

CIS1830CP02

Análisis del procesamiento de imágenes médicas pulmonares para el diagnóstico y
tratamiento del SDRA

Gómez Ganem Juan Miguel
Guayara Rodriguez Cesar Alejandro
Harker Gutierrez Erika
Sánchez Lozano Juan Manuel
Zárate Castillo Luis David

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
SYSTEMS ENGINEERING PROGRAM
BOGOTÁ, D.C.
junio de 2019

CIS1830CP02

Analysis of pulmonary medical image processing for diagnosis and treatment of SDRA

Authors:

Gómez Ganem Juan Miguel
Guayara Rodriguez Cesar Alejandro
Harker Gutiérrez Erika Jeniffer
Sánchez Lozano Juan Manuel
Zárate Castillo Luis David

UNDERGRADUATE FINAL PROJECT REPORT PERFORMED IN ORDER TO ACCOMPLISH ONE OF THE
REQUIREMENTS FOR THE SYSTEMS ENGINEERING DEGREE

Director

Leonardo Flórez Valencia

Juries of the Undergraduate Final Project

Carlos Andrés Parra Acevedo
Alejandro Sierra Múnera

Website of Undergraduate Final Project

<https://pegasus.javeriana.edu.co/~CIS1830CP02/>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
SYSTEMS ENGINEERING PROGRAM
BOGOTÁ, D.C.
May, 2019

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
SYSTEMS ENGINEERING PROGRAM**

President of the Pontificia Universidad Javeriana

Jorge Humberto Peláez Piedrahita, S.J.

Dean of School of Engineering

Eng. Lope Hugo Barrero Solano, PhD.

Head of the Systems Engineering Program

Eng. Mariela Josefina Curiel Huérfano, PhD.

Head of the Systems Engineering Department

Eng. Efraín Ortiz Pabón

Artículo 23 de la Resolución No. 1 de Junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

GRATITUDE

Queremos dar agradecimientos especiales al profesor Leonardo Flórez por el apoyo y la guía durante el desarrollo del trabajo, por su paciencia y carisma a la hora de orientarnos en el desarrollo de este proyecto y a la universidad por la educación y el apoyo brindado a través de los años que esos conocimientos fueron fundamentales para el desarrollo del trabajo de grado.

CONTENT

INTRODUCCIÓN	1
DESCRIPCIÓN GENERAL	3
OPORTUNIDAD Y PROBLEMA	3
<i>Contexto del problema</i>	3
<i>Formulación del problema</i>	5
<i>Solución propuesta</i>	6
<i>Justificación de la solución</i>	6
DESCRIPCIÓN DEL PROYECTO	7
<i>Objetivo general</i>	7
<i>Objetivos específicos</i>	7
<i>Entregables, estándares y justificación</i>	7
CONTEXTO DEL PROYECTO	8
BACKGROUND	8
<i>Procesamiento de imágenes</i>	8
<i>Segmentación de imágenes</i>	8
<i>Algoritmos de segmentación</i>	9
<i>Imágenes medicas</i>	9
<i>Tomografía axial computarizada (TAC)</i>	9
ANÁLISIS DE CONTEXTO	10
ANÁLISIS DEL PROBLEMA	10
REQUERIMIENTOS	10
<i>Requerimientos de Interfaces</i>	10
<i>Atributos de calidad</i>	12
<i>Requerimientos no funcionales</i>	13
SUPUESTOS Y RESTRICCIONES	13
ESPECIFICACIONES FUNCIONALES	14
DISEÑO DE LA SOLUCIÓN	17
ARQUITECTURA	17
<i>Estructura del sistema</i>	17

<i>Flujo de información</i>	18
DISEÑO BASE	19
<i>Clases base</i>	20
<i>Clases para el manejo de información</i>	20
<i>Clases relacionadas con los procesos</i>	20
<i>Extensión del modelo</i>	22
SOLUCIÓN DESARROLLADA	22
HERRAMIENTAS Y TECNOLOGÍAS	22
METODOLOGÍA	24
SOLUCIÓN	25
<i>Proceso de lectura de algoritmos</i>	25
RESULTADOS	25
CONCLUSIONES	26
IMPACT ANALYSIS OF THE PROJECT	26
CONCLUSIONS AND FUTURE WORK	26
REFERENCIAS	26
APENDICES	29

ABSTRACT

The Acute Respiratory Distress Syndrome (ARDS) is a dangerous syndrome that affects the lungs of a person, making breathing difficult or impossible. This syndrome needs be handled very carefully and for that reason doctors looks for technological support, looking for tools and programs that help them with the diagnostic and treatment. Currently there exist many tools that require time to use them effectively and in many cases, those tools don't present the expected results, making the doctor waste his time. The objective of this project is to build a modular tool capable of integrating image segmentation algorithms that helps in executing a better and more clear segmentation process over the medical images that are taken from the patients with ARDS, and this way, bringing the possibility to add new and better algorithms, giving the opportunity to choose which algorithm should be used and even become capable of deleting those which are unnecessary or obsolete.

INTRODUCCIÓN

Actualmente, la tecnología ha ayudado al diagnóstico de diferentes enfermedades sin necesidad de realizar procedimientos invasivos. Algunos de estos diagnósticos se realizan a través de la obtención de imágenes del interior del cuerpo humano mediante diferentes procedimientos como lo son: La tomografía computarizada y las radiografías con rayos X [1]. Para analizar estas imágenes es necesario hacer un procesamiento digital o análisis manual el cual es realizado por el médico, esto con el fin de tener más información para realizar el diagnóstico. Para el procesamiento digital a través de un computador generalmente se realiza un flujo de trabajo donde el usuario carga una imagen 3D y luego usa las vistas 2D para contornear manualmente un número reducido de segmentos diferentes [2].

Una de las herramientas más conocidas para realizar el procesamiento de imágenes es TurtleSeg. Esta es una herramienta interactiva de segmentación de imágenes en 3D. Implementa técnicas que permiten al usuario proporcionar una interacción intuitiva, pero mínima para guiar el proceso de segmentación 3D. [2] Otra herramienta que ha tomado impulso es The Medical Imaging Interaction Toolkit (MITK) el cual, al ser gratuito de código abierto, es de fácil obtención. [4]

Uno de los principales problemas al momento de diagnosticar mediante el procesamiento de imágenes una enfermedad como lo es el SDRA (Síndrome de Dificultad Respiratoria Aguda), es el tiempo requerido para realizar el proceso, puesto que la generación del resultado puede tardar varias horas e incluso días. Ese tiempo de demora puede generar lesiones permanentes en el paciente e incluso la muerte, ya que en el caso del SDRA, el tratamiento se debe realizar en una unidad de cuidados intensivos, pues el paciente puede tardar 4 o 5 días en presentar síntomas, lo que hace que cuando estos llegan a urgencias la enfermedad esté avanzada. [3]

Otro problema que se genera en los sistemas de procesamiento de imágenes está relacionado con la capacitación que necesitan los médicos para manejar los programas de procesamiento de imágenes correctamente, pues cada software maneja algoritmos diferentes y puede tener acciones diferentes. Esto requiere tiempo y, a su vez, implica poder de decisión sobre la mejor opción (o la más adecuada) para realizar el diagnóstico respectivo.

Actualmente, dado a la gran cantidad de recursos y de tiempo utilizados por este procesamiento, se han desarrollado diferentes algoritmos que realizan las operaciones que hacen parte del procesamiento con el objetivo de optimizar el trabajo. Por lo tanto, se crea la necesidad de tener una herramienta que facilite la integración de estos nuevos algoritmos y acciones y a la vez que “disgregue” los algoritmos obsoletos.

De ahí que, con el propósito de disminuir el tiempo de diagnóstico a través del procesamiento de imágenes, así como, el tiempo de capacitación para realizar estos procesos; la Universidad de Lyon en colaboración con la

Universidad De Los Andes y la Pontificia Universidad Javeriana desde 2015 están desarrollando un proyecto titulado: “Cuantificación de la aireación pulmonar en pacientes con síndrome de dificultad respiratoria aguda (SDRA) de imágenes de TC” (Titulo original: “Quantification de l'aération pulmonaire chez les patients avec le syndrome de détresse respiratoire aigüe (SDRA) à partir d'images scanner”). El cual tiene el objetivo de “desarrollar y validar algoritmos rápidos automatizados y herramientas de software inteligente para evaluar la respuesta del paciente a la ventilación y des este modo ayudar a la toma de decisión del médico en función de las imágenes de TC. Donde, los métodos propuestos deben cuantificar con precisión el estrés, la tensión y el reclutamiento (reapertura de los alvéolos colapsados) e identificar las regiones en las que se produce una distensión o un reclutamiento excesivos.” [5] [6]

En el siguiente trabajo se desarrolla un sistema llamado QuimeraTK el cual integra varios algoritmos de procesamiento de imágenes que pueden ejecutar diferentes acciones como: pintar, borrar, suavizar superficies, modificar región, añadir o eliminar un corte, entre otros, mediante pipelines y filtros que permiten integrar nuevos algoritmos que se vayan desarrollando al igual que reemplazar los algoritmos que se conviertan en obsoletos.

El desarrollo del sistema tiene como base ITK (The National Library of Medicine Insight Toolkit), el cual es un sistema de código abierto y multiplataforma para el procesamiento de imágenes médicas desarrollada en C++ [7], The Visualization Toolkit (VTK), que es un sistema de software de código abierto y gratuito para gráficos de computadora en 3D, modelado, procesamiento de imágenes, representación de volúmenes, visualización científica y trazado de gráficos en 2D. [8] Gracias a la compatibilidad que tiene con una gran cantidad de algoritmos tiene una mejor integración con el sistema que se presenta en este documento. Por último, para el desarrollo de la interfaz gráfica, se utilizó qt5, el cual permite una integración más limpia entre la vista, un modelo y unos controladores desarrollados principalmente en C++. [9]

DESCRIPCIÓN GENERAL

Oportunidad y problema

Contexto del problema

El SDRA es una enfermedad que se ha estudiado desde 1967, y entorno a la cual se han desarrollado diversos estudios, con el fin de mejorar su diagnóstico y tratamiento. El auge de la computación y posteriormente los avances alrededor del procesamiento de imágenes han sido de vital importancia para el manejo de esta enfermedad; sin embargo, aún se carece de valoraciones rápidas, y más importante aún, precisas. [10]

Por lo que se propone el desarrollar un sistema que permita integrar distintos algoritmos utilizados actualmente para el análisis y procesamiento de imágenes médicas. Esto con el fin de determinar las mejores alternativas que permitan construir, a futuro, una solución que brinde los resultados más claros y faciliten el realizar los dictámenes médicos para el tratamiento del SDRA.

La principal función del sistema respiratorio es el intercambio de gases entre el ambiente y la sangre, donde la sangre recupera el oxígeno del ambiente a través de los alvéolos pulmonares que se encuentran en el pulmón, y elimina el dióxido de carbono que es el desecho de los procesos metabólicos del cuerpo, convirtiendo a este en el órgano principal en este sistema [11].

Algunos de procesos de las afecciones del sistema respiratorio pueden afectar las propiedades elásticas del tejido pulmonar o modificar la función muscular de la caja torácica o el diafragma, bloqueando las vías respiratorias o los vasos sanguíneos. En los peores casos, estas afecciones pueden alterar los procesos del sistema nervioso central que controla la respiración [12]. Estos cambios en el pulmón afectan negativamente o interrumpe el intercambio normal de gases, generando niveles bajos de oxígeno en la sangre (hipoxemia) y, en consecuencia, insuficiencia en otros sistemas de órganos. [12] [13]

Entre las afecciones respiratorias se encuentra el SDRA, el cual genera un proceso edema pulmonar no hidrostático, que es el resultado de un desequilibrio entre las fuerzas que ocasionan la entrada de líquido en los alvéolos y los mecanismos para retirarlo, e hipoxemia [14]. Esto conlleva una alta morbilidad la cual aumenta bruscamente durante la epidemia de agentes infecciosos, y mortalidad (entre 30 y 50%), además de tener altos costos financieros en el sistema de salud a causa de los costos de los tratamientos y el manejo de los pacientes. [3] [5]

Uno de los problemas para diagnosticar a tiempo el SDRA es que la enfermedad se presenta entre 24 a 48 horas de que se produce la lesión o el edema en los pulmones, sin embargo, puede tardar 4 o 5 días en presentar síntomas. Esto causa que el tratamiento se deba realizar en una unidad de cuidados intensivos. Por otro lado, una vez se

presenta el SDRA, el paciente es más propenso a enfermedades respiratorias, como neumonía. Esto causa que el tratamiento no sea solo sobre el SDRA sino también sobre las enfermedades subyacentes, aumentando la mortalidad de pacientes que tiene el SDRA. [3] Por esta razón, tener un diagnóstico rápido y acertado cobra gran relevancia para salvar la vida de los pacientes.

Aunque el SDRA es una enfermedad que se ha documentado desde 1967, fue definida formalmente en 1994 por la Conferencia del Consenso Americano-Europeo (AECC por sus siglas en inglés). Sin embargo, esto generó diversas discusiones sobre el tema generando problemas al momento de diagnosticar la enfermedad. En 2011, un panel de expertos desarrolló la Definición de Berlín, centrándose en la viabilidad, fiabilidad, validez, y evaluación objetiva de su desempeño [13]. En esta definición se incluyeron variables como tiempo de inicio, grado de hipoxemia, origen del edema, y anormalidades radiológicas. En la Tabla 1 se muestra el resumen de los nuevos criterios establecidos por la definición de Berlín.

Tabla 1: Nuevas variables incluidas en la definición de Berlín para el SDRA. Tomado de “Nueva definición de Berlín de Síndrome de Dístres Respiratorio Agudo” escrito por el Dr. Francisco Arancibia Hernández [15]

NUEVA DEFINICIÓN DE BERLÍN DE SÍNDROME DE DIFICULTAD RESPIRATORIO AGUDO	
Tiempo de Inicio	Inicio dentro de 1 semana de conocida la injuria (daño o lesión) clínica o nuevo o deterioro de los síntomas respiratorios
Imagen torácica	Opacidades bilaterales -no explicable por derrame, atelectasia pulmonar lobar o pulmonar, o nódulos
Origen del edema	Falla respiratoria no explicable completamente por una insuficiencia cardíaca o la sobrecarga de líquidos. Necesita evaluación objetiva (ej. Ecocardiograma) para excluir edema hidrostático si no hay factor de riesgo presente.

En base a esta definición, una de las técnicas de diagnóstico es a través de imágenes médicas como una radiografía de tórax, la cual muestra qué partes de los pulmones se encuentran comprometidas o contiene líquidos a causa del edema, además de mostrar si los vasos sanguíneos del corazón se ven afectados y están provocando una hemorragia [16]. También es utilizado la tomografía computarizada (TC) de tórax la cual muestra diferentes vistas y ángulos del pulmón y el corazón brindando información más detallada sobre la estructura de estos órganos y las áreas afectadas. [12] [13] La TC de tórax teóricamente permite cuantificar la aireación del pulmón y, por lo tanto, clasificar a los pacientes como potencialmente respondedores o no a altos niveles de presión positiva al final de la espiración (PEEP por sus siglas en inglés), el cual es una técnica mecánica que se utiliza al ventilar a un paciente inconsciente donde se aplica una cantidad de presión en los pulmones al final de cada respiración. Desafortunadamente, esto requiere una larga y monótona segmentación pulmonar manual, que es incompatible con el manejo de emergencia de los pacientes. [5]

Para disminuir la dificultad y el tiempo de diagnóstico a través de imágenes médicas para el SDRA la Universidad de Lyon en colaboración con la Universidad De Los Andes y la Pontificia Universidad Javeriana desde 2015 están desarrollando un proyecto titulado “Cuantificación de la aireación pulmonar en pacientes con síndrome de dificultad respiratoria aguda (SDRA) de imágenes de TC” (Titulo original: “Quantification de l'aération pulmonaire chez les

patients avec le syndrome de détresse respiratoire aigüe (SDRA) à partir d'images scanner”) con el objetivo de “desarrollar y validar algoritmos rápidos automatizados y herramientas de software inteligente para evaluar la respuesta del paciente a la ventilación y ayudar a la decisión del médico en función de las imágenes de TC. Donde, los métodos propuestos deben cuantificar con precisión el estrés, la tensión y el reclutamiento (reapertura de los alvéolos colapsados) e identificar las regiones en las que se produce una distensión o un reclutamiento excesivos.” [6] [5]

Formulación del problema

Con lo mencionado en la sección anterior, se evidencia que el SDRA (Síndrome de Dificultad de Respiración Aguda), no tiene un tratamiento específico, sino que por el contrario se intenta cumplir dos objetivos; primero, tratar cualquier problema médico que produjo la lesión pulmonar; segundo, asistir la respiración del paciente (normalmente con un respirador) hasta que los pulmones sanen. La mayoría de las personas con SDRA son tratadas en la unidad de cuidados intensivos (UCI) o en la unidad de cuidados críticos (UCC) de un hospital. [17]

El tiempo que se tiene para poder brindar un diagnóstico es muy importante debido a que el tratamiento del SDRA, depende mucho del tratamiento oportuno que reciba el paciente. Actualmente existen diferentes herramientas para poder procesar las imágenes médicas tomadas de los pulmones del paciente, pero se tienen algunas limitaciones con ellas, el tiempo de aprendizaje que toman para ser manipuladas o los algoritmos de segmentación usados que no siempre brindan un resultado con información relevante o no son los más eficientes, resultando un poco complejo poder brindar un buen diagnóstico en el tiempo adecuado.

Solución propuesta

Se busca brindar ayuda a un estudio actual de la Universidad de Lyon en Francia, por medio de la evaluación y prueba de diferentes programas de código abierto para la segmentación de imágenes. Por otro lado, se busca integrar los diferentes algoritmos que brinda cada herramienta con el fin de poder optimizar los recursos de la máquina, a la vez de poder utilizar los mejores algoritmos que brindan las herramientas.

La solución consiste en proponer e implementar un sistema que permita integrar diferentes algoritmos de segmentación y procesamiento de imágenes, y a su vez que permita gestionarlos de tal manera que posibilite agregar nuevos, quitar antiguos o actualizar existentes. Además, la solución brinda una interfaz gráfica para que el médico pueda interactuar con los algoritmos integrados

Justificación de la solución

El integrar diferentes algoritmos en una sola herramienta, permite al médico hacer uso de una sola herramienta para obtener mejores resultados. Esto disminuye los tiempos de capacitación y a la vez reduce el tiempo empleado que se genera para la toma de la decisión de qué software se debe utilizar para el procesamiento de imágenes

médicas. Por otro lado, mejora los tiempos de procesamiento, pues al elegir los algoritmos que se van a integrar, se pueden utilizar los más eficientes, mientras se dan de baja los algoritmos que pueden entorpecer el diagnóstico.

Además, el estudio se encuentra en desarrollo desde hace tres años y carece de ayuda suficiente. El objetivo final de este estudio es el de lograr crear un programa que apoye a los médicos en el tratamiento del SDRA. Para lograrlo, necesitan segmentar las imágenes provistas para dicho tratamiento. El desconocimiento o falta de decisión para seguir con el estudio se debe a que no se ha desarrollado un estudio acerca de cuál de los programas disponibles actualmente para la segmentación de imágenes médicas se adecua mejor para lograr llegar a dicha solución. Adicionalmente, no se sabe si existe uno que se adecue lo suficientemente bien en su estado actual como para ser usado. Si bien en la fase de análisis de este proyecto no soluciona esto totalmente, da una primera aproximación para que el medico agilice su decisión.

Descripción del proyecto

Objetivo general

Desarrollar un sistema que integre distintos algoritmos de procesamiento de imágenes médicas que brinde información relevante para la toma de decisiones en los diagnósticos y tratamiento del SDRA

Objetivos específicos

- 1. Realizar una investigación de los algoritmos de procesamiento de imágenes médicas que existen actualmente.
- 2. Especificar los requerimientos técnicos y de usuario necesarios para la integración, interfaces, comunicación, y de priorización de los diferentes algoritmos.
- 3. Diseñar una arquitectura para un sistema de integración de los algoritmos de segmentación de imágenes médicas
- 4. Desarrollar una interfaz que permita la integración de los diferentes algoritmos de procesamiento de imagenes
- 5. Realizar pruebas sobre el sistema para validar los requerimientos técnicos y de usuario especificados anteriormente

Entregables, estándares y justificación

Tabla 2: Entregables, estándares y justificación

Entregable	Estándares asociados	Justificación
SPMP	ISO/IEC/IEEE 16326-2009	Muestra la planeación del proyecto al igual que la metodología que se va a utilizará durante su desarrollo
SRS	IEEE 830-1998	Especifica los requerimientos funcionales y no funcionales del proyecto al igual que las especificaciones de software y

		hardware necesarios para su correcto funcionamiento
SAD	ISO/IEC 42010 Descripción de arquitectura IEEE 1016-2009 Descripción del diseño	Describe la arquitectura del proyecto y el diseño de bajo nivel, ambos alineados a los requerimientos especificados en el SRS
Prototipo	ISO/9126 Estándar de calidad del software	Es el resultado final del proyecto, es decir, la primera versión del producto entregable al cliente
Manual de usuario	IEEE 1063-2001 Documentación para el usuario	Describe paso a paso la instalación y uso del producto final para los usuarios
Memoria del proyecto		Muestra características generales del proyecto

Contexto del proyecto

Background

Procesamiento de imágenes

El objetivo principal del procesamiento de imágenes es mejorar el aspecto de las imágenes con el fin de facilitar la búsqueda de información dentro de esta y denotar detalles relevantes para el observador. [18]

En este contexto, una imagen, que representa la salida, es construida a partir de señales provenientes de distintos objetos, que son la entrada del proceso. Así, las imágenes se pueden clasificar de diferentes maneras, por ejemplo, de acuerdo con la radiación o campo utilizado, la propiedad que se investiga o si las imágenes se forman directa o indirectamente. [19]

Por otro lado, el procesamiento de imágenes tiene diferentes procesos que realiza sobre la imagen para lograr sus objetivos. Entre estos se encuentra el filtrado digital, el cual consiste en resaltar o identificar características específicas en una imagen eliminando el ruido que se encuentre en esta. [20] También existe la reconstrucción de imágenes, que tiene como objetivo que un objeto se puede reproducir exactamente a partir de un conjunto de sus proyecciones tomadas desde diversos ángulos. [21] Sin embargo, el principal proceso en este proyecto es la segmentación de imágenes, la cual permite delimitar y destacar regiones con características específicas. [22]

Segmentación de imágenes

La Segmentación es la técnica de tratamiento digital de imágenes que permite extraer información de los objetos a partir de unas escenas dadas y plasmar esta información en un sistema estructurado, que en muchas ocasiones consta de una única estructura. [23]

La segmentación de imágenes médicas implica tres problemas principales relacionados con la imagen. Las imágenes contienen ruido que puede alterar la intensidad de un píxel, por lo que su clasificación se vuelve incierta, las imágenes muestran una intensidad no uniforme en la que el nivel de intensidad de una sola clase de tejido varía gradualmente en la extensión de la imagen, y las imágenes tienen un tamaño de píxel finito y están sujetas a promedio de volumen parcial en el que los volúmenes de píxeles individuales contienen una mezcla de clases de tejidos, por lo que la intensidad de un píxel en la imagen puede no ser coherente con ninguna clase. [24] Esto genera que los algoritmos de segmentación consuman una gran cantidad de recurso de la maquina donde se están ejecutando y por esto cobra relevancia elegir el adecuado para no perder tiempo de diagnóstico vital para el paciente.

Algoritmos de segmentación

La segmentación de imágenes médicas se clasifica en tres generaciones, cada una representando un nuevo nivel de desarrollo algorítmico. Los métodos de procesamiento más antiguos y de menor nivel ocupan la primera generación. El segundo se compone de algoritmos que utilizan modelos de imagen, métodos de optimización y modelos de incertidumbre, y el tercero se caracteriza por algoritmos pertenecientes a sistemas expertos que son capaces de incorporar conocimiento. Estas generaciones indican progreso hacia una imagen médica totalmente automática. [24]

A su vez, los métodos de segmentación automática se clasifican como supervisados o no supervisados. La segmentación supervisada requiere la interacción del operador durante todo el proceso de segmentación, mientras que los métodos no supervisados generalmente requieren la participación del operador solo después de que se complete la segmentación. Se prefieren los métodos no supervisados para garantizar un resultado reproducible; sin embargo, la interacción del operador todavía es necesaria para la corrección de errores en el caso de un resultado inadecuado. [24]

Imágenes medicas

El objetivo de todas las modalidades de imagen médica es visualizar los órganos internos del cuerpo de una manera no invasiva [21] Por ejemplo, estos sistemas de imágenes toman señales de entrada que surgen de varias propiedades del cuerpo de un paciente, como su atenuación de rayos X o la reflexión de ultrasonido. Las imágenes resultantes pueden ser continuas, es decir, analógicas o discretas, es decir, digitales; el primero puede convertirse en el segundo por digitalización. El desafío es obtener una imagen de salida que sea una representación precisa de la señal de entrada, y luego analizarla y extraer la mayor cantidad posible de información de diagnóstico de la imagen [18]

Tomografía axial computarizada (TAC)

Técnica radiológica que se sustenta en la obtención de imágenes por planos, basada en la emisión de rayos X y un algoritmo computarizado para reconstruir la imagen. A su vez esta genera imágenes transversales, también llamados “cortes”, del cuerpo. Estos cortes se llaman imágenes tomográficas y contienen información más detallada que los rayos X convencionales. [25]

Síndrome de Dificultad Respiratoria Aguda (SDRA): un proceso edema pulmonar no hidrostático, que es el resultado de un desequilibrio entre las fuerzas que ocasionan la entrada de líquido en los alvéolos y los mecanismos para retirarlo, e hipoxemia [13].

Análisis de contexto

A continuación, se muestran algunas soluciones, las cuales se refieren a aplicaciones, marcos u otros resultados similares a los que se desarrollaron en este proyecto.

En primer lugar, se encuentra el estudio “Evaluation of open-source software for the lung segmentation” realizado en 2016. En este estudio se hace una evaluación a un grupo de piezas de software usadas en la segmentación y procesamiento de imágenes pulmonares, las cuales se encuentran bajo licencias Open-Source. Para hacer dicha evaluación se tuvieron en cuenta cuatro atributos: funcionalidad, usabilidad, calidad de la segmentación y exportación. A cada uno de estos atributos se les asignó cierto valor y una justificación de dicho valor. Para la prueba de cada una de las piezas se usó un set de datos original de 1219 cortes y 616 MB en imágenes; sin embargo, para tener una mayor fluidez y prevenir errores se acortó el set de datos a 59 cortes y 61 MB en imágenes. A partir de los resultados presentados por los autores se realizó un ranking el cual está encabezado por el software *Pulmonary Toolkit*, seguido de *ITK-Snap*, *MITK*, *3D Slicer* y *TurtleSeg* en la última posición. Finalmente, presentan una explicación del por qué quedó el ranking tal como está y nos invitan a realizar pruebas similares, puesto que como se dijo anteriormente, las piezas son Open-Source. [26] Así, a través de este estudio, se pudo analizar las características de la herramientas y algoritmos que se encuentran implicados en el desarrollo del sistema QuimetraTK

En segundo lugar, el estudio “Computed tomography findings from patients with ARDS due to Influenza A (H1N1) virus-associated pneumonia” realizado en 2010. En el cual se muestran las características analizadas en la tomografía para generar los parámetros de segmentación para la detección del SDRA. Entre estas características se identifican los parámetros de la superficie del pulmón, así como el tamaño y la forma dentro de unos intervalos especificados por la asociación médica americana en la definición de Berlín. [27] Con estos parámetros dados por el estudio, se analizó los parámetros necesarios para la selección de algoritmos.

Análisis del problema

Requerimientos

Requerimientos de Interfaces

Interfaces con el sistema

El sistema debe funcionar en diferentes sistemas operativos. Por lo tanto, para el desarrollo y las pruebas se van a realizar en los siguientes sistemas operativos:

Tabla 3: Sistemas operativos donde funciona QuimeraTK

Sistema Operativo	Versión
Windows	7, 8, 10
Linux	Antergos, Ubuntu 18.04 LTS, Zorin OS 12.4
Mac	macOS High Sierra 10.13.4

Tabla 4: Requerimientos mínimos de hardware

Requerimiento	Mínimo
RAM	2 GB
GPU	1 GB
CPU	Quad-Core 3.0 GHz
Periféricos	Mouse y Teclado Estándar

Interfaces con el usuario

El usuario usará el sistema por medio de un mouse y teclado. No existirá otra manera. El sistema contará con una interfaz gráfica de usuario con todo lo necesario para que el usuario acceda a las funcionalidades del sistema.

Tabla 5: Productos para desarrollar la interfaz gráfica de usuario

Producto de Software	Descripción	Propósito de Uso	Versión
Qt	Es un framework de desarrollo de aplicaciones multiplataforma en C++. Cuenta con su propio preprocesador de C++[9] La selección de esta plataforma se dio por su compatibilidad en C++ con los algoritmos que proporciona ITK y VTK, además permite la ejecución en varias plataformas	Creación de la interfaz gráfica de usuario	5.11

Interfaces de Integración

En la siguiente tabla se muestra los programas de segmentación de imágenes que tienen algoritmos que se van a integrar en el sistema.

Tabla 6: Sistemas que proporciona algoritmos para integrar en QuimeraTK

Producto de Software	Descripción	Versión
TurtleSeg	Es una herramienta interactiva de segmentación de imágenes en 3D. Una característica clave de TurtleSeg es Spotlight, un sistema automatizado para dirigir automáticamente al usuario hacia el siguiente mejor lugar para proporcionar información. Por último, en un flujo de trabajo típico en donde implica que el usuario cargue una imagen 3D, puede luego utilizar 2D Livewire para contornear manualmente un número reducido de segmentos diferentes. La segmentación 3D completa se puede construir automáticamente utilizando la información proporcionada por el usuario. [28]	1.30
PTK	Pulmonary Toolkit es un paquete de software para el análisis de imágenes médicas 3D de pulmones para uso en investigación académica. PTKImage es un contenedor para una imagen en bruto, que proporciona campos y métodos que le permiten acceder y manipular los datos. De manera crucial, PTKImage mantendrá automáticamente el enlace entre los datos que representa y los datos originales de los cuales se derivó. Esto garantiza la integridad de los datos, algo que no ocurre en la mayoría de los kits de herramientas de imagen. [29]	6
MITK	The Medical Imaging Interaction Toolkit (MITK) es un sistema gratuito de software de código abierto para el desarrollo de software interactivo de procesamiento de imágenes médicas. MITK combina The Insight Toolkit (ITK) y The Visualization Toolkit (VTK) con un marco de aplicación.[4]	2017.07
ITK-SNAP	Es una aplicación de software utilizada para segmentar estructuras en imágenes médicas en 3D, cuya visión era crear una herramienta que se dedicaría a una función específica, segmentación, y sería fácil de usar y aprender. ITK-SNAP es gratuito, de código abierto y multiplataforma. ITK-SNAP proporciona una segmentación semiautomática utilizando métodos de contorno activo, así como la delineación manual y la navegación de imágenes. [30]	3.8
VTK	The Visualization Toolkit (VTK) es un sistema de software de código abierto y disponible gratuitamente para gráficos de computadora en 3D, modelado, procesamiento de imágenes, representación de volúmenes, visualización científica y trazado de gráficos en 2D [8]	8.2

Atributos de calidad

- *Desempeño*: Actualmente el procesamiento de imágenes sobre un TAC puede llegar demorar entre 7 y 10 minutos, el procesamiento de las imágenes de QuimeraTK obtenidas del TAC no debe ser mayor a 10 minutos.
- *Usabilidad*: El sistema debe proveer una interfaz de usuario que permita manipular la imagen carga-da en el sistema mediante los algoritmos integrados en este
- *Mantenibilidad*: Para hacer que el software sea fácilmente sostenible, se mantendrá la mayor independencia, entre clases, que sea posible para así evitar que pequeños cambios puedan afectar todo el sistema. También está enfocado a la alta cohesión, es decir que el sistema sea coherente con las responsabilidades que tiene
- *Portabilidad*: Los diferentes módulos que componen el sistema deben ser soportados por los sistemas operativos más comunes: Windows y Ubuntu -Linux.
- *Seguridad*: Al ser un software que maneja imágenes médicas, las cuales tienen información sensible y confidencial de los pacientes, la información no va a estar en servidores públicos, por lo que la aplicación va a ser standalone para conservar la información en los computadores del cliente.
- *Plataforma Tecnológica*: La arquitectura de software del sistema QuimeraTK, será implementada en el lenguaje de programación C++.

Requerimientos no funcionales

Desempeño

Tabla 7: Requerimiento no funcional 1: Tiempo de procesamiento

ID:	NF 001	Nombre:	Tiempo de procesamiento
Descripción o resumen:	El procesamiento de las imágenes obtenidas del TAC no debe ser mayor a 10 minutos.		
Criterio aceptación	Dado que el usuario ya selecciono la imagen a segmentar, cuando uno de los procesos sea especificado, y se inicie el procesamiento, entonces el sistema de responder con una imagen en no más de 2 minutos		
Restricciones	La imagen ya está en el computador que posee el sistema		

Mantenibilidad

Tabla 8: Requerimiento no funcional 2: Independencia de clases

ID:	NF 002	Nombre:	Independencia de clases
Descripción o resumen:	Se mantendrá la mayor independencia, entre clases, que sea posible para así evitar que pequeños cambios puedan afectar todo el sistema.		
Criterio aceptación	Dado que se realiza una pequeña actualización del sistema, cuando esta actualización se ejecute sobre el código del software, entonces solo se deben ver afectados los componentes involucrados directamente con la actualización		
Restricciones			

Supuestos y restricciones

El producto debe integrar varios algoritmos de procesamiento de imágenes médicas, por lo tanto, el sistema se enfoca en integrar al menos 2 algoritmos de procesamiento de imágenes y se enfocará en el procesamiento de imágenes del pulmón, pues es ahí donde se presentan los síntomas de la enfermedad.

El principal supuesto que se tiene en el proyecto es la claridad en las imágenes, en donde se entiende que las imágenes a procesar serán lo suficientemente claras para que los algoritmos que las usaran puedan funcionar correctamente.

A continuación, se muestra las restricciones relacionadas con el desarrollo del producto

Tabla 9: Restricciones del sistema QuimeraTK

Clasificación	Restricciones
Hardware	Debe ser un computador (portátil o de escritorio) que tenga las características mínimas que se requieren para poder ejecutar correctamente los programas de segmentación de imágenes en los computadores. Debe tener un mínimo de 2GB de RAM y 1GB de GPU
Software	Debe tener sistema operativo alguno de los siguientes sistemas operativos: <ul style="list-style-type: none">• Windows 7 o superior• Linux: Antergos, Ubuntu 18.04 LTS, Zorin OS 12.4• macOS High Sierra 10.13.4 El sistema debe poder conectarse con al menos dos programas de segmentación de imágenes médicas. La implementación se va a realizar en C/C++
Persistencia	Al ser un sistema stand-alone no es necesario que la aplicación se encargue de la persistencia de las imágenes Los computadores donde se ejecute el sistema deben tener espacio suficiente para almacenar la imagen original y la imagen procesada
Usuario	La aplicación debe tener un manejo de GUI básica para el usuario final. El sistema estará diseñado para usuario que lean español, inglés y/o francés El Sistema está diseñado para persona que no tengan discapacidad visual grave (ceguera) El sistema está diseñado para usuarios que tengan conocimientos básicos en el manejo de herramientas de procesamiento de imágenes.
Legales	Tener Licenciamiento de TurtleSeg Permisos de uso sobre algoritmos que no sean de código libre

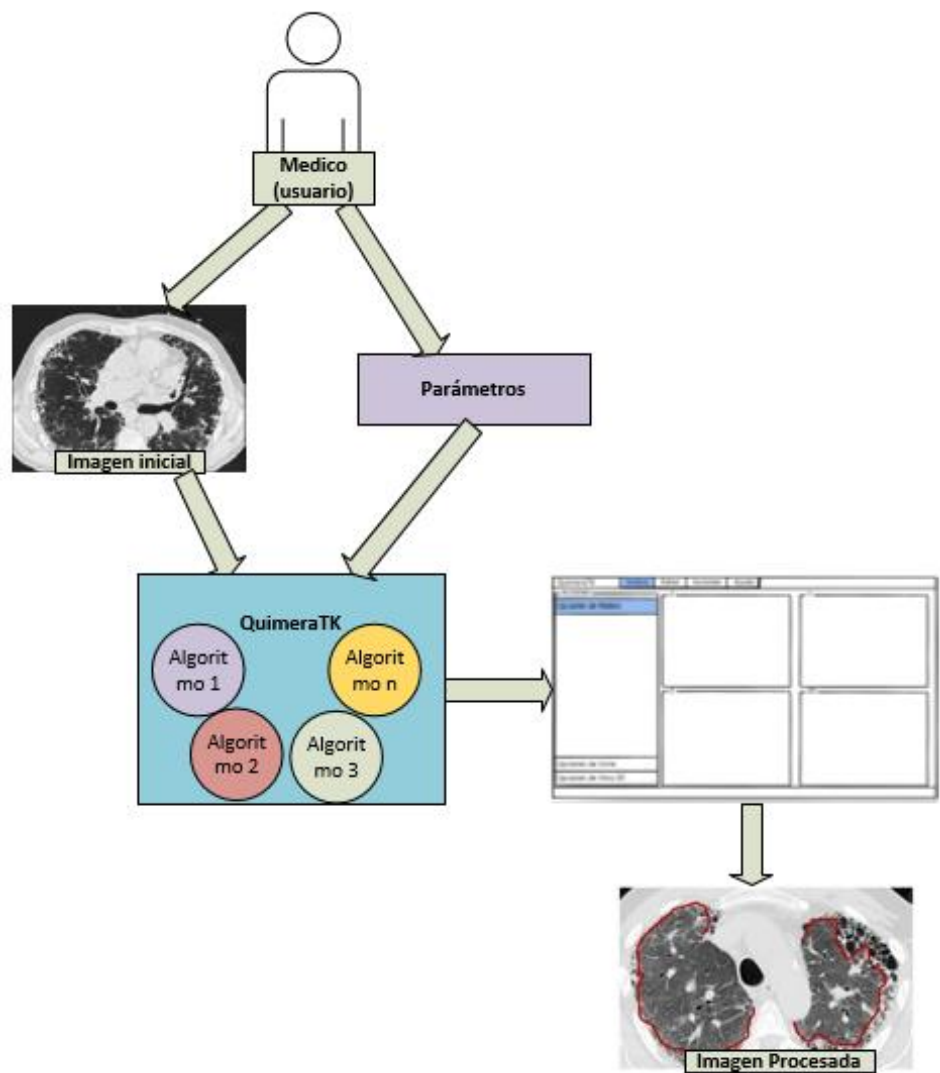
Especificaciones funcionales

El objetivo de integración de algoritmos se realiza con el fin de poder brindar información para la toma de decisiones en el diagnóstico y selección del tratamiento del SDRA. Al integrar diferentes algoritmos en un solo sistema facilita

la capacitación de los médicos puesto que solo será necesario aprender una herramienta. Por otro lado, permitirá obtener un mejor desempeño en la obtención de la segmentación de imágenes, pues solo utilizará las funciones necesarias para realizar los procesos.

El flujo principal del sistema inicia cuando el usuario, que es un médico, introduce una imagen y selecciona el procesamiento que desea hacer sobre esta. A continuación, esta información se envía al sistema QuimeraTK. Aquí se realiza el proceso y devuelve la imagen con el procesamiento elegido y la muestra a través de una interfaz gráfica con la que puede interactuar el usuario. En el siguiente diagrama se muestra el flujo descrito anteriormente.

Ilustración 1. Flujo del funcionamiento de QuimeraTK

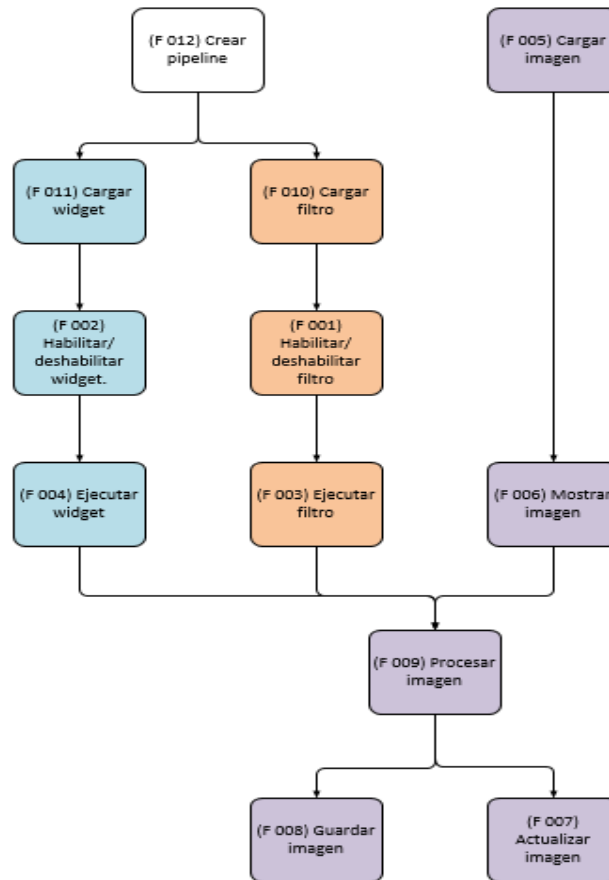


En las siguientes tablas, se muestra la descripción de los requerimientos funcionales que tiene el sistema:

Tabla 9: Requerimientos funcionales de QuimeraTK

ID	Nombre	Requerimiento
F 001	Habilitar/deshabilitar filtro.	El sistema debe permitir activar o desactivar cualquiera de los filtros disponibles
F002	Habilitar/deshabilitar widget.	El sistema debe permitir activar o desactivar cualquiera de los widgets disponibles
F003	Ejecutar filtro	El sistema debe permitir ejecutar cualquiera de los filtros disponibles, según los parámetro enviados por el usuario
F004	Ejecutar widget	El sistema debe permitir ejecutar cualquiera de los widgets disponibles, según los parámetro enviados por el usuario
F005	Cargar imagen	El sistema debe cargar una imagen seleccionada por el usuario para realizar el procesamiento de imágenes
F006	Mostrar imagen	El sistema debe mostrar la imagen cargada por el médico en la interfaz gráfica del usuario
F007	Actualizar imagen	El sistema debe actualizar la imagen con los resultados del algoritmo aplicado por el médico en la interfaz gráfica del usuario
F008	Guardar imagen	El sistema debe permitir guardar la imagen con las modificaciones realizadas por los procesamientos elegidos por el medico
F009	Procesar imagen	El sistema debe modificar la imagen cargada según el proceso y los parámetros enviados por el usuario
F010	Cargar filtro	El sistema debe permitir cargar un nuevo filtro al sistema e integrarlo, siempre y cuando este cumpla con la estructura de los filtros del sistema
F011	Cargar widget	El sistema debe permitir cargar un nuevo widget al sistema e integrarlo, siempre y cuando este cumpla con la estructura de los widgets del sistema
F012	Crear pipeline	El sistema debe crear pipelines que conecte los filtros y los widgets con la vista y enviar la imagen y los parámetros necesarios para le ejecución de estos.

Ilustración 2. Priorización de requerimientos funcionales



Diseño de la solución

Arquitectura

Estructura del sistema

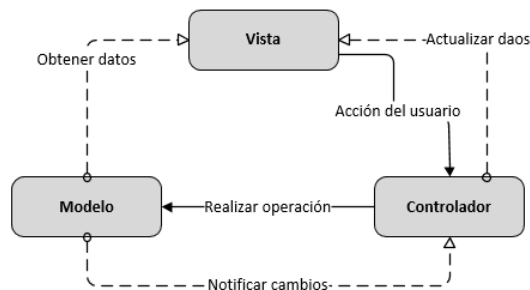
Arquitectura del sistema: Modelo- Vista - Controlador

- El Modelo es el objeto que representa los datos del programa y a la vez la lógica del negocio. Por lo tanto, aquí se maneja todas las transformaciones de los datos,[31] es decir, los algoritmos de procesamiento de imágenes. Como tal, el Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni

contiene referencias a ellos. Así, el modelo es alimentado por los algoritmos desarrollado por externos, sin necesidad de conocer el sistema en donde serán integrados.

- La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Interactúan preferentemente con el controlador, pero es posible que interactúe con el Modelo.[31] Está basado en los diseños que utiliza los actuales softwares de procesamiento de imágenes médicas.
- El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo.[31] Así, a través del Controlador se gestiona las entradas del usuario.

Ilustración 3. Estructura básica Modelo-Vista-Controlador



En el siguiente diagrama se muestra los componentes que constituyen QuimeraTK

Ilustración 4. Diagrama de componentes de QuimeraTK

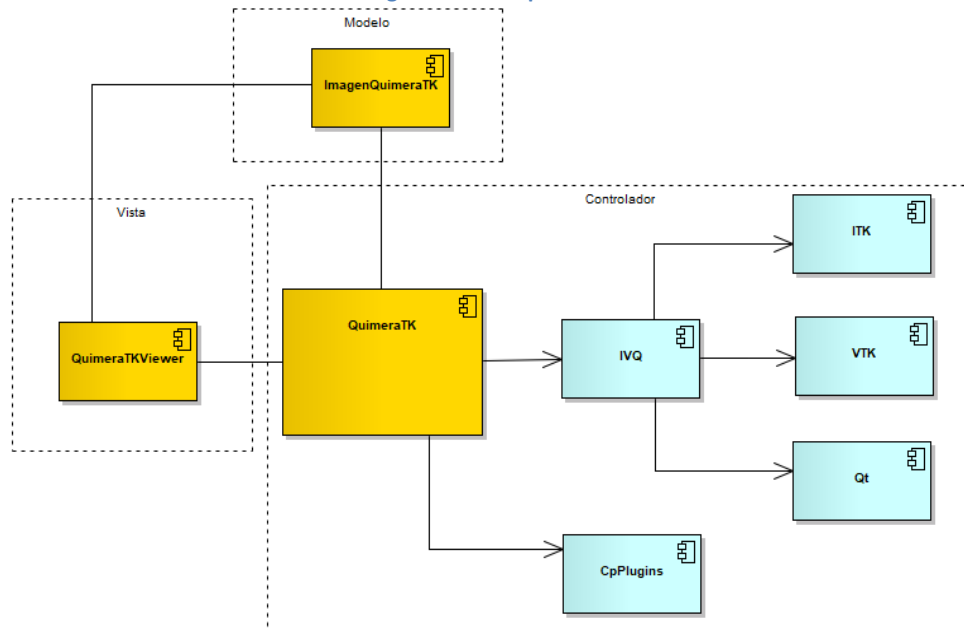


Tabla 9: Descripción de componentes

Componente	Descripción
QuimeraTKViewer	Es el encargado de generar la interfaz gráfica para el médico, donde este podrá manipular la imagen mediante los algoritmos integrados y guardar el resultado final
QuimeraTK	Su objetivo es conectar la interfaz gráfica con los algoritmos de procesamiento de imágenes
ImagenQuimeraTK	Su objetivo es guardar la imagen en el sistema mientras se realiza los procesamientos elegidos por el usuario e informar a la vista los cambios que existan en la imagen
IVQ	Es una un sistema para la integración de pipelines que utiliza ITK-VTK-QT. Este componente proviene de un proveedor externo pero es instalado en el computador donde se va a ejecutar QuimeraTK
ITK	Contiene filtros y widgets de procesamiento de imágenes y visualización especializadas en segmentación. Proviene de un proveedor externo
VTK	Contiene filtros y widgets de procesamiento de imágenes y visualización. Proviene de un proveedor externo
Qt	Provee herramientas para mostrar en la vista los resultados de los algoritmos utilizados en las imágenes. Proviene de un proveedor externo
CpPlugins	Sistema de plugins utilizado para la creación de pipelines que proporcionan las funcionalidades de estructuración, procesamiento y visualización de imágenes. Proviene de un proveedor externo

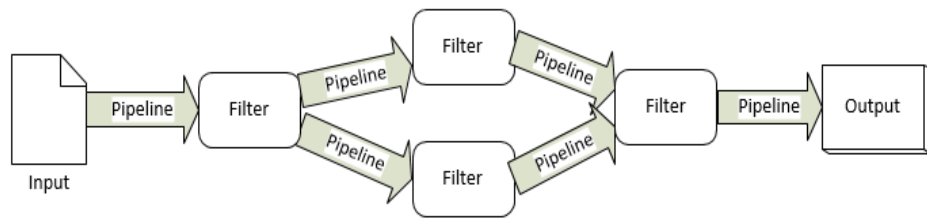
Flujo de información

Patrón de transferencia de datos: Filtros y tuberías (filter and pipelines)

El objetivo principal del sistema es integrar diferentes algoritmos. Para este fin, se utiliza el patrón de filtros y tuberías, el cual se aplica cuando los datos de entrada requiere una transformación para mostrar una respuesta en los datos de salida.[32] Esta transformación se realiza mediante una serie de componentes que realizarán los cálculos o la manipulación.

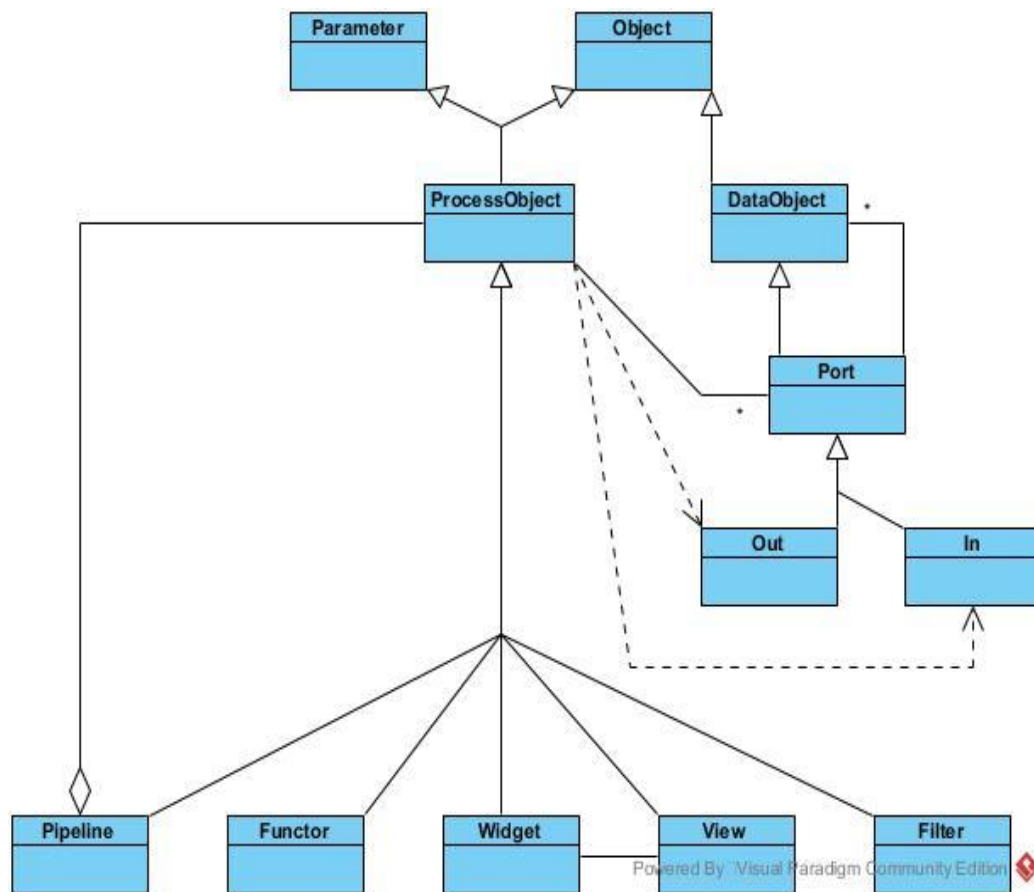
Una de las características más importantes del patrón para el sistema es que cada filtro trabaja de manera independiente de los componentes que se encuentren situados antes o después de él, lo que permite la integración de diferentes algoritmos, al igual que retiras los que sean obsoletos.

Ilustración 4. Funcionamiento básico del patrón filtros y tuberías (filters and pipelines)



Diseño base

Ilustración 5. Diagrama de dominio base del funcionamiento de los filtros y pipelines de QuimeraTK



Clases base

Tabla 11: Descripción de clases base del modelo

Nombre	Descripción
Parameter	Representa un conjunto de valores escalares.
Object	Clase que implementa un recolector de basura. Es la base de toda la jerarquía. Gracias a esta clase se obtiene el poliformismo que se utiliza en el sistema, además de proporcionar el uso de marcas de tiempo (timestamp) y permite que todo el modelo cpPlugins pueda utilizar el recolector de basura

Clases para el manejo de información

Tabla 12: Descripción de clases relacionadas al manejo de información

Nombre	Descripción
DataObject	Clase que representa y proporciona acceso a los datos que son utilizados en las transferencias para los procesos. [7]
Port	Clase que representa un espacio por el cual hacer una conexión entre ProcessObjects. Permiten el flujo de datos entre ProcessObjects
Out	Clase encargada de generar la conexión de salida ya sea hacia otro componente, como la salida final del procesamiento.
In	Clase encargada de generar la conexión de entrada hacia un componente.

Clases relacionadas con los procesos

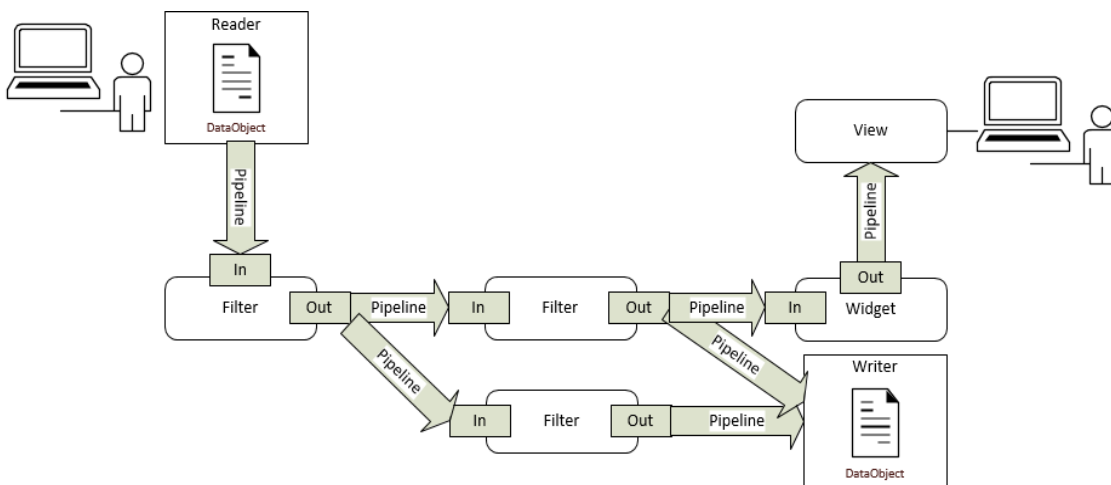
Tabla 13: Descripción de clases relacionadas al manejo de información

Nombre	Descripción
ProcessObject	Clase que representa un procesamiento de datos. Tiene puertos de entrada que actúan como entrada de datos para ser procesados y un puerto de salida con el resultado del procesamiento. Es decir, ProcessObjects operan en objetos de datos de entrada, produciendo nuevos objetos de datos como salida.
Pipeline	Clase encargada de generar el puente de comunicación entre componentes.
Functor	Clase que representa un objeto matemático que recibe parámetros y retorna un resultado. Un functor representa un cálculo
Widget	Clase que representa una funcionalidad que requiere de interacción con el usuario. Esta funcionalidad puede ser activada o desactivada. Solo un widget puede estar activo al tiempo. La interacción se da a través de Views
View	Clase que representa una vista de los datos siendo procesados en un ProcessObject. Un View se conecta y funciona como interfaz entre usuario y ProcessObject, haciendo que las interacciones entre el usuario y la vista actúen como entrada a dicho ProcessObject.
Filter	Clase que envuelve un procesamiento el cual se aplica sobre la entrada que recibe, puede arrojar un resultado final o conectar con otro filter para otro procesamiento. Es decir, envuelve la función de un determinado algoritmo para posteriormente ser enviada para su respectivo procesamiento. De esta forma la

interacción con el algoritmo se facilita al solo enfocarse en los datos necesarios para ejecutar la función. [7]

Los procesos de QuimeraTK están diseñados para ser usados bajo el patrón de filtros y tuberías (pipelines), y teniendo en cuenta la distribución en las clases y la generación de los procesos representados por objetos dentro del sistema (filter, widget, functor). Así, el uso de los algoritmos que se encuentran integrados dentro del sistema realiza un uso similar de los filtros y los pipelines, como se muestra en la siguiente imagen.

Ilustración 6. Funcionamiento básico de los filtros y pipelines de QuimeraTK

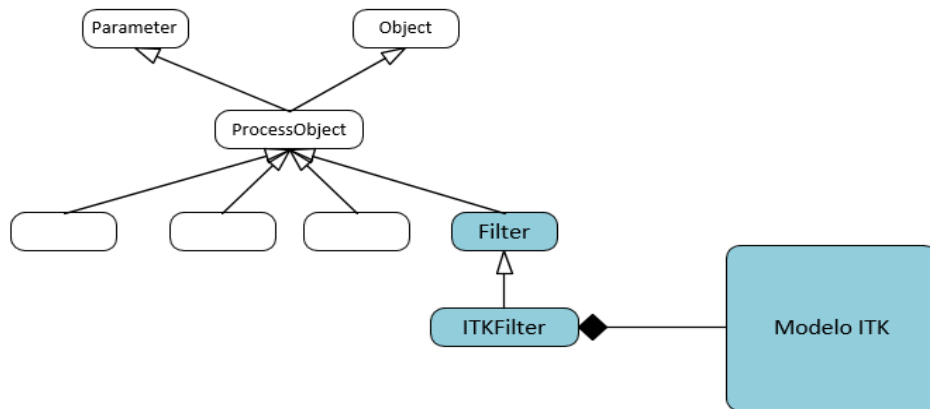


Extensión del modelo

Una de las principales características de ITK es que esta implementa en C ++. A la vez, utiliza un entorno de compilación conocido como CMake para administrar el proceso de compilación de forma independiente de la plataforma. Además, un proceso de ajuste automático (Cable) genera interfaces entre C ++ y lenguajes de programación interpretados como Tcl, Java y Python. [7]

El estilo de implementación de C ++ de ITK se conoce como programación genérica, es decir, utiliza plantillas (templates) para que el mismo código pueda aplicarse genéricamente a cualquier clase o tipo que sea compatible con las operaciones utilizadas. Esto permite que pueda integrarse con el modelo del sistema, esto con el fin de poder utilizar los algoritmos ya implementados en ITK para el desarrollo del sistema.

Ilustración 7. Extensión del modelo con ITK



Solución desarrollada

Herramientas y tecnologías

Una vez se tiene claro el diseño y la arquitectura básica del sistema QuimeraTK, se procede a definir las tecnologías que proporcionan el mayor beneficio sobre las cuales se construirá dicho sistema. Estas tecnologías están reflejadas en la siguiente imagen.

A continuación, se proporcionará una breve descripción de las tecnologías utilizadas para desarrollar los diferentes componentes de QuimeraTK:

- **cpPlugins:** Es un sistema de plugins utilizado para la creación de pipelines que proporcionan las funcionalidades de estructuración, procesamiento y visualización de imágenes.
- **IVQ:** ITK-VTK-QT es un sistema para la integración de pipelines provenientes de cpPlugins con plugins y algoritmos provenientes de las librerías anteriormente descritas.
- **CMake:** Es una herramienta multiplataforma, gratis y de código abierto que apoya en el proceso de integración de librerías y compilación de proyectos. Varias de las ventajas que nos ofrece CMake son, en primer lugar, hay gran soporte para proyectos que hacen uso de múltiples Librerías y herramientas y cada una de estas pueda tener una estructura de directorios compleja. En segundo lugar, permite una capa adicional de extensibilidad, permitiendo adaptar CMake a las necesidades del proyecto. En tercer lugar, las librerías y frameworks a continuación hacen uso de esta herramienta, lo cual facilita la integración del sistema.
- **Qt:** Es una serie de herramientas y widgets de código abierto y gratis para la creación de interfaces de usuario. Varias de las ventajas que nos ofrece QT son, en primer lugar, está implementada en el lenguaje de programación C++, lo cual nos facilita la integración con el proyecto. En segundo lugar, el uso de widgets para la creación de interfaces, lo cual proporciona facilidad para reutilizar código, conectar con sistemas internos y prototipar de manera cómoda y eficiente. En tercer lugar, Qt soporta múltiples plataformas de

hardware y software, lo cual nos da una facilidad de apuntar y exportar el sistema en varios sistemas operativos.

- ITK: Insight Sementation and Registration Toolkit es una framework multiplataforma para el desarrollo de aplicaciones especializadas en segmentación y registro de imágenes gratis y de código abierto. Varias de las ventajas que ofrece ITK son, en primer lugar y de forma similar a Qt, ITK esta implementado en C++ lo cual facilita la integración al proyecto. En segundo lugar, la implementación de ITK está separada por componentes individuales y autónomos, lo cual nos permite hacer uso de dichos componentes sin tener que importar partes innecesarias de la librería. En tercer lugar, ITK nos provee la capacidad de procesar imágenes provenientes de escáneres de resonancia magnética, tomografías o ultrasonido de manera rápida.
- VTK: Visualization Toolkit es una librería gratis y de código libre para el desarrollo de sistemas que la necesidad de trabajar con computación gráfica 3D, procesamiento de imágenes y visualización. Varias de las ventajas que ofrece VTK son, en primer lugar y manera similar a las anteriores dos librerías, la fácil integración al proyecto debido a que VTK está implementado en C++. En segundo lugar, VTK provee una capa de interacción en tiempo real que facilita la integración con la interacción por parte del usuario. En tercer lugar y de forma similar a ITK, la implementación de VTK está separada por componentes y filtros los cuales nos permiten simplificar el sistema sin tener que importar componentes innecesarios.

Metodología

El proyecto se desarrolló en tres fases, con el fin de cumplir los objetivos:

- Configuración e investigación: En esta fase se realizó una indagación sobre las principales herramientas para la segmentación de imágenes médicas que lideran actualmente el mercado, aquí se realizó comparaciones con respecto a su usabilidad, su interoperabilidad, su asequibilidad y a los algoritmos implementados por cada herramienta.
- Análisis de requerimientos y diseño: En esta fase se diseñó y analizó los requerimientos con los cuales el sistema deberá contar, ya sea su infraestructura, lenguaje, interfaz, UX y UI. Teniendo en cuenta los recursos con los cuales el usuario final contará, y como sería la manera más óptima y fácil tal que el usuario pueda interactuar con el sistema, todo esto basándonos en los resultados previos de la investigación, y de esta manera determinar que herramientas específicas se usarán, y cómo se usarán.
- Desarrollo y plan de pruebas: En esta fase se comenzó con el desarrollo del sistema especificado en la fase anterior, teniendo en cuenta todos los requerimientos de sistema, software, y usuario. Aquí, se llegó a un prototipo funcional al cual se le desarrolló un plan de pruebas.

Para el desarrollo del proyecto se apropiaron varios aspectos de la metodología Scrum. En primer lugar, se adoptó el uso de Sprints para asignar y desarrollar una serie de tareas a los integrantes del equipo con una duración fija de una semana. En segundo lugar, hicimos uso de Sprint Review para la revisión de las tareas y feedback por parte del

director. En tercer lugar, el mismo día que se realizaba el Sprint Review se realizaba el evento de Sprint Planning, donde se revisaban las tareas por realizar y asignaban a los integrantes del equipo.

Con el fin de desarrollar la solución propuesta se implementó una interfaz capaz de comunicarse con dos fuentes de algoritmos de segmentación de imágenes siendo estos IVQ y CpPlugins. La interfaz es capaz de identificar la cantidad de algoritmos envueltos dentro de cada uno de estos y generar una opción (botón) de interacción dentro de la interfaz.

Solución

La solución le aporta al médico una herramienta que permite utilizar algoritmos eficientes de procesamiento de imágenes para minimizar el tiempo de respuesta de la herramienta y así agilizar el diagnóstico

Por otro lado, esta solución le aporta al proyecto global de investigación “Cuantificación de la aireación pulmonar en pacientes con síndrome de dificultad respiratoria aguda (SDRA) de imágenes de TC” para integrar más algoritmos ahora y a futuro de una manera sencilla.

A continuación, se describe algunas características que tiene la solución final presentada en este proyecto.

Interfaz de usuario

En cuanto al desarrollo de la solución primero se diseñó la estructura y posibles opciones de interacción que la interfaz podría tener, contemplando así la cantidad de ventanas con los cortes a mostrar, la forma en que se organizarían y desplegarían los distintos algoritmos leídos por la interfaz.

Proceso de lectura de algoritmos

El proceso de lectura solo necesita de saber el nombre del algoritmo, y los datos de entrada que dicho algoritmo necesita para su posterior uso. De esta forma a medida que la interfaz recorre los algoritmos encapsulados por los proveedores genera una opción dentro de la interfaz con el nombre que dicho algoritmo

Control de calidad

Para el aseguramiento de calidad en el proyecto se definió el siguiente plan de pruebas:

- **Alcance:** Para este primer prototipo se realizarán pruebas manuales con base a los requerimientos definidos durante cada sprint y pactados con el product owner, enfocándonos en principalmente en requerimientos funcionales.
- **Niveles de pruebas:** En vista que QuimeraTk trabaja como la vista de otros sistemas, las pruebas se realizarán a nivel de integración y sistema, buscando que los módulos internos de QuimeraTk puedan conectarse entre sí y que las conexiones y operaciones con los otros, como cpplugins, puedan generarse cumpliendo los requerimientos básicos.

- **Tipos de pruebas:** En cuanto al tipo de pruebas realizadas, se efectuaron solo pruebas funcionales y de caja negra en este primer prototipo.

A medida que se avanzaba en cada sprint y a partir de los requerimientos definidos en cada reunión con el product owner se realizaban las pruebas siguiendo los lineamientos previamente definidos. Algunos casos de prueba que se trabajaron fueron los siguientes:

- **Conectar a Cppplugins**

Dado que cppplugins se conecta con QuimeraTk, cuando QuimeraTk inicie, entonces la interfaz de QuimeraTk debe desplegar las opciones detectadas de cppplugins en forma de botones.

- **Carga de algoritmos por XML**

Dado que el usuario desea cargar un nuevo algoritmo, cuando el usuario seleccione la opción de abrir archivo y seleccione el archivo con extensión XML, entonces QuimeraTk debe mostrar el nuevo algoritmo entre las opciones desplegadas.

- **Nombramiento de opciones**

Dado las opciones cargadas en QuimeraTk, cuando el usuario posicione el mouse sobre esta, entonces el nombre del widget o algoritmo debe ser desplegado sobre este.

- **Cargar imagen a QuimeraTk**

Dado que el usuario selecciona la opción de abrir archivo, cuando el usuario seleccione una imagen, entonces la imagen debe ser cargada y mostrada en QuimeraTk

- **Guardar imagen**

Dado que el usuario selecciona la opción de guardar, cuando el usuario guarde la imagen con el nombre deseado, entonces la imagen debe estar almacenada en el lugar elegido y con el nombre establecido.

- **Pintar sobre imagen con widget**

Dado que el usuario carga una imagen en QuimeraTk, cuando el usuario seleccione la opción de brocha, entonces el usuario debe poner pintar sobre la imagen previamente cargada.

- **Bloqueo dinámico de opciones**

Dado que el usuario selecciona un widget o un algoritmo, cuando el usuario haga click sobre la opción deseada, entonces las demás opciones deben bloquearse.

- **Desbloqueo dinámico de opciones**

Dado que el usuario deja de usar un widget previamente seleccionado, cuando el usuario libere la opción, entonces las demás opciones deben estar desbloqueadas.

Nota: Para el prototipo de esta entrega solo se tuvieron en cuenta casos de caminos felices,

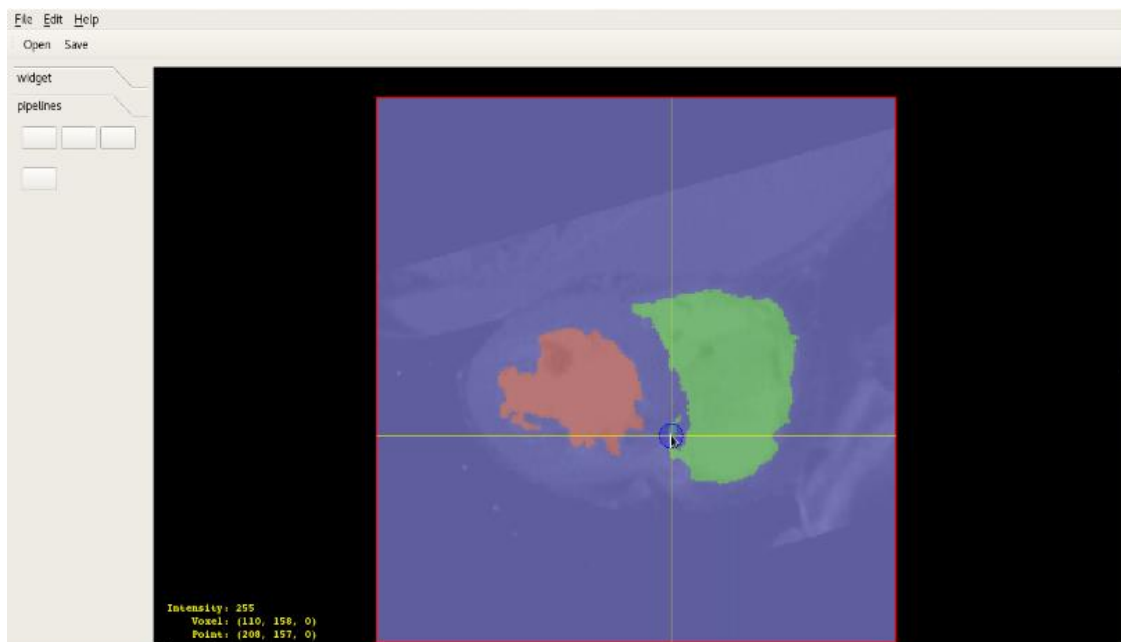
donde cada caso de prueba cuenta con la data correcta y necesaria para el cumplimiento básico de cada uno de los requerimientos definidos

RESULTADOS

En el transcurso del proyecto, se logró implementar una interfaz con la capacidad de usar múltiples algoritmos con la restricción de que solo se permite una interacción simultánea al tiempo con los datos de entrada.

La interfaz divide las herramientas por categoría. Al momento de escoger una, esta se activa desactivando la herramienta activa anterior si existe. Adicionalmente, permite lectura y escritura en diferentes formatos de imágenes.

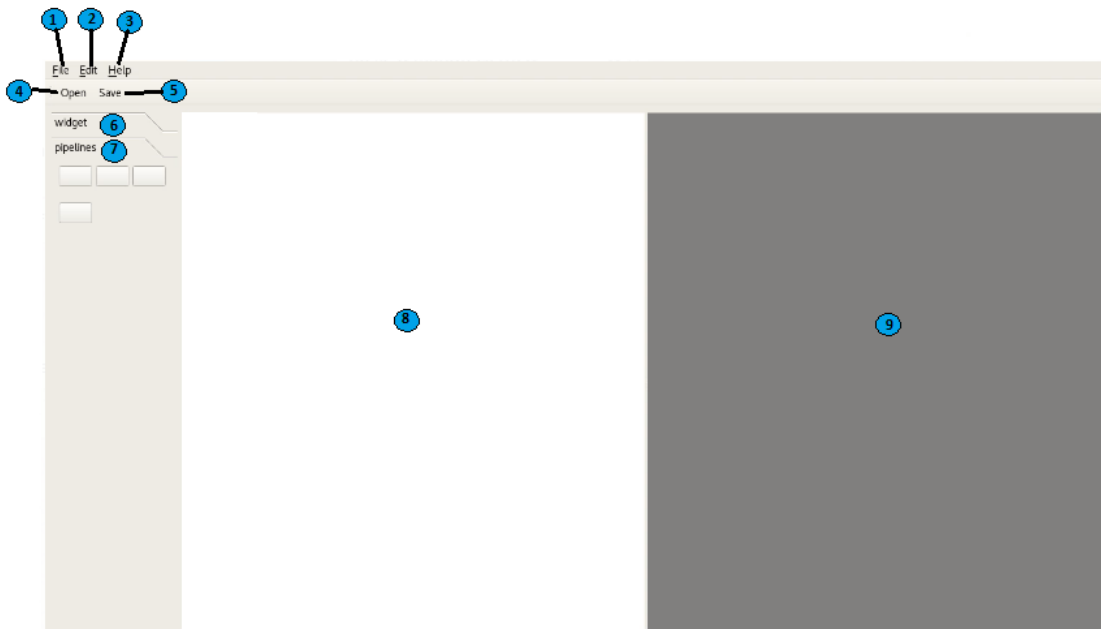
Ilustración 7. Resultado final del prototipo



A continuación, se describe cada parte que se encuentra en el prototipo

1. File: Menú desplegable donde se encuentran las opciones relacionadas con los archivos, entre las cuales están “Open image” y “Save image”
2. Edit: Menú desplegable donde se encuentran opciones relacionadas con la edición de las imágenes
3. Help: Menú desplegable donde se encuentran instructivos sobre el funcionamiento de QuimeraTK
4. Open: Botón de acceso rápido a la opción de “Open image” para abrir una imagen y colocarla en el visor de imágenes
5. Save: Botón de acceso rápido a la opción de “Save image” para guardar el resultado final del procesamiento de imágenes
6. Widget: Lista de widgets que se encuentran disponibles en QuimeraTK. Los widgets se cargan dinámicamente en el sistema
7. Pipelines: Lista de algoritmos de procesamiento de imágenes que se encuentran disponibles en QuimeraTK. Los widgets se cargan dinámicamente en el sistema
8. Visor de imagen 2D: Permite ver las imágenes cargadas en 2D y los procesamientos que se realizan sobre ella
9. Visor 3D: Permite visualizar los resultados de los procesamientos que generan resultados en 3D (Se implementará en trabajos futuros)

Ilustración 10. Resultado final del prototipo



CONCLUSIONES

Análisis de impacto del proyecto

1. A corto plazo, el impacto del proyecto es la integración de algoritmos y herramientas en una interfaz gráfica, brindando así una herramienta que permita familiaridad del proceso ante un usuario y fomentar el uso de esta.
2. A mediano plazo, se espera que el proyecto se haya integrado con herramientas y algoritmos que vayan más allá de la segmentación pulmonar, permitiendo el análisis y segmentación de imágenes que apliquen a más términos médicos.
3. A largo plazo, se espera que el proyecto sea herramienta que además de permitir segmentar las imágenes médicas, provea la funcionalidad de generar modelos en tres dimensiones a partir de las segmentaciones, brindando la posibilidad de imprimirlo en 3d o realizar una exploración virtual del mismo.

Conclusiones y trabajo futuro

Tal como se dijo anteriormente, este proyecto se está realizando en conjunto con la Universidad de Lyon, a través de la cual el proyecto seguirá evolucionando a partir de las necesidades de la investigación realizada allá.

Además, este proyecto puede evolucionar realizando el procesamiento de imágenes directamente en la nube para distribuir los recursos necesarios para realizar la segmentación y para iteraciones futuras tener la posibilidad de que el proyecto sea colaborativo.

REFERENCIAS

- [1] J. Jan, Medical Image processing, Reconstruction and restoration: Concepts and Methods. United States of America: Taylor & Francis Group, 2006.
- [2] «TurtleSeg». [En línea]. Disponible en: <http://www.turtleseg.org/index.php?page=software>. [Accedido: 06-may-2019].
- [3] G. R. Bernard et al., «The American-European Consensus Conference on ARDS. Definitions, mechanisms, relevant outcomes, and clinical trial coordination», Am. J. Respir. Crit. Care Med., vol. 149, n.o 3 Pt 1, pp. 818-824, mar. 1994.
- [4] «The Medical Imaging Interaction Toolkit (MITK) - mitk.org». [En línea]. Disponible en: [http://mitk.org/wiki/The_Medical_Imaging_Interaction_Toolkit_\(MITK\)](http://mitk.org/wiki/The_Medical_Imaging_Interaction_Toolkit_(MITK)). [Accedido: 13-nov-2018].

-
- [5] «Quantification de l'aération pulmonaire chez les patients avec le syndrome de détresse respiratoire aigüe (SDRA), à partir d'images scanner | CREATIS». [En línea]. Disponible en: <https://www.creatis.insa-lyon.fr/site7/en/node/44954>. [Accedido: 28-ago-2018].
- [6] «Quantification of pulmonary aeration in CT images of patients with acute respiratory-distress syndrome», Develop and validate automated fast algorithms and smart software tools to assess the patient's response to ventilation and assist physician's decision based on CT images. Namely, the proposed methods should accurately quantify stress, strain and recruitment (reopening of collapsed alveoli), and identify the regions where either over-distension or recruitment occur. [En línea]. Disponible en: <https://www.creatis.insa-lyon.fr/site7/fr/node/46602>. [Accedido: 28-ago-2018].
- [7] «The ITK Software Guide». [En línea]. Disponible en: <https://itk.org/ITKSoftwareGuide/html/>. [Accedido: 06-may-2019].
- [8] «About | VTK» [En línea]. Disponible en: <https://vtk.org/about/> [Accedido: 13-nov-2018]
- [9] «What's New in Qt 5 | Qt 5.12». [En línea]. Disponible en: <https://doc.qt.io/qt-5/qt5-intro.html>. [Accedido: 06-may-2019].
- [10] C. Pierrakos, M. Karanikolas, S. Scolletta, V. Karamouzou, y D. Velissaris, «Acute Respiratory Distress Syndrome: Pathophysiology and Therapeutic Options», J. Clin. Med. Res., vol. 4, n.o 1, pp. 7-16, feb. 2012.
- [11] Inkling, Pulmonary Physiology by Michael G. Levitzky | eBook on Inkling, 8.a ed. McGraw-Hill, 2013.
- [12] Handbook of Medical Imaging. Vol 2. Medical Image Processing and Analysis. .
- [13] American Medical Association, «Acute Respiratory Distress Syndrome, The Berlin Definition», JAMA, vol. 307, n.o 23, jun. 2012.
- [14] L. Pappas y G. Filippatos, «Congestión pulmonar en la insuficiencia cardiaca aguda: de la hemodinámica a la lesión pulmonar y la disfunción de la barrera alveolocapilar», Rev. Esp. Cardiol., vol. 64, n.o 09, pp. 735-738, sep. 2011.
- [15] F. Arancibia, «Nueva definición de Berlín de Síndrome de Distrés Respiratorio Agudo», Rev. Chil. Med. Intensiva, vol. 27, n.o 1, pp. 35-40, 2012.
- [16] T. M. Deserno, Biomedical Image Processing. Springer Science & Business Media, 2011.
- [17] American Thoracic Society, «¿Qué es el Síndrome de dificultad respiratoria aguda?», SERIE DE INFORMACIÓN AL PACIENTE. [En línea]. Disponible en: <https://www.thoracic.org/patients/patient-resources/resources/spanish/acute-respiratory-distress-syndrome-ards.pdf>.
-

- [18] S. E. Umbaugh, Computer Imaging: Digital Image Analysis and Processing. CRC Press, 2005.
- [19] G. Dougherty, Digital image processing for medical applications. California, United State of America: Cambridge University Press, 2009.
- [20] Y. Alvarez Germade, E. Morales, y O. Rodriguez Ramirez, «FILTRADO DIGITAL EN EL PROCESAMIENTO DE IMÁGENES EMPLEANDO MATLAB», 2010.
- [21] G. Kontaxakis, J. J. Vaquero López, y A. Santos, «Reconstrucción de imagen en tomografía por emisión de positrones», 2002.
- [22] A. Mansoor et al., «Segmentation and Image Analysis of Abnormal Lungs at CT: Current Approaches, Challenges, and Future Trends», RadioGraphics, vol. 35, n.o 4, pp. 1056-1076, jul. 2015.
- [23] C. Arenal, «SEGMENTACIÓN DE IMÁGENES MÉDICAS». [En línea]. Disponible en: <http://bibing.us.es/proyectos/abreproy/11854/fichero/Volumen+1%252FCapitulo+3.pdf>.
- [24] D. J. Withey y Z. J. Koles, Medical Image Segmentation: Methods and Software. .
- [25] «¿Qué es tomografía axial computarizada?» [En línea]. Disponible en: <https://www.cun.es/diccionario-medico/terminos/tomografia-axial-computarizada>. [Accedido: 23-oct-2018].
- [26] A. Alnaser, B. Gong, y K. Moeller, «Evaluation of open-source software for the lung segmentation», Curr. Dir. Biomed. Eng., vol. 2, n.o 1, pp. 515–518, 2016.
- [27] C. Grieser et al., «Computed tomography findings from patients with ARDS due to Influenza A (H1N1) virus-associated pneumonia», Eur. J. Radiol., vol. 81, n.o 2, pp. 389-394, feb. 2012.
- [28] «TurtleSeg». [En línea]. Disponible en: <http://www.turtleseg.org/index.php?page=support>. [Accedido: 13-nov-2018].
- [29] T. Doel, Lung medical image analysis and visualisation software for Matlab.: tomdoel/pulmonarytoolkit. 2018.
- [30] «ITK-SNAP Home». [En línea]. Disponible en: <http://www.itksnap.org/pmwiki/pmwiki.php>. [Accedido: 13-nov-2018].
- [31] Y. D. González y Y. F. Romero, «Patrón Modelo-Vista-Controlador.», Rev. Telemática, vol. 11, n.o 1, pp. 47-57, jun. 2012.
- [32] M. D. Meyer y D. P. Agrawal, «A modular pipelined implementation of a delayed LMS transversal adaptive filter», en IEEE International Symposium on Circuits and Systems, 1990, pp. 1943-1946 vol.3.

APENDICES

- *SPMP (SOFTWARE PROJECT MANAGEMENT PLAN)*
- *SRS (SOFTWARE REQUIREMENTS SPECIFICATION)*
- *SAD (SOFTWARE ARCHITECTURE DOCUMENT)*
- *Manual de Usuario*