En el caso de prueba ID: IP01, se verifica que el método de incidencia pélvica muestra el ángulo correcto al trazar las líneas en el panel. Las entradas son las coordenadas de las líneas trazadas en el panel, y el resultado esperado es que el ángulo se presenta correctamente en la interfaz. En el único escenario, el usuario traza una línea en la pelvis desde (30,30) hasta (80,80) y otra desde (80,80) hasta (130,130), esperando que el ángulo se calcule correctamente y se muestre en la interfaz.

Requisito Funcional 4: Método de Versión Pélvica. En el caso de prueba ID: VP01, se verifica que el método de versión pélvica muestra el ángulo correcto al trazar las líneas en el panel. Las entradas son las coordenadas de las líneas trazadas en el panel, y el resultado esperado es que el ángulo se presenta correctamente en la interfaz. En el primer escenario, el usuario traza una línea desde (40,40) hasta (90,90) y otra desde (90,90) hasta (140,140), esperando que el ángulo se presente correctamente en la interfaz. En el segundo escenario, el usuario introduce datos de entrada para otros ángulos pélvicos, esperando que la interfaz refleje el ángulo correcto de acuerdo con los datos ingresados.

Se realizan tres iteraciones de evaluación, y se identifican las siguientes áreas claves de defectos clasificados en:

- Errores de arquitectura
- Errores de funcionalidades
- Errores de interfaz
- Errores de visualización

Tabla 15. Resultados de la prueba de caja negra.

No. Iteración	Defectos detectados	Funcionalidad	Interfaz	Visualización	Arquitectura	Resueltas
1era Iteración	7	2	2	2	1	7
2da Iteración	4	1	1	1	1	4
3era Iteración	0	0	0	0	0	0

Se llevaron a cabo 3 iteraciones para eliminar todos los defectos identificados. Los resultados de estas pruebas se presentan en la tabla. Una vez solucionados los defectos en cada iteración, se realizaron pruebas de regresión. Con los casos de prueba diseñados, se verificó el 100 % de las funcionalidades desarrolladas para la herramienta de medición del grado de escoliosis en un visor de imágenes médicas en el sistema Xavia PACSViewer, corrigiendo todos los defectos encontrados. En la tercera iteración no se detectaron nuevos defectos, concluyendo así esta fase de pruebas. El proceso de pruebas realizado demuestra el cumplimiento de las exigencias establecidas en el levantamiento de requisitos.

No. Iteración	NC Detectadas	Descripción	Complejidad
1ra Iteración	1	Necesidad de cambiar a WPF (Prueba Arquitectura)	Alta
	2	Error de escalamiento de imagen (NC Funcional)	Media
	3	Forma del cursor incorrecta (NC de Visualización)	Baja
	4	Grosor y color de líneas no adecuados (NC de Visualización)	Baja
	5	Incorrecto funcionamiento de botones en WPF (NC Funcional)	Alta
	6	Desajustes en la ventana de ortopedia (NC de Interfaz)	Media
	7	Mensajes de error no descriptivos (NC de Interfaz)	Baja
2da Iteración	8	Falta de instrucciones claras en la interfaz (Prueba de Interfaz)	Media
	9	Problemas con la compatibilidad de versiones (NC Arquitectura)	Alta
	10	Tiempos de respuesta lentos al trazar líneas (NC Funcional)	Media
	11	Visualización incorrecta en diferentes resoluciones (NC de Visualización)	Media
3ra Iteración			

Tabla 14. Listado de no conformidades detectadas por iteración

Pruebas de regresión

Las pruebas de regresión son un tipo de evaluación de software que se realizan para garantizar que las modificaciones o actualizaciones en el código no introduzcan nuevos errores ni afecten negativamente a las funcionalidades ya existentes. Este tipo de pruebas es crucial tras efectuar cambios en el software, como la incorporación de nuevas características, la corrección de errores o las mejoras de rendimiento, asegurando que el sistema continúe funcionando según lo esperado. El proceso de pruebas de regresión generalmente implica la ejecución de un conjunto de casos de prueba predefinidos que abarcan las funcionalidades del software. Estos casos de prueba pueden ser ejecutados manualmente o, con mayor frecuencia, mediante herramientas automatizadas. La automatización es especialmente útil en las pruebas de regresión, ya que permite realizar pruebas repetitivas con mayor rapidez y precisión, facilitando así la detección temprana de defectos (Patil, 2020).

Durante el desarrollo de la nueva funcionalidad de Ortopedia para el sistema XAVIA PACSViewer, se llevaron a cabo pruebas de regresión en cada iteración del proceso, siguiendo una estrategia de pruebas de caja negra. Estas pruebas permitieron verificar que todas las no conformidades identificadas se habían corregido, asegurando que las soluciones implementadas no afectaran negativamente a otras funcionalidades del sistema. Además, se realizaron pruebas de caja blanca para evaluar la lógica interna del código y asegurar que todos los requisitos se cumplieran correctamente. Esta metodología integral de pruebas contribuyó a garantizar la estabilidad y calidad del sistema en su conjunto.

Principales No Conformidades

Se hallaron un grupo de no conformidades durante el proceso de construcción y ejecución de la aplicación:

- No conformidad de tipo unitaria: Se modificó una unidad del código agregando más condicionales en cada uno de los métodos de medición debido a que en un principio si se trazaba una línea de derecha a izquierda el método calculaba mal el ángulo y daba un valor negativo, sin embargo si se trazaba la línea de izquierda a derecha, tanto la primera como la segunda, entonces el método calculaba el ángulo correctamente. Luego esta no conformidad fue corregida por el desarrollador y validada por el tutor.
- No conformidad de tipo arquitectura: Se tuvo que modificar el tipo de proyecto en C# a WPF lo que se tenía anteriormente en Windows forms, esto es un error en la arquitectura en la que se desarrolló el componente, la primera no era la correcta, se validó por el arquitecto, se corrigió y se cambió a WPF. Es una no conformidad alta. Fue validada por el arquitecto y posteriormente se le dió solución.
- No conformidad de tipo funcionalidad: Hubo un error con el escalamiento de la imagen que la validó el especialista cuando se le mostró la aplicación. Al estar la foto de la interfaz de la propiedad stretch en fill, cuando se realizaban las mediciones en el panel, no eran correctas, pero no porque el código de

las mediciones estuviese mal, sino porque la imagen tenía la propiedad stretch mal, luego se cambió stretch a none y se configuró para que el panel donde se hace el trazado de las líneas tuviese las mismas dimensiones que la de la foto.

- No conformidad de tipo visualización: Se tuvo que cambiar la forma del cursor original al de una cruz mientras el cursor se movía por el panel. Fue un error que se corrigió y se validó por el especialista.
- No conformidad de tipo visualización: Se cambió el grosor y el color de las líneas, debido a que en un principio el grosor y el color no eran los idóneos para el visor. Esta es una no conformidad que se validó por la tutora y más tarde se corrigió.

3.5. Pruebas No Funcionales

Las pruebas no funcionales son un tipo de evaluación que se realizaron para medir aspectos del software que no están relacionados directamente con funciones específicas de la aplicación, sino con cómo funciona el sistema en su conjunto. Estas pruebas se enfocaron en atributos como el rendimiento, la escalabilidad, la seguridad, la usabilidad y la fiabilidad del sistema. Consistieron en realizar una serie de pruebas que examinaron cómo el software respondía bajo diferentes condiciones de carga, su capacidad para manejar múltiples usuarios simultáneamente, la resistencia a ataques de seguridad y la facilidad de uso para los usuarios finales. Durante estas pruebas, se recopilaban datos sobre tiempos de respuesta, uso de recursos, comportamiento bajo estrés y vulnerabilidades potenciales. El objetivo fue asegurar que el sistema no solo cumpliera con sus funciones, sino que también proporcionara un rendimiento robusto y confiable en todas las condiciones operativas esperadas.(Baeldung, 2023).

Pruebas de rendimiento

Las pruebas de rendimiento son un conjunto de evaluaciones destinadas a determinar cómo una aplicación se comporta bajo ciertas cargas y condiciones. Consisten en medir el tiempo de respuesta, la capacidad de procesamiento, el uso de recursos del sistema como la CPU y la memoria, y la estabilidad general de la aplicación. Estas pruebas ayudan a identificar cuellos de botella, optimizar el rendimiento y asegurar que la aplicación funcione eficientemente cuando se enfrenta a distintas situaciones.

El objetivo fundamental es definir el comportamiento estándar del sistema. No se busca identificar errores en la aplicación, sino que se enfoca en aspectos de medición, como los tiempos de respuesta y el rendimiento. Esto puede llevarse a cabo de diversas maneras, dependiendo de la herramienta utilizada.(Meir, 2017).

Tabla 16. Características del entorno de prueba.

Características del entorno de prueba	Especificaciones
Memoria RAM	8 GB
Capacidad de disco	500 GB
Procesador	i7 3687U

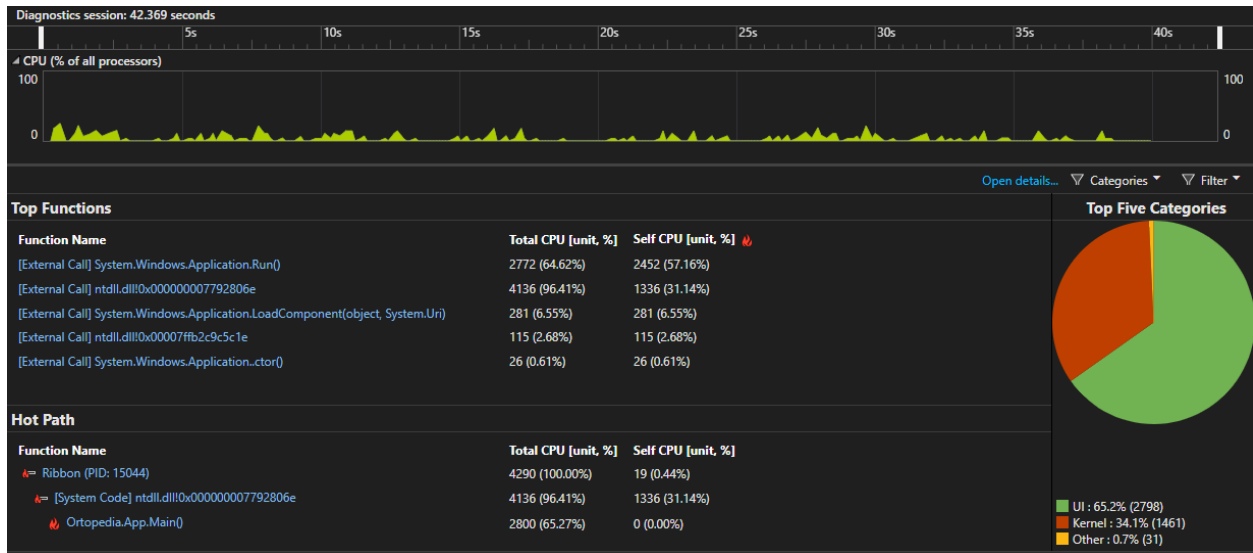


Figura 20. Resultados de pruebas basadas en tiempo de CPU

En el caso de la aplicación Ortopedia, se realizaron pruebas de rendimiento en el propio Visual Studio. Las evaluaciones mostraron que la aplicación, desde su inicio, presentó un rendimiento óptimo debido a su naturaleza sencilla y al bajo consumo de la CPU. Los picos en la gráfica de uso de CPU fueron bajos y no sobrepasaron el 25 %, indicando que la carga de trabajo de la aplicación es manejada de manera eficiente sin causar estrés significativo al sistema. Este comportamiento refleja la eficiencia del diseño y la implementación de la aplicación, donde las operaciones clave como la carga de imágenes, el trazado de líneas y el cálculo de ángulos se ejecutaron sin problemas. La ausencia de cuellos de botella y el uso moderado de recursos respaldan la conclusión de que la aplicación cumple con los estándares de rendimiento esperados, asegurando una experiencia de usuario fluida y sin interrupciones.

3.6. Listas de verificación

Las listas de verificación son herramientas utilizadas para organizar y verificar tareas de manera eficiente y efectiva. Estas listas permiten controlar el cumplimiento de una serie de requisitos o recopilar datos de

forma ordenada y sistemática. Se utilizan en una variedad de campos, desde la inspección de edificios hasta cirugías médicas complejas, asegurando que no se olvide ningún paso importante. Las listas de verificación ayudan a reducir errores y garantizar la coherencia e integridad en el cumplimiento de procesos (DataScope, 2021).

3.6.1. Listas de verificación basadas en Usabilidad

Las listas de verificación basadas en usabilidad son herramientas específicas diseñadas para evaluar la facilidad de uso de un producto o sistema. Estas listas se utilizan comúnmente en pruebas de usabilidad para identificar problemas y mejorar la experiencia del usuario. Incluyen criterios y módulos que permiten a los evaluadores verificar aspectos como la navegación, la claridad de la interfaz y la accesibilidad. Al utilizar estas listas, las organizaciones pueden asegurarse de que sus productos sean intuitivos y fáciles de usar para los usuarios finales (Hotjar, 2023).

Tabla 17. Lista de Chequeo de Usabilidad

Nº	Pregunta de Usabilidad	Respuesta
1	¿Es la interfaz del usuario clara y fácil de entender?	Sí, la interfaz está diseñada para ser intuitiva y accesible para el especialista, facilitando la comprensión.
2	¿Los botones y enlaces están claramente etiquetados?	Sí, todos los botones y enlaces están etiquetados claramente para guiar al especialista en sus tareas.
3	¿El sistema proporciona retroalimentación inmediata y apropiada al usuario?	Sí, el sistema ofrece retroalimentación en tiempo real para las acciones realizadas, mejorando la experiencia del especialista.
4	¿Las instrucciones son fáciles de seguir y comprender?	Sí, las instrucciones son claras y concisas, facilitando su seguimiento por parte del especialista.
5	¿El diseño de la interfaz es consistente a lo largo de todo el sistema?	Sí, se mantiene una consistencia en el diseño de la interfaz para asegurar una experiencia uniforme.
6	¿Se puede acceder a todas las funciones principales fácilmente desde la interfaz de inicio?	Sí, todas las funciones principales son accesibles directamente desde la página de inicio.
7	¿El sistema permite deshacer fácilmente las acciones del usuario?	Sí, el sistema cuenta con una opción para eliminar las mediciones.
8	¿Los mensajes de error son claros y útiles para el usuario?	No, actualmente no hay mensajes de error.
9	¿El sistema proporciona ayuda y documentación accesible?	No, no hay documentación y ayuda accesible actualmente.
10	¿El tiempo de respuesta del sistema es aceptable?	Sí, el tiempo de respuesta es adecuado y permite realizar tareas de manera eficiente.

Nº	Pregunta de Usabilidad	Respuesta
11	¿El sistema es accesible para usuarios con discapacidades?	No se ha validado específicamente la accesibilidad para usuarios con discapacidades, dado que el único usuario es un especialista sin estas necesidades.
12	¿La navegación es intuitiva y permite al usuario encontrar lo que necesita fácilmente?	Sí, la navegación es intuitiva y facilita al especialista encontrar la información que necesita.
13	¿Las funciones avanzadas están ocultas para no confundir a los usuarios novatos?	No, todas las funciones están disponibles para el especialista, quien está familiarizado con el uso del sistema.
14	¿El sistema evita el uso excesivo de jergas técnicas?	Sí, se evita el uso de jergas técnicas para asegurar la comprensión del especialista.
15	¿El sistema es consistente con las expectativas del usuario?	Sí, el sistema cumple con las expectativas generales del especialista en el módulo de Ortopedia.
16	¿Los iconos utilizados son reconocibles y consistentes?	Sí, los iconos son claros y uniformes a lo largo del sistema.
17	¿El diseño visual del sistema es atractivo y no distrae al usuario?	Sí, el diseño visual es atractivo y mantiene al especialista enfocado en sus tareas.
18	¿El texto es legible y de tamaño adecuado?	Sí, el texto es fácilmente legible y de un tamaño adecuado.
19	¿El sistema permite personalización por parte del usuario?	No, el sistema no permite actualmente la personalización por parte del especialista.
20	¿Las tareas comunes se pueden completar en pocos pasos?	Sí, las tareas comunes están optimizadas para completarse en el menor número de pasos posible.
21	¿El sistema proporciona atajos de teclado para usuarios avanzados?	No, el sistema no ofrece actualmente atajos de teclado para el especialista.
22	¿El sistema es capaz de manejar entradas incorrectas o inesperadas de manera elegante?	Sí, el sistema maneja las entradas incorrectas o inesperadas de manera adecuada.
23	¿El flujo de trabajo del usuario es claro y lógico?	Sí, el flujo de trabajo es claro y sigue una lógica intuitiva.
24	¿Los formularios son fáciles de llenar y no requieren información innecesaria?	Sí, los formularios están diseñados para ser fáciles de llenar y solicitan solo la información necesaria.
25	¿El sistema recuerda las preferencias del usuario y las configura automáticamente?	No, el sistema no guarda preferencias del usuario para configuraciones automáticas.
26	¿El sistema ofrece opciones para la recuperación de contraseñas o cuentas?	N/A, no aplica ya que el módulo de Ortopedia no maneja cuentas de usuario.
27	¿El sistema notifica al usuario sobre actualizaciones importantes?	No, actualmente no se implementan notificaciones de actualizaciones.
28	¿El sistema permite al usuario guardar su progreso y continuar más tarde?	No, el sistema no cuenta con esta funcionalidad ya que no utiliza una base de datos.
29	¿El sistema incluye mecanismos de validación de datos en tiempo real?	Sí, se han implementado mecanismos de validación de datos en tiempo real.

Nº	Pregunta de Usabilidad	Respuesta
30	¿El usuario puede acceder fácilmente a su historial de acciones?	No, actualmente no hay una función para acceder al historial de acciones del usuario.

El módulo de Ortopedia cumplió con diecinueve de treinta criterios de usabilidad. La interfaz es clara, intuitiva y consistente, facilitando la navegación y la ejecución de tareas por parte del especialista. Sin embargo, no ofrece mensajes de error claros, documentación accesible, ni opciones de personalización o atajos de teclado. La aplicación se usa exclusivamente en sistemas Windows como una aplicación de escritorio, cumpliendo con las expectativas actuales, pero tiene áreas clave que necesitan mejoras para optimizar la usabilidad general.

3.6.2. Listas de verificación basadas en Portabilidad

Las listas de verificación basadas en portabilidad se utilizan para evaluar la capacidad de un sistema o software para ser transferido y operado en diferentes entornos o plataformas. Estas listas incluyen criterios que verifican aspectos como la compatibilidad, la facilidad de instalación y la adaptabilidad del sistema a diferentes configuraciones tecnológicas. Son esenciales para garantizar que un producto funcione correctamente en diversos entornos sin necesidad de modificaciones significativas (SlideTeam, 2023).

Tabla 18. Lista de Chequeo - Portabilidad

ITEM	REQUISITOS	SI	NO	OBSERVACIONES
ADAPTABILIDAD				
1	Se debe verificar si una aplicación determinada puede funcionar correctamente en todos los entornos de destino previstos (hardware, software, sistema operativo, etc.).	x		El sistema ha sido probado en diferentes entornos Windows.
2	Al cambiar de navegador la interfaz del mismo debe seguir siendo la misma.		x	No aplica, la aplicación es de escritorio, no basada en navegador.
3	El sistema debe ser capaz de ser adaptado en diversos ambientes, bajo distintas condiciones y modificando ciertas variables de su entorno.	x		El sistema ha demostrado ser adaptable en varias configuraciones de hardware y software.
INSTALABILIDAD				

ITEM	REQUISITOS	SI	NO	OBSERVACIONES
4	Se debe validar que el software pueda instalarse correctamente siguiendo las instrucciones de un manual de instalación (incluida la ejecución de cualquier script de instalación) o utilizando un asistente de instalación. Esto incluye ejecutar las opciones de instalación para diferentes configuraciones de hardware / software y para diversos grados de instalación (por ejemplo, inicial o actualización).	x		La instalación se realizó correctamente siguiendo las instrucciones proporcionadas.
5	Se debe comprobar si el software de instalación resuelve correctamente las fallas que ocurren durante la instalación sin dejar el sistema en un estado indefinido.	x		Se ha comprobado que el instalador maneja correctamente las fallas.
6	Se debe comprobar si se puede completar una instalación / desinstalación parcial.	x		La instalación/desinstalación parcial se completó sin problemas.
7	El proceso de instalación debe completarse en un número específico de minutos o en menos de un número específico de pasos.	x		La instalación se completa en un tiempo razonable y en pocos pasos.
8	Se debe realizar una prueba de la aplicación en varias versiones de navegador.		x	No aplica, la aplicación es de escritorio, no basada en navegador.
	Se debe realizar la prueba de la aplicación en diferentes dispositivos (Apple, iOS, Google, y Android).		x	No aplica, la aplicación es de escritorio y solo se usa en Windows.
9	Se debe realizar la prueba de la aplicación en varios sistemas operativos.		x	La aplicación solo está diseñada para funcionar en Windows.
REEMPLAZABILIDAD				
10	El componente que reemplaza al componente anterior debe producir los mismos resultados que el componente anterior en todos los entornos de destino.	x		Los componentes actualizados producen los mismos resultados que los anteriores.

ITEM	REQUISITOS	SI	NO	OBSERVACIONES
11	El componente anterior debe tener el mismo propósito que el componente que se reemplaza.	x		Los componentes de reemplazo tienen los mismos propósitos que los originales.

En el proceso de revisión de la portabilidad del módulo de Ortopedia, se evaluaron varios criterios de adaptabilidad, instalabilidad y reemplazabilidad. De un total de once criterios evaluados, se determinó que siete fueron cumplidos y cuatro no se cumplieron. La aplicación en cuestión está diseñada exclusivamente para ser utilizada en sistemas operativos Windows y es una aplicación de escritorio.

En cuanto a la adaptabilidad, se verificó que la aplicación funciona correctamente en los entornos de destino previstos, incluyendo diversas configuraciones de hardware y software. Sin embargo, no se aplicaron verificaciones relacionadas con la compatibilidad en navegadores web, dado que la aplicación no está basada en navegadores.

Respecto a la instalabilidad, se validó que el software puede instalarse correctamente siguiendo las instrucciones proporcionadas, manejando adecuadamente las fallas durante el proceso de instalación y completando tanto la instalación como la desinstalación parcial sin problemas. La instalación se realiza en un tiempo razonable y en pocos pasos. No obstante, no se realizaron pruebas en navegadores web ni en diferentes dispositivos móviles, ya que no son relevantes para este tipo de aplicación de escritorio.

En términos de reemplazabilidad, se confirmó que los componentes actualizados producen los mismos resultados que los anteriores y tienen el mismo propósito en todos los entornos de destino.

En conclusión, el módulo de Ortopedia muestra un buen desempeño en la mayoría de los criterios evaluados, demostrando estabilidad y eficiencia en su contexto de uso previsto. A pesar de no cumplir con algunos criterios relacionados con la compatibilidad en navegadores web y dispositivos móviles, cumple satisfactoriamente con los requisitos esenciales para su entorno de ejecución actual, proporcionando una experiencia consistente y fiable para el especialista que utiliza la aplicación.

Conclusiones parciales

La incorporación de buenas prácticas en la implementación de la solución permitió obtener un código claro, limpio y documentado la reutilización de los componentes. Los resultados de las pruebas aplicadas demostraron un producto de calidad cuya respuesta es la deseada y esperada por el usuario, ajustándose a los requerimientos definidos para el sistema.

Conclusiones generales

- El análisis de los métodos para el diagnóstico de la escoliosis y el balance sagital permitió identificar las mediciones angulares que se realizan en cada uno.
- La caracterización de los sistemas homólogos confirmó el empleo del método de Cobb para el diagnóstico de la escoliosis y desechando el método de Ferguson para este propósito.
- El análisis de las tecnologías para el desarrollo de software permitió elaborar la documentación de las nuevas funcionalidades desarrolladas.
- Las pruebas de calidad aplicadas propició la obtención de un producto de calidad que facilita el diagnóstico de la escoliosis por el cálculo de la amplitud angular mediante el método de Cobb y el balance sagital mediante la medición de la amplitud angular de los parámetros espino-pélvicos.

Recomendaciones

Para futuras investigaciones se recomienda:

- Continuar utilizando y actualizando los estándares de codificación para mantener la calidad y mantenibilidad del código a lo largo del ciclo de vida del proyecto.
- Continuar la investigación, permitiendo incluir otras mediciones para facilitar el diagnóstico de otras enfermedades y padecimientos en la rama de ortopedia y así incrementar las funcionalidades del módulo.
- Incluir opciones para permitir la personalización del sistema por parte del usuario, así como la inclusión de atajos de teclado para aumentar la eficiencia en el uso del módulo.
- Considerar la migración del desarrollo del sistema a tecnologías multiplataformas, aumentando su portabilidad y accesibilidad.
- Evaluar regularmente las necesidades y expectativas del usuario final, asegurando que el sistema evolucione de acuerdo con sus requerimientos y siga proporcionando valor añadido en el ámbito de la ortopedia.

- [1] Gacitúa M, V., González C, M., Sanz, C., Mulli, V., Goddard, P., Rolón D, E., Castello, P., Hiebra, M. D. C., & Pappolla, R. (2016). Adolescent idiopathic scoliosis. *Archivos Argentinos de Pediatría*, 114(6), 585-594. Disponible en: <https://doi.org/10.5546/aap.2016.585>
- [2] García Fontecha, C. Dolor de espalda. *Pediatría Integral*, 2014, 18(7):413-424. Disponible en: <https://www.researchgate.net/publication/299369530>
- [3] Lisintuña, J. Prevalencia de escoliosis de columna lumbar en pacientes de 45 a 70 años con rayos x digital en la clínica cruz blanca de la ciudad de Quito período del 2014. Ecuador: Repositorio de la Universidad Central del Ecuador, 2015.
- [4] Jiménez Ávila, José María. (2011). Diagnóstico de la escoliosis. *Orthotips*, 7(2):83-88. Disponible en: <http://www.medigraphic.org.mx/>
- [5] Rodríguez, L., Pérez, J., & García, M. (2023). Incidencia y causas desconocidas de la escoliosis en poblaciones pediátricas". *Revista de Investigaciones Médicas*, 25(3), 123-135.
- [6] González, A. (2022). "Evaluación de tratamientos ortopédicos en la escoliosis: Limitaciones y avances". *Journal of Orthopedic Research*, 34(2), 89-101.
- [7] Troelsen, A., Japikse, P. (2017). *Pro C# 7: With .NET and .NET Core*. Apress.
- [8] Smith, J., Harrington, P. (2012). *MVVM in WPF: Unleashing the Model-View-ViewModel Pattern in Windows Presentation Foundation*. CreateSpace Independent Publishing Platform.
- [9] Wajchenberg, M., Astur, N., Kanas, M., Martins, D.E., Ferreira, R. (2016). "Adolescent idiopathic scoliosis: current concepts on neurological and muscular etiologies." *Scoliosis and Spinal Disorders*, 11, 4.
- [10] Timms, R. (2019). *Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers*. Packt Publishing.
- [11] Suh, C.H., et al. (2018). "Machine Learning and Radiomics: The Future of Medical Imaging." *American Journal of Roentgenology*, 211(2), 292-299.

- [12] Sommerville, I. (2015). *Software Engineering*. Pearson.
- [13] Pressman, R.S., Maxim, B.R. (2020). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
- [14] Kuroki, H., Inomata, N., Iwata, A., Nagai, T., Ito, H. (2012). "Development of a 3D measurement system for scoliosis diagnosis." *Medical & Biological Engineering & Computing*, 50(8), 815-824.
- [15] Negrini, S., Aulisa, A.G., Aulisa, L., Circo, A.B., de Mauroy, J.C., Durmala, J., Grivas, T.B., Knott, P., Kotwicki, T., Maruyama, T., O'Brien, J.P., Price, N., Parent, E., Rigo, M., Romano, M., Schraub, S., Stikeleather, L., Wynne, J., Zaina, F. (2012). "2011 SOSORT guidelines: Orthopaedic and Rehabilitation treatment of idiopathic scoliosis during growth." *Scoliosis*, 7, 3.
- [16] Lim, K.B., Nam, K.S., Lee, H.J., Lee, J.H. (2018). "Three-dimensional correction for scoliosis using the new non-fusion treatment system." *European Spine Journal*, 27(8), 1946-1952.
- [17] Cheung, J.P.Y., Lee, K.H., Tam, V., Yeung, K.H., Ng, L.W., Cheung, K.M.C., Luk, K.D.K. (2017). "The relation of 3D radiological parameters to curve severity and brace treatment in adolescent idiopathic scoliosis." *European Spine Journal*, 26(6), 1576-1583.
- [18] Aota Y, Saito T, Uesugi M, Kato S, Kuniya H, Koh R. Optimal arm position for evaluation of spinal sagittal balance. *J Spinal Disord Tech*. 2011;24:105?9. doi: 10.1097/BSD.0b013e3181da36c4. [DOI] [PubMed] [GoogleScholar]
- [19] Chairholder, Gary L. Horowitz. *CLSI C28-A3: Defining, Establishing, and Verifying Reference Intervals in the Clinical Laboratory; Approved Guideline, Third Edition*. 2008 [GoogleScholar]
- [20] Gelb DE, Lenke LG, Bridwell KH, Blanke K, McEnery KW. An analysis of sagittal alignment in 100 asymptomatic middle and older aged volunteers. *Spine*. 1995;20:1351?8. [PubMed] [GoogleScholar]
- [21] Jackson RP, McManus AC. Radiographic analysis of sagittal plane alignment and balance in standing volunteers and patients with low back pain matched for age, sex, and size: A prospective controlled clinical study. *Spine*. 1994;19:1611?8. doi: 10.1097/00007632-199407001-00010. [DOI] [PubMed] [GoogleScholar]
- [22] Legaye J. Influence of age and sagittal balance of the spine on the value of the pelvic incidence. *Eur Spine J*. 2014;23:1394?9. doi: 10.1007/s00586-014-3207-0. [DOI] [PubMed] [GoogleScholar]
- [23] Legaye J, Duval-Beaupere G, Hecquet J, Marty C. Pelvic incidence: A fundamental pelvic parameter for three-dimensional regulation of spinal sagittal curves. *Eur Spine J*. 1998;7:99?103. doi: 10.1007/s005860050038. [DOI] [PMCfreearticle] [PubMed] [GoogleScholar]

- [24] Le Huec JC, Roussouly P. Sagittal spino-pelvic balance is a crucial analysis for normal and degenerative spine. *Eur Spine J*. 2011;20(Suppl 5):S556?7. doi: 10.1007/s00586-011-1943-y. [DOI] [PMCfreearticle] [PubMed] [GoogleScholar]
- [25] Mac-Thiong JM, Berthonnaud E, Dimar J, Betz R, Labelle H. Sagittal alignment of the spine and pelvis during growth. *Spine*. 2004;29:1642?7. doi: 10.1097/01.brs.0000132312.78469.7b. [DOI] [PubMed] [GoogleScholar]
- [26] Protopsaltis T, Bronsard N, Soroceanu A, Henry J, Lafage R, Smith J, et al. Cervical sagittal deformity develops after PJK in adult thoracolumbar deformity correction: Radiographics analysis utilizing a novel global sagittal angular parameter, the CTPA. *Eur Spine J*. 2017;26:1111?20. doi: 10.1007/s00586-016-4653-7. [DOI] [PubMed] [GoogleScholar]
- [27] Roussouly P, Gollogly S, Berthonnaud E, Dimnet J. Classification of the normal variation in the sagittal alignment of the human lumbar spine and pelvis in the standing position. *Spine*. 2005;30:346?53. doi: 10.1097/01.brs.0000152379.54463.65. [DOI] [PubMed] [GoogleScholar]
- [28] Ryan D, Protopsaltis T, Ames C, Hostin R, Klineberg E, Mundis G, et al. T1 Pelvic Angle (TPA) effectively evaluates sagittal deformity and assesses radiographical surgical outcomes longitudinally. *Spine*. 2014;39:1203?10. doi: 10.1097/BRS.0000000000000382. [DOI] [PubMed] [GoogleScholar]
- [29] Schwab F, Blondel B, Bess S, Hostin R, Shaffrey C, Smith J, et al. Radiographical spinopelvic parameters and disability in the setting of adult spinal deformity. *Spine*. 2013;38:E803?12. doi: 10.1097/BRS.0b013e318292b7b9. [DOI] [PubMed] [GoogleScholar]
- [30] Schwab F, Lafage V, Boyce R, Skalli W, Farcy JP. Gravity line analysis in adult volunteers. Age-related correlation with spinal parameters, pelvic parameters and foot position. *Spine*. 2006;31:959?67. doi: 10.1097/01.brs.0000248126.96737.0f. [DOI] [PubMed] [GoogleScholar]
- [31] Vaz G, Roussouly P, Berthonnaud E, Dimnet J. Sagittal morphology and equilibrium of pelvis and spine. *Eur Spine J*. 2002;11:80?7. doi: 10.1007/s005860000224. [DOI] [PMCfreearticle] [PubMed] [GoogleScholar]
- [32] Vedantam R, Lenke LG, Keeney JA, Bridwell KH. Comparison of standing sagittal spinal alignment in asymptomatic adolescents and adults. *Spine*. 1998;23:211?5. doi: 10.1097/00007632-199801150-00012. [DOI] [PubMed] [GoogleScholar]

Referencias bibliográficas

AUP Metodología, 2019 (vid. pág. 14).

BAELDUNG, J., 2023. Functional Testing in Software Development. *Journal of Software Testing* (vid. págs. 45, 56).

BOULAY, C. et al., 2006. Influencia de la Incidencia Pélvica en la Alineación Espinal. *Spinal Health Journal* (vid. págs. 9, 11).

BROWN, T. et al., 2023. Método de Ferguson en el diagnóstico de escoliosis. *Journal of Spinal Disorders* (vid. pág. 6).

Business Process Model and Notation, 2024 (vid. pág. 15).

CHEUNG, J. et al., 2016. Fiabilidad de la medición digital del ángulo de escoliosis. *Digital Orthopedics Journal* (vid. pág. 7).

COBB, J. R., 1948. *Fundamentos de la Medicina del Ángulo de Cobb*. Publicaciones Médicas (vid. pág. 7).

DAVIS, K. et al., 2019. Alineación mecánica de la columna vertebral en relación con la pelvis. *Spinal Mechanics Journal* (vid. pág. 1).

DIEZ, E., 2003. *Estándares de Codificación en C#*. Tech Publications.

GABRIEL, M. et al., 2020. Innovative Approaches in Software Development. *Journal of Software Engineering* (vid. págs. 14, 17).

HRESKO, M. T., 2013. Definición y clasificación de la escoliosis. *Journal of Scoliosis Research* (vid. pág. 7).

ISO/IEC/IEEE 42010:2011 - Systems and Software Engineering. Architecture Description, 2011 (vid. pág. 17).

JONES, R. et al., 2022. Métodos de medición de la escoliosis. *Orthopedic Measurement Journal* (vid. pág. 6).

LEGAYE, J. et al., 1998. Balance Sagital y Parámetros Espino-Pélvicos. *Spinal Health Journal* (vid. págs. 9, 10).

LISINTUNA, M., 2015. Validación del método de Cobb en la escoliosis. *Journal of Spinal Disorders* (vid. pág. 6).

- LOENING, Andy, 2022. *AMIDE: a Medical Image Data Analysis Tool* (vid. pág. 12).
- LONSTEIN, J. E. et al., 1994. Estudios de escoliosis a largo plazo. *Long-term Scoliosis Studies Journal* (vid. pág. 7).
- LOPEZ, P., 2023. Decisiones clínicas personalizadas en ortopedia. *Clinical Decisions in Orthopedics* (vid. pág. 2).
- MANUEL, J., 2023. *Best Practices in Software Development*. Software Engineering Institute (vid. pág. 42).
- MARIN, J., 2020. *System Testing in Software Development*. Tech Publications (vid. pág. 51).
- MARTÁNEZ, J. et al., 2020. Escoliosis: Desviación lateral de la columna vertebral. *Journal of Orthopedic Research* (vid. pág. 5).
- MARTÁNEZ, J. et al., 2022. Mejoras en la Medicina de Escoliosis. *Journal of Orthopedic Research* (vid. pág. 2).
- MARTIN, R. C., 2021. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall (vid. pág. 42).
- MEIR, A., 2017. *Software Performance and Testing*. Tech Press (vid. pág. 56).
- MICRODICOM, 2022. *MicroDicom: A Free DICOM Viewer* (vid. pág. 13).
- MILLER, A. et al., 2018. Parámetros espino-pelvicos en el diagnóstico ortopédico. *Journal of Orthopedic Analysis* (vid. págs. 1, 2).
- NEGRINI, S. et al., 2012. Clasificación de la severidad de la escoliosis. *Spinal Research Journal* (vid. pág. 7).
- Object Management Group: *UML 3.0*, 2024 (vid. pág. 16).
- PRESSMAN, R. S., 2010. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education (vid. pág. 45).
- PRESSMAN, R. S., 2020. *Software Engineering: A Practitioners Approach*. McGraw-Hill Education (vid. pág. 45).
- RIGO, M. et al., 2010. Aplicación del Teorema de Tales en la Medicina de Escoliosis. *Spinal Health Journal* (vid. pág. 8).
- RODRÁGUEZ, J. A. C., 2020. CMMI-DEV v1.3 Standards for Software Development. *Journal of Advanced Computing* (vid. pág. 15).
- ROMERO ECHEGARAY, W. R.; BOBADILLA MINAYA, D. E.; CASTRO ROJAS, M. C.; PAREDES CAMPOS, F. J. y ZUÑIGA OSORIO, J. R., 2020. Escoliosis en adolescentes con la valoración radiográfica mediante el método de Ferguson. *Universidad Nacional Federico Villarreal* (vid. pág. 8).
- RONG, R. et al., 2019. Precisión en la medicina de la columna de escoliosis. *Orthopedic Precision Journal* (vid. pág. 7).

- SMITH, A. et al., 2021. *Diagnóstico de escoliosis mediante evaluaciones clínicas e imagen..* Medical Imaging Publications (vid. pág. 6).
- SMITH, J., 2023. *Diagnóstico de Escoliosis.* *Medical Imaging Journal* (vid. pág. 1).
- SMITH, J. et al., 2020. *Evaluación y Tratamiento en Ortopedia.* *Journal of Orthopedic Research* (vid. págs. 1, 36).
- SOMMERVILLE, Ian, 2005. *Software Engineering.* Pearson Education (vid. págs. 19, 20).
- SPILLNER, A., 2021. *Black-Box Testing Techniques.* Software Testing Institute (vid. pág. 52).
- TEAM, Horos Project, 2022. *Horos: An Open-Source DICOM Viewer* (vid. pág. 12).
- TEAM, Weasis, 2022. *WeAsis: A DICOM Viewer* (vid. pág. 13).
- TUDOSE, L., 2020. *NUnit: A Unit Testing Framework for .NET* (vid. pág. 46).
- Unit Testing Best Practices*, 2024 (vid. pág. 46).
- Variación de AUP para la UCI (AUP – UCI) – Metodología tradicionales*, 2020 (vid. pág. 14).
- Visual Paradigm 17.0*, 2022 (vid. pág. 16).
- What are Functional Requirements?*, 2023. Url: <https://www.visuresolutions.com/functional-requirements/> (vid. pág. 20).
- White Box Testing Techniques*, 2024 (vid. pág. 46).
- WILLIAMS, R. et al., 2019. *Métodos alternativos en la evaluación de la escoliosis.* Spinal Health Publishing (vid. pág. 6).
- WILLIAMS, T. et al., 2021. *Impacto de la Deformidad Espinal en la Biomecánica.* *Clinical Orthopedics Journal* (vid. págs. 1, 2).

Apéndices

.1. RF 5. Cargar Imagen

RFA 5. Cargar Imagen

Tabla 19. Descripción del RF 5 Cargar Imagen.

Descripción textual	Permite visualizar y cargar una imagen para el trazado de líneas.
Actor	Especialista (inicia).
Precondiciones	Seleccionar una medición.
Flujo de eventos	
Flujo básico Visualizar Imagen.	
1.	El Especialista selecciona cualquier medición.
2.	El Sistema carga y visualiza una imagen de una radiografía.
3.	El Sistema resalta la medición seleccionada.
4.	El requisito termina.
Pos-condiciones	
1.	Se visualizó la imagen radiográfica satisfactoriamente.
Flujos alternativos	
Flujo alternativo 1 'Seleccionar otra medición'	
1.	El Especialista al seleccionar otra medición, actualizará el panel mostrando otra imagen.
Flujo alternativo 2 'Seleccionar un botón que no sea una medición'	
1.	El sistema no mostrará ninguna imagen hasta que se seleccione alguna medición.
Requisitos especiales	N/A
Asuntos pendientes	N/A

.2. RF 6. Invertir Perpendicular de PI

RF 6. Invertir Perpendicular de PI

Tabla 20. Descripción del RF 6 Invertir Perpendicular de PI.

Descripción textual	Permite invertir la perpendicular de la primera línea trazada en PI.
Actor	Especialista (inicia).
Precondiciones	Seleccionar la medición PI y trazar una línea.
Flujo de eventos	
Flujo básico Invertir Perpendicular de PI.	
1.	El Especialista selecciona la medición PI.
2.	El Sistema carga y visualiza la imagen de una radiografía en plano sagital.

3.	El Sistema resalta la medición seleccionada.
4.	El Especialista da clic sobre la imagen para indicar el punto de inicio de la primera línea.
5.	El Especialista mueve el mouse hasta la posición en la que colocará el segundo punto de fin de la primera línea y da clic en dicha posición.
6.	El Sistema dibuja la perpendicular de la línea trazada.
7.	El Especialista da clic sobre el botón invertir Perpendicular PI.
8.	El Sistema invierte la perpendicular de la línea trazada.
9.	El Sistema actualiza la medición según la perpendicular invertida.
Pos-condiciones	
10.	Se invirtió la perpendicular satisfactoriamente.
Flujos alternativos	
Flujo alternativo 1 'Seleccionar otra medición que no sea PI'	
1.	El Especialista al seleccionar una medición que no sea PI, el botón de Invertir Perpendicular no tendrá función.
Flujo alternativo 2 'Seleccionar un botón que no sea una medición'	
1.	Al seleccionar el botón Invertir Perpendicular de PI no sucederá nada.
Requisitos especiales	N/A
Asuntos pendientes	N/A

.3. RF 7. Eliminar trazo

RF 7. Eliminar trazo

Tabla 21. Descripción del RF 7 Eliminar trazo

Descripción textual	Permite eliminar las mediciones hechas.
Actor	Especialista (inicia).
Precondiciones	Seleccionar una medición y trazar al menos una línea.
Flujo de eventos	
Flujo básico Eliminar.	
1.	El Especialista selecciona cualquier medición.
2.	El Sistema carga y visualiza la imagen de una radiografía.
3.	El Sistema resalta la medición seleccionada.
4.	El Especialista da clic sobre la imagen para indicar el punto de inicio de la primera línea.
5.	El Especialista mueve el mouse hasta la posición en la que colocará el segundo punto de fin de la primera línea y da clic en dicha posición.
6.	El Sistema dibuja la línea trazada.
7.	El Especialista da clic sobre el botón Eliminar.

8.	El Sistema actualiza el panel eliminando las mediciones trazadas.
Flujos alternativos	
Flujo alternativo 1 'Seleccionar una medición y no realizar trazos de líneas'	
1.	El Especialista al seleccionar una medición y no realizar mediciones el boton Eliminar no tendrá función.
Flujo alternativo 2 'Seleccionar un botón que no sea una medición'	
1.	Al seleccionar el botón Eliminar no sucederá nada.
Requisitos especiales	N/A
Asuntos pendientes	N/A

4. Diagrama del proceso de negocio sin sistema

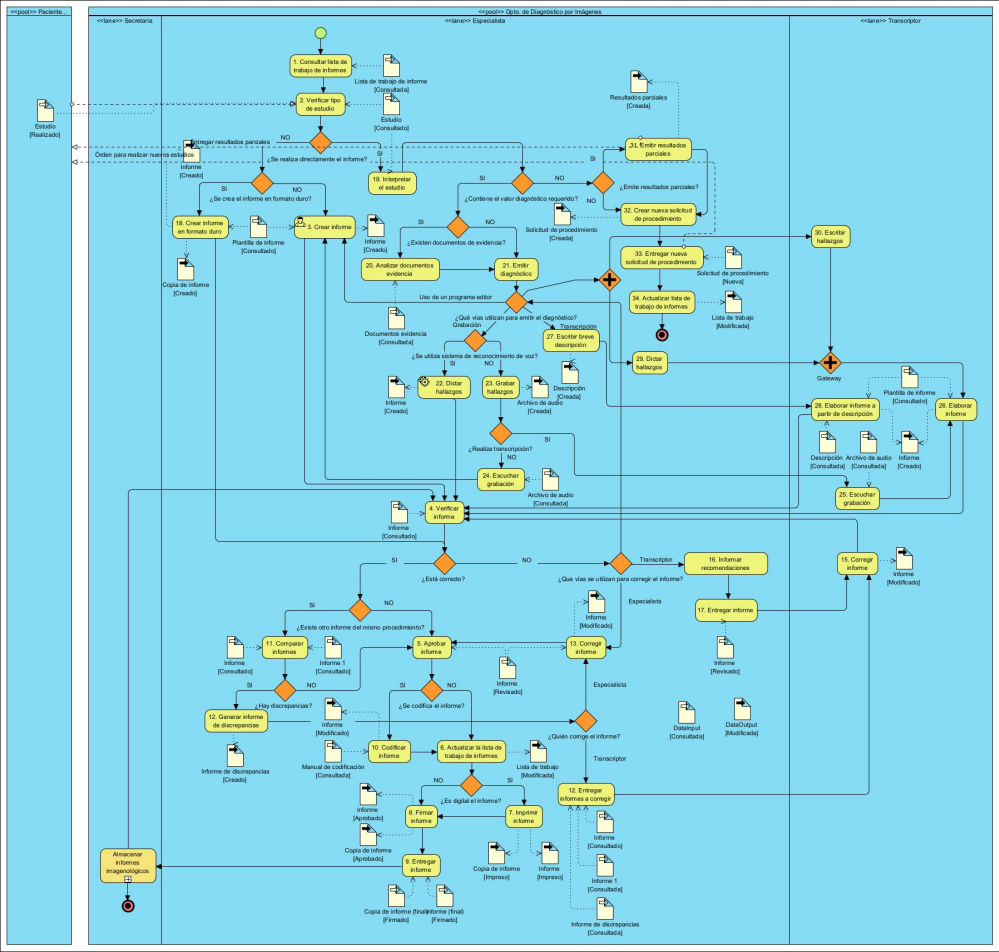


Figura 21. Diagrama del proceso de negocio sin sistema