

## **CIS1610IN02**

Modelado y manipulación del pulmón mediante la Interacción Natural de Usuario

Daniela Prada Sánchez  
Juan Sebastián Ulloa Monroy

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
CARRERA DE INGENIERIA DE SISTEMAS  
BOGOTÁ, D.C.  
2016



CIS1610IN02

Modelado y manipulación del pulmón mediante la Interacción Natural de Usuario

**Autores:**

Daniela Prada Sánchez

Juan Sebastián Ulloa Monroy

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO  
DE LOS REQUISITOS PARA OPTAR AL TITULO DE INGENIERO DE  
SISTEMAS

**Director**

Leonardo Flórez Valencia

**Página web del Trabajo de Grado**

<http://pegasus.javeriana.edu.co/~CIS1610IN02/>

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
CARRERA DE INGENIERIA DE SISTEMAS  
BOGOTÁ, D.C.

Mayo, 2016

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
CARRERA DE INGENIERIA DE SISTEMAS**

**Rector Magnífico**

Jorge Humberto Peláez Piedrahita, S.J.

**Decano Académico Facultad de Ingeniería**

Ingeniero Jorge Luis Sánchez Téllez

**Director de la Carrera de Ingeniería de Sistemas**

Ingeniera Mariela Josefina Curiel Huérfano

**Director Departamento de Ingeniería de Sistemas**

Ingeniero 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”*

## AGRADECIMIENTOS

En primera instancia, queremos agradecer a nuestras familias y compañeros universitarios por el apoyo incondicional que nos brindaron a lo largo de todo nuestro proceso académico, especialmente en el desarrollo de este proyecto. De igual manera a nuestro director de trabajo de grado, Leonardo Flórez Valencia, por brindarnos la oportunidad de llevar a cabo este trabajo de investigación.

## CONTENIDO

<b>CONTENIDO.....</b>	<b>v</b>
<b>INTRODUCCIÓN .....</b>	<b>10</b>
<b>I - DESCRIPCION GENERAL .....</b>	<b>11</b>
1.1. PROBLEMÁTICA .....	11
1.1.1. Formulación del problema que se resolvió .....	12
1.1.2. Impacto Esperado.....	12
1.2. DESCRIPCIÓN DEL PROYECTO .....	13
1.2.1. Objetivo general.....	13
1.2.2. Hipótesis .....	13
1.2.3. Objetivos específicos .....	13
1.3. METODOLOGÍA .....	13
1.3.1. Fase Metodológica 1 - Exploración .....	14
1.3.2. Fase Metodológica 2 – Planeación.....	15
1.3.3. Fase Metodológica 3 – Iteraciones.....	16
1.3.4. Fase Metodológica 4 – Documentación.....	17
<b>II – MARCO TEÓRICO.....</b>	<b>18</b>
2.1. ESTADO DEL ARTE.....	18
2.1.1. Tomografía axial computarizada del tórax .....	18
2.1.2. Segmentación del pulmón a partir una tomografía axial computarizada .....	20
2.1.3. Procesamiento de imágenes tridimensionales mediante ITK.....	24
2.1.4. Visualización de imágenes tridimensionales mediante VTK.....	26
2.1.5. Interacción natural de usuario .....	28
2.2. JUSTIFICACIÓN .....	35

---

<b>III - DESCRIPCIÓN DE LA SOLUCIÓN.....</b>	<b>38</b>
3.1. LIMITACIONES Y DEPENDENCIAS DEL PROTOTIPO.....	39
3.2. FUNCIONALIDADES DEL PROTOTIPO .....	40
3.2.1. Acercar y alejar .....	40
3.2.2. Trasladar .....	40
3.2.3. Rotar.....	40
3.2.4. Deformar .....	40
3.3. DISEÑO .....	40
3.3.1. Diseño del prototipo.....	40
3.4. DESARROLLO DE GESTOS .....	43
3.4.1. Alejar y acercar .....	43
3.4.2. Trasladar .....	46
3.4.3. Rotar.....	47
3.4.4. Deformar .....	49
3.4.5. Finalizar gesto.....	50
3.4.6. Estado neutro .....	52
3.5. DESARROLLO VTK .....	52
3.5.1. Alejar y acercar .....	54
3.5.2. Trasladar .....	54
3.5.3. Rotar.....	54
3.5.4. Deformar .....	55
3.6. DESARROLLO ITK.....	55
3.7. INTEGRACIÓN.....	56
3.7.1. Integración de gestos con VTK.....	56
3.7.2. Integración de ITK con VTK.....	58
3.7.3. Integración de gestos con ITK y VTK.....	59
<b>IV - PRUEBAS.....</b>	<b>61</b>

4.1.	PROCESO DE PRUEBAS.....	61
4.2.	RESULTADOS .....	63
<b>V -</b>	<b>CONCLUSIONES.....</b>	<b>66</b>
5.1.	ANÁLISIS DE IMPACTO .....	66
5.2.	CONCLUSIONES DEL PROYECTO .....	67
<b>VI -</b>	<b>REFERENCIAS Y BIBLIOGRAFÍA .....</b>	<b>69</b>
<b>VII -</b>	<b>ANEXOS.....</b>	<b>73</b>
	ANEXOS .....	73

## FIGURAS

<i>Figura 1. TAC del tórax. ....</i>	19
<i>Figura 2. Arquitectura tubo-filtro. ....</i>	24
<i>Figura 3. Malla simplex y malla triangula. ....</i>	25
<i>Figura 4. Arquitectura típica de VTK. ....</i>	28
<i>Figura 5. Esquema de Kinect. ....</i>	31
<i>Figura 6. Partes del Kinect. ....</i>	31
<i>Figura 7. Esqueleto detectado por Kinect. ....</i>	33
<i>Figura 8. Diseño del prototipo. ....</i>	42
<i>Figura 9. Diagrama de estados acercar y alejar. ....</i>	45
<i>Figura 10. Diagrama de estados trasladar. ....</i>	47
<i>Figura 11. Diagrama de estados rotar. ....</i>	48
<i>Figura 12. Diagrama de estados deformar. ....</i>	50
<i>Figura 13. Finalizar gesto. ....</i>	51
<i>Figura 14. Gesto posición neutra. ....</i>	52
<i>Figura 15. Pasos para desplegar un vtkPolyData. ....</i>	53
<i>Figura 16. Diagrama de secuencia de acercar y alejar. ....</i>	56
<i>Figura 17. Diagrama de secuencia de traslación. ....</i>	57
<i>Figura 18. Diagrama de secuencia de rotación. ....</i>	58
<i>Figura 19. Diagrama de secuencia de deformación. ....</i>	60

## TABLAS

<i>Tabla 1. Sensores de interacción natural _____</i>	37
<i>Tabla 2. Resultados porcentaje de precisión _____</i>	64
<i>Tabla 3. Resultados porcentaje de rendimiento _____</i>	65
<i>Tabla 4. Resultados encuesta _____</i>	66

## **ABSTRACT**

The Computerized Axial Tomography (CAT) is currently the most used technique for the detection of abnormalities of the lungs. A CAT scan exploration produces multiple images of the area of interest, generally transverse. Thanks to this sequence of images, it is possible to construct three-dimensional models. However, the analysis of a single organ can be an impossible task because of the overlapping of organs. For example, in thorax models, it becomes necessary to use segmentation algorithms to be able to examine the lungs properly, however, all these algorithms have errors. With this in mind, the following research proposes a natural interaction mode using the Microsoft Kinect for improving three-dimensional model of the lung.

## **RESUMEN**

La tomografía axial computarizada (TAC) es actualmente la técnica más empleada para la detección de anomalías en el área pulmonar. Una exploración TAC produce múltiples imágenes transversales de una sola área del cuerpo humano. Gracias a esta sucesión de imágenes, se hace posible la construcción de modelos tridimensionales. No obstante, el análisis de un solo órgano a partir de estos modelos puede ser una tarea imposible. En el caso de los modelos del tórax, se hace necesario emplear algoritmos de segmentación para lograr examinar adecuadamente el pulmón, sin embargo, todos estos algoritmos presentan errores. Teniendo esto en cuenta, el siguiente trabajo de investigación propone un modo de interacción natural basado en el dispositivo Kinect de Microsoft, para el mejoramiento del modelo tridimensional del pulmón.

## INTRODUCCIÓN

El presente trabajo de investigación pretende determinar si el Kinect de Microsoft es una herramienta adecuada para reducir el error presentado en modelos tridimensionales del pulmón. Este documento se divide en cinco secciones: descripción, marco teórico, descripción de la solución, pruebas y conclusiones.

En la primera sección, se encuentra la descripción de la problemática y el impacto esperado del trabajo de investigación. Así mismo, se presenta el objetivo general, los objetivos específicos y la metodología que se siguió para el desarrollo del trabajo.

En la segunda sección se presenta el estado del arte del proyecto, que abarca temas como los algoritmos de segmentación, el procesamiento y visualización de imágenes tridimensionales y la interacción natural de usuario. De igual manera, en esta sección se encuentra la justificación del proyecto.

En la tercera sección se describen las limitaciones, dependencias y funcionalidades del prototipo. Así mismo, se presenta la forma en la que se desarrollaron e integraron cada uno de los componentes del mismo.

En la cuarta sección se encuentra la descripción del proceso de pruebas que se aplicó sobre el prototipo desarrollado, junto con los resultados y conclusiones.

Por último, en la quinta sección se realiza un análisis del impacto disciplinar, social y económico del proyecto. Adicionalmente se exponen las conclusiones del trabajo de investigación.

## I - DESCRIPCION GENERAL

### 1.1. Problemática

Desde el descubrimiento de los Rayos X en el año 1895, las imágenes médicas se han convertido en una herramienta indispensable al momento de realizar diagnósticos. A lo largo de los últimos 60 años se han desarrollado diferentes modalidades para la obtención de imágenes médicas, estas han permitido practicar diagnósticos con un mayor nivel de precisión (Doi, 2006).

La tomografía axial computarizada (TAC) es una de las modalidades más empleadas hoy en día para generar imágenes médicas. Una tomografía integra múltiples imágenes de Rayos X, tomadas desde diferentes ángulos para producir imágenes transversales de la zona de estudio (Iyer, Levin, & Shea, 2006).

Las exploraciones TAC de la zona torácica proporcionan información que permite una rápida detección de anomalías en el área pulmonar, como tumores primitivos y derrames pleurales. Gracias a que el resultado de una exploración TAC se encuentra intrínsecamente en tres dimensiones, todas las imágenes producidas se pueden agrupar en un único modelo tridimensional (Martins, 2004).

Para analizar un solo órgano en el modelo tridimensional del tórax es necesario emplear un algoritmo de segmentación. Hasta el momento se ha desarrollado una gran cantidad de algoritmos de segmentación especializados en el pulmón, sin embargo, ninguno de estos ha logrado producir un resultado preciso: El modelo se genera con áreas faltantes o con partes de otros órganos (Withey & Koles, 2007).

Los médicos deben corregir manualmente la estructura del modelo mediante técnicas de interacción tradicionales como el de Ventana, Icono, Menú y Puntero (WIMP por sus siglas en inglés). No obstante, este estilo presenta inconvenientes en cuanto a ergonomía y facilidad de uso (Heathcote, 2003).

En este trabajo se desarrolló un esquema de interacción que facilita el proceso de disminuir el error, mediante la técnica de Interacción Natural de Usuario (NUI por sus siglas en inglés). NUI es un tipo de interacción hombre-máquina basada en el comportamiento de los usuarios. Este comportamiento refiere al conjunto de actividades que realizan las personas en su vida diaria para interactuar con su entorno, lo que permite que la interacción resulte intuitiva (Heathcote, 2003) (Ogiela & Hachaj, 2015).

La tecnología de interacción natural que se utilizó en esta investigación fue el sensor Kinect de Microsoft. Este sensor fue inicialmente diseñado para interactuar en diferentes entornos de juego. Sin embargo, debido a la capacidad de captura de datos de profundidad, movimiento, reconocimiento facial y de voz, se ha empleado en diferentes áreas del conocimiento; y para este proyecto estas características fueron suficientes (Khoshelham & Elberink, 2012). Además, cuenta con la ventaja de que es uno de los pocos dispositivos de interacción natural que no requiere contacto físico con el usuario, este es un aspecto muy importante ya que en el área de la medicina contar con un ambiente estéril es primordial.

### **1.1.1. Formulación del problema que se resolvió**

¿Cómo desarrollar un modo de interacción natural para mejorar el modelo de un pulmón que es segmentado a partir de una imagen TAC?

### **1.1.2. Impacto Esperado**

A corto plazo se espera que el prototipo creado en este trabajo sea empleado como base para la elaboración de un sistema. Este debe permitir la manipulación y deformación de modelos tridimensionales del pulmón.

A mediano plazo se espera que el software sea utilizado en diferentes hospitales del país como el Hospital Universitario San Ignacio.

A largo plazo se espera que el sistema desarrollado sea empleado en hospitales del exterior y además se implemente el esquema de interacción natural para la manipulación de diferentes órganos.

## 1.2. Descripción del Proyecto

### 1.2.1. Objetivo general

Desarrollar un esquema de interacción basado en el Kinect de Microsoft, para manipular y modelar pulmones.

### 1.2.2. Hipótesis

El dispositivo Kinect de Microsoft permite una interacción natural adecuada para manipular y modelar pulmones.

### 1.2.3. Objetivos específicos

1. Caracterizar el modo de interacción mediante historias de usuario
2. Diseñar las pruebas sobre las historias de usuario
3. Implementar un prototipo del modo de interacción
4. Realizar pruebas de usuario sobre el prototipo

## 1.3. Metodología

Para el desarrollo de este trabajo de grado se empleó una metodología basada en Extreme Programming (XP)(Larman, 2004), el cual es un método ágil que está enfocado en la creación temprana, incremental e iterativa de software con una alta retroalimentación del usuario. Aunque el presente proyecto es de investigación, XP fue adecuado para el desarrollo del mismo. Esto debido a que la creación del prototipo fue indispensable para obtener resultados en la investigación.

Extreme Programming se divide en 5 etapas: Exploración, planeación, iteraciones de entrega, producción y mantenimiento. Debido al tiempo con el que se contaba para la elaboración de este trabajo se eliminaron las últimas dos etapas y se agregó una de documentación.

La etapa de **Exploración** tuvo como objetivo hacer un primer acercamiento a las estimaciones de entrega y a las historias de usuario, que definen las necesidades de los usuarios. Adicionalmente, en esta etapa se incluyó el proceso de aprendizaje de las diferentes herramientas que se utilizaron. En la etapa de **Planeación** se definieron y se priorizaron las

historias de usuario, además se establecieron las fechas para el lanzamiento de las entregas. En las **Iteraciones de entrega** se desarrollaron las tareas derivadas de las historias de usuario junto con sus respectivas pruebas (Larman, 2004; Sommerville & Galipienso, 2005). Por último, se realizó la etapa de **Documentación**, en la cual se recopiló la información del trabajo de grado en un artículo y otros documentos.

### **1.3.1. Fase Metodológica 1 - Exploración**

A lo largo de esta fase se realizó la caracterización del modo de interacción del prototipo. Para ello, fue necesario realizar un proceso de recopilación y análisis de información relacionada con la herramienta Kinect.

Además, se identificaron las principales necesidades de los usuarios. A partir de esta información se logró llevar a cabo un primer acercamiento a las historias de usuario y a las estimaciones de tiempo y esfuerzo para el desarrollo de las mismas.

Al mismo tiempo se llevó a cabo un proceso de aprendizaje, en el cual se estudió el funcionamiento básico del SDK de Kinect y de las librerías VTK e ITK.

#### **1.3.1.1. Actividades**

1. Elaboración del estado del arte distribuido en la manipulación de imágenes médicas empleando Kinect como una herramienta para la interacción natural de usuario, la librería VTK para la visualización de modelos tridimensionales y la librería ITK para su respectivo procesamiento.
2. Elaboración del bosquejo de las historias de usuario.
3. Definición de las estimaciones para las historias de usuario.
4. Aprendizaje del SDK de Kinect
  - 4.1. Detección del esqueleto
  - 4.2. Detección de gestos
5. Aprendizaje de librería VTK
  - 5.1. Visualización de objetos tridimensionales
  - 5.2. Cambios geométricos de objetos tridimensionales
6. Aprendizaje de la librería ITK

- 6.1. Creación de mallas triangulares
- 6.2. Conversión de mallas triangulares a simplex y viceversa.
- 6.3. Deformación de mallas simplex

### **1.3.1.2. Resultados**

Al finalizar esta fase se obtuvo un documento con el estado del arte totalmente finalizado. Así mismo, se logró un cumplimiento parcial del primer objetivo específico del proyecto ([ver sección 1.2.3](#)) al desarrollar una primera versión de las historias de usuario.

### **1.3.2. Fase Metodológica 2 – Planeación**

En esta fase se especificaron las historias de usuario que fueron parcialmente definidas en la fase anterior. Posteriormente, se estableció la prioridad, costo y esfuerzo de cada una de ellas, para luego dividir las en tareas (ver anexo Historias de usuario). Estas historias de usuario permitieron definir las funcionalidades básicas del prototipo que se desarrolló.

A lo largo de la especificación de las historias de usuario, también se diseñaron las pruebas de usuario correspondientes. Finalmente, durante esta etapa se definió el diseño del sistema y la calendarización de tareas y entregas para cada iteración.

#### **1.3.2.1. Actividades**

1. Definición de las historias de usuario de forma sustancial
2. Asignación del esfuerzo y del costo de las historias de usuario
3. Priorización las historias de usuario
4. División de las historias de usuario en tareas
5. Calendarización
6. Diseño de las pruebas de usuario
7. Elaboración del diseño del prototipo

#### **1.3.2.2. Resultados Esperados**

Al finalizar la fase se obtuvo un documento con las historias de usuario completamente definidas (ver anexo Historias de usuario), por lo cual se cumplió del primer objetivo específico

del proyecto ([ver sección 1.2.3](#)). Adicional a esto se cumplió el segundo objetivo específico del proyecto al diseñar las pruebas de usuario ([ver sección 1.2.3](#)).

Por último, se realizó la calendarización de las tareas mencionadas previamente y el diseño del sistema.

### **1.3.3. Fase Metodológica 3 – Iteraciones**

En esta fase se realizaron las tareas definidas anteriormente, tal y como se especificó en la calendarización. Asimismo, se llevó a cabo una retroalimentación con el cliente (director trabajo de grado) al finalizar cada iteración.

La última iteración correspondió a una etapa de pruebas de usuario. Los resultados de estas pruebas fueron fundamentales para corroborar la hipótesis de la investigación.

#### **1.3.3.1. Actividades**

1. Iteración de diseño de historias de usuario
  - 1.1. Diseño de historia de usuario (incluye diseño de gestos)
  - 1.2. Retroalimentación de diseño de historias de usuario
2. Iteración de implementación de gestos
3. Iteración de implementación VTK
4. Iteración de implementación ITK
5. Iteración de integración ITK, VTK y gestos
6. Iteración de pruebas
  - 6.1. Ejecución de pruebas de usuario

#### **1.3.3.2. Resultados**

En esta fase se realizaron las tareas propuestas para cada iteración, como consecuencia se logró una versión incremental del prototipo ajustado a las necesidades y restricciones del cliente (director del trabajo de grado).

Debido a que el presente trabajo se realizó de manera incremental, a lo largo de las iteraciones se cumplieron los objetivos específicos 3 y 4 ([ver sección 1.2.3](#)).

#### **1.3.4. Fase Metodológica 4 – Documentación**

A lo largo de esta etapa se recopiló la información resultante de la fase de exploración, junto con la documentación y resultados de las pruebas de usuario que se obtuvieron en la iteración de pruebas. Por último, se realizó un artículo de investigación con estos datos, sustentados en el prototipo realizado.

##### **1.3.4.1. Actividades**

1. Recopilación de información y de resultados obtenidos
2. Evaluación del resultado de las pruebas de usuario
3. Elaboración del artículo de investigación
4. Elaboración del manual de usuario
5. Elaboración del manual de instalación
6. Elaboración del video de demostración del prototipo

##### **1.3.4.2. Resultados**

A partir del proceso que se llevó a cabo en este trabajo se realizó un artículo que recopilaba los resultados de la investigación. Además de esto, se elaboró un manual de usuario y un manual de instalación.

Los resultados de las actividades de esta fase figuran como anexos del presente trabajo.

## II – MARCO TEÓRICO

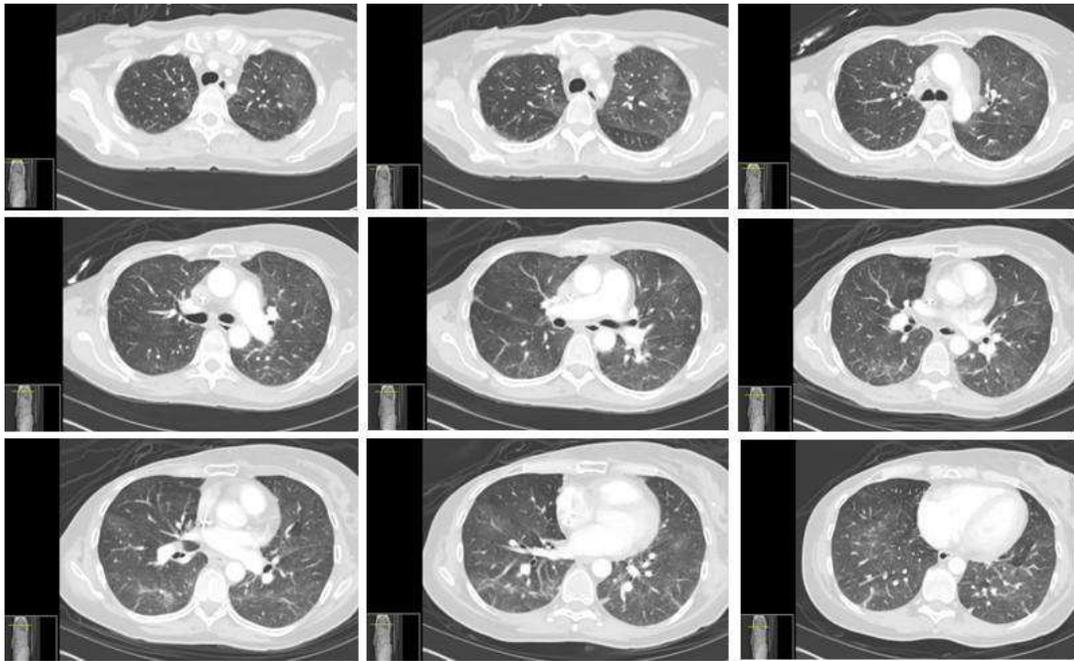
### 2.1. Estado del arte

#### 2.1.1. Tomografía axial computarizada del tórax

El tórax es un área comúnmente sometida a exámenes de diagnóstico médico como radiografías, ultrasonidos, tomografías y resonancias magnéticas. Esto se debe a que en el tórax se encuentra ubicado el pulmón, un órgano susceptible a una vasta lista de enfermedades que pueden ser fácilmente diagnosticadas mediante imágenes clínicas.

No obstante, la técnica más empleada es la tomografía axial computarizada (TAC), esto se debe a que es más eficiente al momento de generar imágenes del pulmón en comparación a las otras técnicas mencionadas. En primer lugar, las radiografías únicamente sirven como una base para definir si hay que realizar estudios más profundos; por otro lado, las imágenes a través de ultrasonidos y resonancias magnéticas pueden convertirse en procesos muy complejos, e incluso imposibles debido al aire en los pulmones (Schaefer-Prokop, 2013).

La tomografía axial computarizada es una técnica que hace uso de los Rayos X para producir imágenes detalladas del interior de cuerpo. Sin embargo, a comparación de las radiografías que también emplean Rayos X, las tomografías no producen imágenes con superposición ya que generan cortes trasversales del área de interés (Schaefer-Prokop, 2013). Dichos cortes, de igual manera hacen posible generar modelos tridimensionales. Lo anterior conlleva a que las imágenes TAC sean excelentes herramientas para la detección de nódulos pulmonares y por consiguiente la detección de cáncer. En la siguiente figura se pueden apreciar varios cortes de una tomografía axial computarizada del tórax.



**Figura 1. TAC del tórax.** Recuperado de

<http://www.swjpc.com/pulmonary/2012/2/24/february-2012-pulmonary-case-of-the-month.html>

Existen varias clases de tomografías, entre estas se encuentran las tomografías de alta resolución y las tomografías computarizadas de múltiples detectores, estas se describen a continuación.

Las tomografías de alta resolución (HRCT por sus siglas en inglés) producen cortes de entre 1 y 1.5 milímetros, por lo cual permiten realizar un análisis morfológico más detallado de las anomalías que se presentan en el tejido pulmonar. Gracias a esto, la tomografía de alta resolución se ha convertido en una herramienta invaluable en el diagnóstico y tratamiento de enfermedades difusas del pulmón como la fibrosis pulmonar (Nakata et al., 1985).

La tomografía computarizada de múltiples detectores (MDCT por sus siglas en inglés) es uno de los avances más grandes en el campo de las imágenes médicas en los últimos años. Esta, a comparación de las tomografías tradicionales ofrece un tiempo de adquisición más corto, una mayor cobertura y una resolución superior. Gracias a estos factores se hace posible la creación

de modelos tridimensionales con una alta precisión del área examinada (Das, Seith, & Mukhopadhyay, 2007).

Finalmente, vale la pena destacar que el rendimiento de la tecnología vinculada a las tomografías se ha duplicado cada año durante los últimos veinte años, lo que conlleva a un incremento en diagnósticos acertados y disminuyendo la necesidad de usar métodos de diagnósticos riesgosos e invasivos como la angiografía (Schaefer-Prokop, 2013).

### **2.1.2. Segmentación del pulmón a partir una tomografía axial computarizada**

Aunque la visualización tridimensional de las tomografías ofrece grandes ventajas para el médico, existe un problema cuando se quiere analizar un órgano por separado, como por ejemplo el pulmón. Debido a que los tejidos del pulmón poseen una baja densidad en comparación a otras partes del cuerpo que lo rodean, en la representación tridimensional los demás órganos se superponen y cubren el pulmón, lo cual dificulta el análisis y la detección de enfermedades (Li, Wang, Dai, & Zhang, 2015).

A lo largo de los últimos años se han realizado múltiples trabajos relacionados con la segmentación de los pulmones, por lo cual existe una gran cantidad de algoritmos para realizar esta tarea. Esto se debe a que es imposible aplicar un solo algoritmo de manera global sobre todos las imágenes del tórax; la estructura de los pulmones varía mucho de una persona a otra, y más aún en aquellos casos en donde los pulmones poseen grandes anomalías por patologías como nódulos, fibrosis quística o mesotelioma (Lo, 2010).

A continuación, se describe qué es la segmentación, cuáles son los problemas recurrentes en el área y los algoritmos que se han empleado según los autores del artículo *Medical Image Segmentation: Methods and Software* (Withey & Koles, 2007).

La segmentación es el proceso de dividir una imagen médica en subregiones, logrando separar áreas específicas como el pulmón. La segmentación manual es posible, sin embargo, no es un proceso recomendable ya que suele ser tedioso y consume mucho tiempo. Debido a esto la segmentación automatizada o semiautomatizada es la mejor opción.

Sin embargo, estos métodos de segmentación de pulmones suelen tener imprecisiones no solo por las diferencias estructurales entre los diversos pulmones, sino también debido a tres problemáticas: el ruido, la intensidad no uniforme, y el promedio del volumen parcial en las tomografías, por lo cual continúa siendo hoy en día un área de investigación abierta.

En primer lugar, el ruido hace referencia a aquella información no deseada que contamina una imagen. Las tomografías normalmente contienen ruido, el cual puede alterar la intensidad de los píxeles y por lo tanto la clasificación de los mismos durante el proceso de la segmentación. En segundo lugar, las tomografías presentan intensidades no uniformes, es decir, la intensidad de un mismo tejido varía a lo largo de toda la imagen. Finalmente, el promedio del volumen parcial de una tomografía hace referencia a que en una imagen de este tipo, los píxeles poseen un tamaño definido y pueden contener mezclas de diferentes clases de tejido.

En general los algoritmos de segmentación de imágenes médicas se encuentran divididos en tres categorías, en donde cada uno representa un mayor nivel de desarrollo. La primera categoría incluye técnicas de bajo nivel en donde se requiere muy poca información previa. Además de esto vale la pena destacar que los algoritmos comprendidos dentro de esta categoría son muy susceptibles a las tres problemáticas mencionadas anteriormente. Las técnicas de la primera categoría se describen brevemente a continuación:

- **Umbral:** Esta técnica emplea un umbral de intensidad para segmentar imágenes. Aquellos píxeles cuyo valor de intensidad se encuentra por debajo del umbral son clasificados dentro de un segundo plano (*background*) y aquellos que se encuentran por encima de este valor se ubican en un primer plano (*foreground*). (Gonzalez & Woods, 2002)
- **Crecimiento de región:** Esta técnica está basado en segmentar aquellos píxeles que cumplen con un criterio definido. Para ello, la técnica parte de un píxel semilla y verifica si cumple con el criterio, luego se examinan los píxeles adyacentes y así sucesivamente.
- **Detección de bordes:** En esta técnica la segmentación se hace posible a través de la detección de píxeles de borde adyacentes.

La segunda categoría difiere de la primera al introducir modelos de incertidumbre y métodos optimizados, además de que evita por completo las heurísticas. En estos métodos ya se hace un esfuerzo por superar las tres problemáticas, sin embargo, los resultados aún no son completamente acertados. Dentro de esta clasificación algunos métodos son: reconocimiento estadístico de patrones, agrupamiento, modelos deformables y redes neuronales. El agrupamiento es una de las técnicas más utilizadas de la segunda categoría, esta consiste en reunir los píxeles de las imágenes en grupos, basándose en un conjunto de características descriptivas. Para poder crear los grupos, los valores de las características deben ser normalizados, generalmente con valores entre 0 y 1.

En el caso de la técnica de las redes neuronales, primero se debe ingresar unos datos de entrenamiento que determinarán cómo se debe segmentar la imagen que posteriormente será procesada, de esta manera se clasifican los píxeles que cumplen con los datos ingresados al inicio de la técnica.

Las técnicas incluidas dentro la tercera categoría incorporan un nivel más alto de conocimiento, como información previa y reglas. Algunos métodos incluidos dentro de esta generación son: modelos de formas, modelos de apariencia, segmentación basado en atlas, segmentación basada en reglas y superficies acopladas. La segmentación basada en reglas es un método no supervisado que emplea reglas como guía para la segmentación. Estas reglas son generalmente derivadas de algoritmos de primera y segunda generación.

A continuación, se presentan algunos trabajos que han utilizado las técnicas mencionadas para la segmentación del pulmón:

El primero, *A Novel Supervised Approach for Segmentation of Lung Parenchyma from Chest CT for Computer-Aided Diagnosis* (Darmanayagam, Harichandran, Cyril, & Arputharaj, 2013), utiliza como base las técnicas de umbral y redes neuronales. En general el proceso consta de cuatro partes: la primera, es una segmentación general de los pulmones; la segunda consta en extraer las características de contorno de cada pulmón; la tercera es determinar si la segmentación ha finalizado basándose en las características que se obtuvieron en la segunda etapa; y por último se deben reconstruir los pulmones si alguno de ellos está incompleto. El último paso se logra siempre y cuando al menos un pulmón se encuentre segmentado

correctamente y así se pueden aplicar métodos gráficos como de reflexión y traslación sobre el otro.

Los resultados de este trabajo presentan una precisión del 97.37%, sin embargo, se excluyó en dicha investigación a aquellos pacientes que presentaban una enfermedad en los dos pulmones; era necesario que alguno de los dos estuviera sano para que la segmentación fuera posible.

El segundo trabajo, *A Fully Automatic Method for Lung Parenchyma Segmentation and Repairing* (Wei, Shen, & Li, 2013), hace uso de las técnicas de umbral iterativo óptimo y crecimiento de región 3D, entre otras. Para mejorar la segmentación resultante se utiliza un método basado en la cadena de código mejorada y el algoritmo de Bresenham (Bresenham, 1965). Esta investigación estuvo enfocada en la segmentación del parénquima pulmonar y en la segmentación e inclusión de aquellos nódulos que están adheridos a los tejidos del pulmón, no obstante, en el caso en que se quiera examinar otro tipo de enfermedades la precisión del proceso que se utilizó puede no resultar la más acertada.

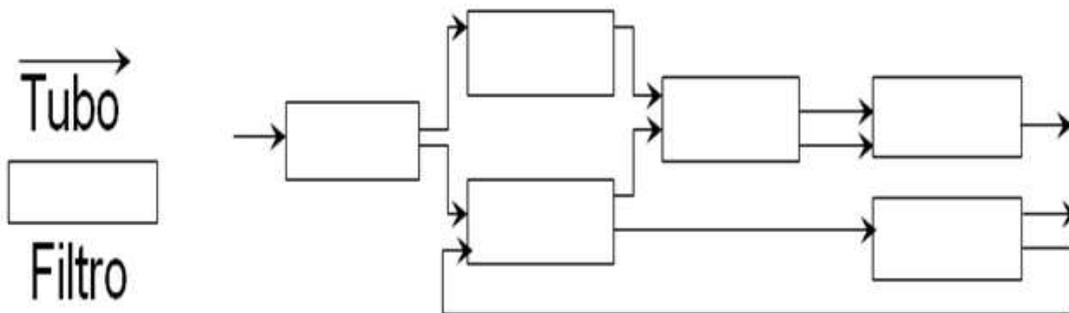
Por último, *Automatic Lung Segmentation in CT Images with Accurate Handling of the Hilar Region* (De Nunzio et al., 2011), propone una segmentación automatizada tridimensional para identificar nódulos pulmonares malignos. Utiliza técnicas como umbral adaptativo y crecimiento de región. El proceso que se lleva en este trabajo es el siguiente: primero, se realiza una segmentación sobre las vías respiratorias externas, luego sobre cada uno de los pulmones y por último, completa las cavidades y huecos que se encuentran en los pulmones, estas cavidades pueden incluir pequeños nódulos y vasos pulmonares. El problema que existe en el último paso es que al llenar esas cavidades existentes pueden generarse errores dependiendo del tamaño que se quiere suplir.

Como se puede observar existe algún grado de error en los trabajos que se mencionaron, lo cual conlleva a que los modelos segmentados resulten con áreas faltantes o sobrantes, es por esto que es necesaria una solución que complemente y corrija el resultado de los algoritmos de segmentación.

### 2.1.3. Procesamiento de imágenes tridimensionales mediante ITK

*Insight Registration and Segmentation Toolkit* (ITK) es una librería de código abierto que posee algunos de los algoritmos de segmentación que se mencionaron en la sección anterior, y otras características que permiten al desarrollador manipular imágenes tridimensionales. Esta librería tiene como base la programación genérica, apuntadores inteligentes, una arquitectura tubo-filtro y Multihilo. (Johnson, McCormick, & Ibanez, 2014)

En la arquitectura tubo-filtro (ver figura 2) de ITK se encuentran los objetos de datos (tubos) que contienen la información, y los objetos de procesos (filtros) que se encargan de modificar esta información. Esta arquitectura posee dos características importantes. La primera, es que tan pronto se encuentre una entrada o un cambio en algún filtro, los demás filtros se actualizan automáticamente. La segunda es el proceso de transmisión (*streaming*), que permite desglosar la información en elementos pequeños para ser procesados uno a uno, y luego unirlos para obtener el resultado final (Johnson et al., 2014).



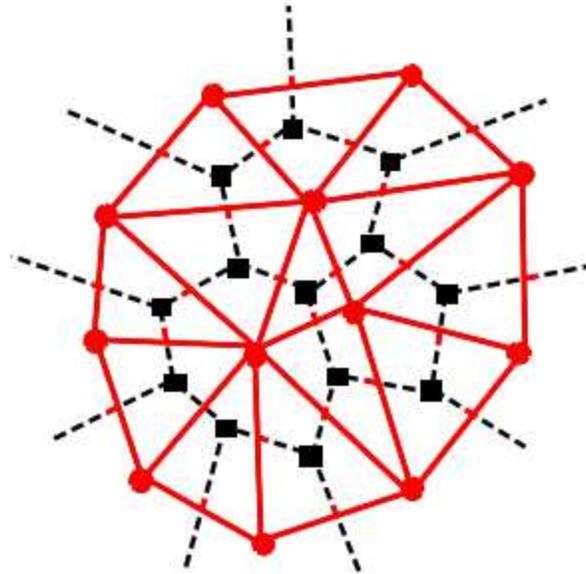
**Figura 2. Arquitectura tubo-filtro.** Recuperado de

<http://dallanmnr.blogspot.com.co/2011/08/estilos-de-arquitectura.html>

En ITK los datos se representan principalmente a través de imágenes y mallas, ambas implementadas en las clases `itk::Image` e `itk::Mesh`, respectivamente. En particular, las mallas están compuestas por un conjunto de puntos, celdas y límites de celdas. (Johnson et al., 2014)

Dentro de los tipos de mallas implementadas en ITK, se encuentra la malla simplex que fue inicialmente definida por Hervé Delingette (Delingette, 1994). Esta malla, a diferencia de la

mallas triangulares, que es construida a partir de celdas de tres puntos, tiene la característica de suavizarse luego de deformarse, ya que sus celdas están conformadas por seis puntos (ver figura 3). Esto a su vez evita la adición de ruido al modelo y la pérdida de datos durante el proceso de deformación. Dentro de su artículo, Delingette define las fuerzas internas y externas que permiten al modelo deformarse con las cualidades mencionadas anteriormente.



**Figura 3. Malla simplex y malla triangular:** La malla triangular se encuentra en rojo y la malla simplex con líneas negras punteadas. Tomada de (Delingette, 1994).

Con respecto a ITK, tanto la malla triangular como la malla simplex se encuentran definidas. Los objetos de datos que se reciben de entrada, generalmente están representados como mallas triangulares, por lo que si se quiere utilizar la malla simplex y sus filtros de deformación se debe pasar por un filtro que permita la conversión entre mallas. En ITK, la clase que realiza esta acción es `itk::TriangleMeshToSimplexMeshFilter`.

Gracias a las características de suavizado y reducción de ruido de las mallas simplex, concluimos que estas son las más adecuadas para representar el modelo tridimensional en el prototipo. Es importante que la deformación que se realiza para mejorar los modelos de los pulmones cuente con estas características, ya que de lo contrario se puede entorpecer el proceso del diagnóstico.

En el caso del prototipo se empleó una esfera tridimensional representada por una malla simplex, esta esfera es equivalente al modelo tridimensional del pulmón. Como se mencionó anteriormente esto es debido a que los dos modelos son homotópicos, es decir que uno puede deformarse continuamente hasta convertirse en el otro. (Crossley, 2005).

#### **2.1.4. Visualización de imágenes tridimensionales mediante VTK**

A través de las imágenes médicas de dos dimensiones se puede construir un modelo tridimensional que permite a un médico realizar un análisis cualitativo y cuantitativo de la zona que está examinando (Dong, Xia, Zhang, & Cai, 2013). La representación en tres dimensiones de las tomografías supone un salto importante en el campo de las imágenes médicas. Gracias a esto, los médicos no tienen que evaluar cada una de las imágenes en dos dimensiones e imaginarse cómo son los órganos, al contrario, esta nueva técnica permite en cuestión de segundos visualizar el modelo real de la zona de interés.

Estas imágenes tridimensionales permiten que el proceso de realizar un diagnóstico sea más sencillo, como es el caso de la detección de nódulos pulmonares. El médico puede manipular el modelo para visualizarlo desde diferentes ángulos y así determinar la ubicación exacta y dimensión de estas lesiones. De esta manera, los modelos tridimensionales facilitan los procesos que se realizan sobre tomografías en dos dimensiones, como la detección temprana del cáncer, el diagnóstico de enfermedades difusas del pulmón, y el progreso de los tratamientos. (como se menciona en la [sección 2.1.1](#))

Anteriormente, la reconstrucción y visualización de estas imágenes tridimensionales sólo se podía lograr utilizando máquinas UNIX y a través de software creado de manera independiente (McAuliffe et al., 2001). Hoy en día estas tareas se pueden realizar en equipos de cómputo económicos que cuenten una tarjeta gráfica apropiada y un software basado en algoritmos existentes. VTK es un software de código abierto que permite la visualización, procesamiento y renderizado de imágenes tridimensionales, además funciona en diversos sistemas operativos como Windows, OS y Linux. La librería de VTK están escritas en C++ pero posee *wrappers* para Python, Tcl y Java (Dong et al., 2013).

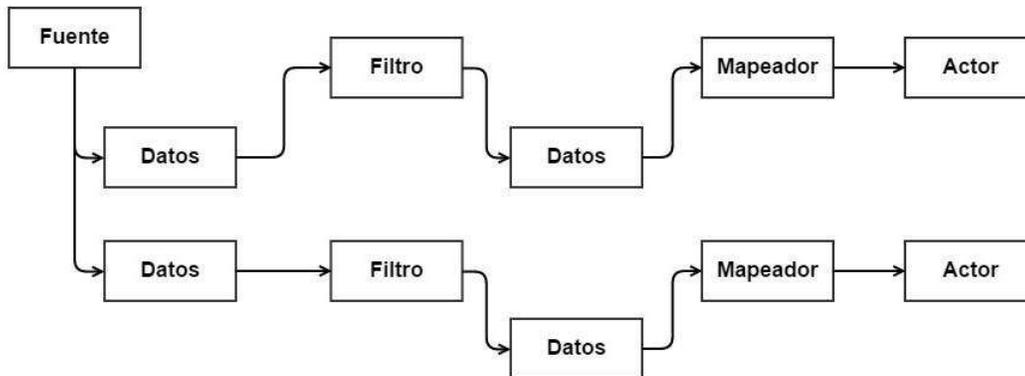
En VTK existen dos componentes de diseño esenciales: el modelo gráfico y el modelo de visualización. El modelo gráfico básicamente es un modelo abstracto para los gráficos tridimensionales basado en una interfaz GUI. Existen 9 objetos básicos que conforman este modelo, a continuación se describe cada uno de estos (Schroeder, Martin, & Lorensen, 1996):

1. **Maestro de renderizado:** Elemento que se encarga de coordinar los dispositivos y crear las ventanas de renderizado.
2. **Ventana de renderizado:** Objeto encargado de manejar la ventana que se encuentra en el monitor del dispositivo. Uno o más renderizados pueden estar presentes en una ventana de renderizado, esto produce una escena.
3. **Render:** Coordina las luces renderizadas, las cámaras y los actores.
4. **Luz:** Elemento que ilumina la escena.
5. **Cámara:** Objeto que define la posición de la vista, el punto focal y otras características de la cámara.
6. **Actor:** Objeto creado por un render dentro de una escena.
7. **Propiedad:** Representa los atributos renderizados de un actor como color, textura, sombra, etc.
8. **Mapeador:** Representa la definición geométrica de un actor y mapea el objeto a través de una tabla de búsqueda.
9. **Transformador:** Objeto que está conformado por una matriz 4x4 y métodos para modificar dicha matriz.

El modelo de visualización está conformado por dos tipos de objetos: objetos de datos y objetos de procesos. El primero, representa la información como un campo, de este tipo de objeto heredan los conjuntos de datos. Estos contienen una estructura topológica y una geométrica, además están asociadas a un conjunto de atributos que los definen: escalares, vectores, normales, coordenadas de textura y datos del campo.

El segundo, el objeto de procesos, refiere a todas las operaciones que se pueden realizar sobre los objetos de datos. La forma en cómo interactúan los objetos de procesos y los de datos es mediante una arquitectura tubo-filtro, idéntica a la ITK, donde cada tubo es un objeto de datos y cada filtro uno de procesos. Esta arquitectura puede elaborarse según las necesidades del

programador, y puede tener ramificaciones y ciclos en cada una de ellas. En la siguiente figura se puede apreciar la arquitectura típica de VTK.



**Figura 4. Arquitectura típica de VTK (Tomada de (Brown & Wilson, n.d., n.d.; Johnson et al., 2014))**

Existen diferentes objetos de procesos y se pueden agrupar en tres tipos: fuentes, filtros y mapeadores. Las fuentes no tienen ningún dato de entrada, pero producen una o más salidas; los filtros aceptan una o más entradas y producen una o más salidas; y los mapeadores, son los que finalizan la arquitectura, ya sea agrupando las gráficas del subsistema o guardándolas en disco.

VTK ofrece grandes ventajas en relación a software similares a este, como lo son: Open Inventor, Hoops 3D y Coin 3D. VTK está enfocado a la investigación y a universidades, además es gratuito y de código abierto, sin mencionar que ofrece una amplia documentación a través de libros y guías de usuario. De igual manera, posee una larga lista de características de visualización que no requieren compra de licencias adicionales como el caso de Open Inventor (Laramee, 2008). Finalmente, la gran ventaja que tiene VTK es la facilidad de integración con la librería ITK, la cual es indispensable para el desarrollo del prototipo.

### 2.1.5. Interacción natural de usuario

Como se pudo apreciar en la [sección 2.1.2](#), ningún algoritmo desarrollado hasta el momento ha podido suplir la necesidad de generar modelos de pulmón totalmente acertados. Debido a esto, se propuso una alternativa en la que el médico puede disminuir el error manualmente al

interactuar con una máquina. Existe una disciplina encargada del diseño, evaluación e implementación de sistemas de cómputo con los que las personas pueden interactuar, denominada interacción hombre máquina (HCI por sus siglas en inglés).

El objetivo principal de esta disciplina es facilitar la transferencia de información entre el usuario y el computador. Para ello se deben tener en cuenta dos aspectos: la funcionalidad y usabilidad. La funcionalidad hace referencia al conjunto de actividades que el usuario puede realizar al interactuar con el computador; mientras que la usabilidad refiere a la facilidad con la que se realizan estas actividades (Mathew, Al Hajj, & Al Abri, 2011).

Uno de los retos más grandes que afronta la disciplina de HCI se encuentra en el área de la medicina, puesto que los enfoques tradicionales del primero no suplen en su totalidad las necesidades de los médicos. La interacción debe diseñarse de tal manera que se pueda adaptar a la rutina diaria y al entorno de los expertos, para que se desempeñen en su labor de una manera eficiente y efectiva (Thimbleby, 2007).

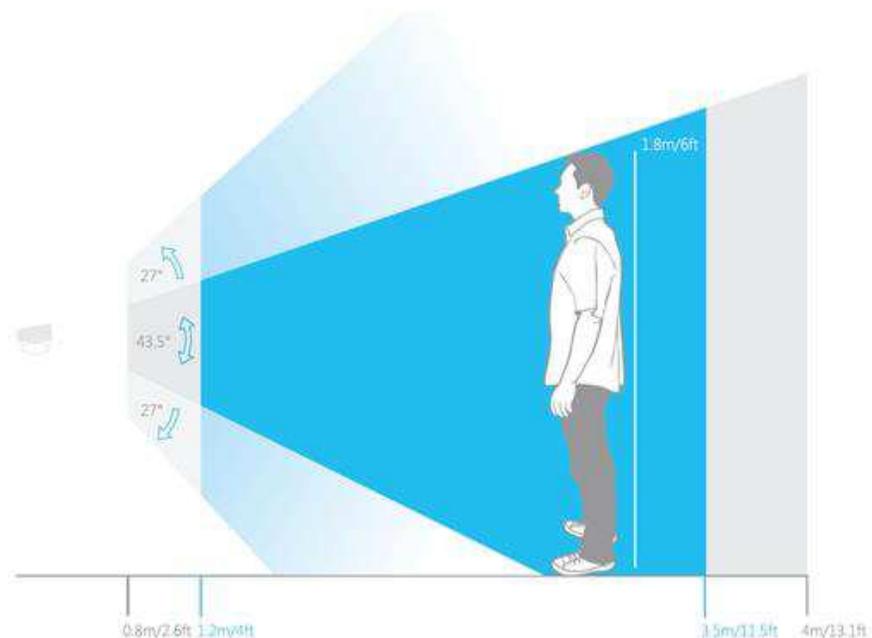
En HCI, WIMP es uno de estos enfoques de interacción tradicionales basado en el uso de ventanas, iconos, menús y punteros. Actualmente, es el estilo de interacción más conocido y utilizado, esto se debe a que es la interfaz por defecto de la mayoría de sistemas de computación interactiva. Aunque el estilo de interacción WIMP es muy utilizado hoy en día, no brinda necesariamente un esquema ni un modo de interacción sencillo para el usuario. WIMP en ocasiones convierte tareas sencillas en tareas dispendiosas y puede dificultar aún más aquellas tareas que son complicadas por sí solas (Kumar, 2005). En relación a la corrección de la segmentación del pulmón, WIMP podría utilizarse como método para solucionar los errores que se presentan en el modelo, sin embargo, no resultaría ser el más adecuado.

Un médico necesita un modo de interacción sencillo de utilizar, que no requiera un largo período de aprendizaje. En este orden de ideas, una técnica complementaria que permitiría solucionar el problema de la segmentación es la interfaz natural de usuario (NUI por sus siglas en inglés): NUI es un estilo de HCI basado en el análisis automático del comportamiento natural del usuario, estas acciones son interpretadas por la máquina como comandos que controlan diversas operaciones. Es importante denotar que el comportamiento natural es el conjunto de actividades que las personas realizan en su vida diaria, con el objetivo de interactuar con los

objetos animados e inanimados de su entorno. Dado que NUI soporta una interacción intuitiva, el aprendizaje es más rápido y fácil en comparación a los métodos tradicionales (Ogiela & Hachaj, 2015).

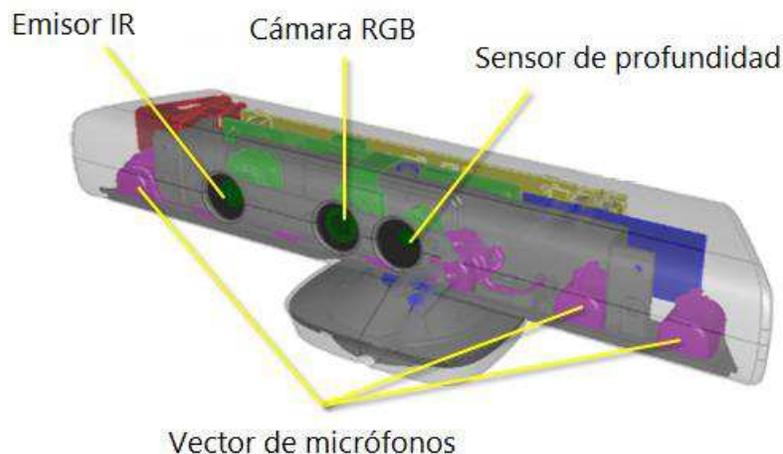
Por lo tanto, resulta más sencillo para un médico interactuar con los equipos de cómputo si se encuentra familiarizado con el tipo de acciones que tiene que realizar. Tampoco hay que afirmar que se deben descartar por completo los métodos tradicionales de interacción como lo es WIMP, ya que las personas se encuentran acostumbradas a algunas características de este estilo: es más sencillo representar una acción o un objeto mediante un ícono que lo refleje en el mundo real, a representarlos mediante palabras (van den Boomen, 2009).

Actualmente existe una amplia gama de tecnologías en el mercado basadas en NUI, que podrían emplearse para mejorar el error de segmentación mencionado en la [sección 2.1.2](#). Sin embargo, la tecnología que más se adecúa a las necesidades de los médicos es el Kinect de Microsoft, ya que no se requiere tener contacto físico con un control. En la siguiente figura se muestra la forma en la que se interactúa con el Kinect junto con el campo visual vertical y horizontal del mismo.



**Figura 5. Esquema de Kinect.** Recuperado de <https://msdn.microsoft.com/es-co/library/hh973074.aspx>

El Kinect de Microsoft cuenta con diferentes capacidades que permiten la corrección del modelo del pulmón, dentro de ellas se encuentran el reconocimiento de gestos. Para realizar estas tareas, Kinect está compuesto por diferentes sensores que capturan información y que luego es enviada al procesador. Estos sensores son: una cámara RGB, un sensor de profundidad y cuatro micrófonos (ver figura 6) (Ladzinski, 2012).



**Figura 6. Partes del Kinect.** Recuperado de <https://msdn.microsoft.com/es-co/library/jj131033.aspx>

La cámara RGB tiene una resolución de 640x480 píxeles, una distancia focal de 525 píxeles y campo visual horizontal de 62.7 grados. Esta cámara es utilizada principalmente para brindar una retroalimentación al usuario (Ladzinski, 2012).

El sensor de profundidad está conformado por dos elementos: un láser que proyecta luz infrarroja y una cámara que obtiene la información arrojada por dicha luz. La cámara de este sensor posee una resolución de 640x480 píxeles, y una sensibilidad de 11 bits por píxel, lo cual genera 2045 niveles de resolución individual. Así mismo es capaz de realizar la lectura de cada fotograma a 30 Hz. Este sensor principalmente se encarga de reconocer a las personas y seguir sus acciones (Ladzinski, 2012).

Los cuatro micrófonos pueden grabar sonidos en muy alta calidad, que es necesaria para realizar el reconocimiento de voz. Además, cuentan con la funcionalidad de reconocer la ubicación de la fuente comparando la intensidad del sonido en cada uno de los micrófonos y utilizando métodos de triangulación, formalmente este método se conoce como formación de haz (*beamforming*) (Ladzinski, 2012).

Por otro lado, Microsoft ofrece un SDK para Windows, una librería que facilita el desarrollo de aplicaciones para Kinect. Esta librería permite construir aplicaciones en C#, VB.NET y C++. Como se mencionó previamente, una de las capacidades de Kinect es el reconocimiento de gestos, esto se logra a través de la creación de un esqueleto que representa a la persona que se encuentra frente al dispositivo. El SDK de Kinect se encarga de llevar a cabo el proceso de detección del esqueleto facilitando el trabajo del desarrollador, para ello emplea el sensor de profundidad (Jana, 2012).

El esqueleto está conformado por veinte puntos que representan las diferentes partes del cuerpo, como la cabeza, los hombros, las rodillas, entre otras (ver figura 7). A cada punto se le asigna una coordenada en tres dimensiones, esto permite determinar cambios en el espacio en cada uno de ellos y finalmente concluir si se realizó un gesto o no (Zhang, 2012).



**Figura 7. Esqueleto detectado por Kinect.** Recuperado de <https://msdn.microsoft.com/es-co/library/hh973074.aspx>

Como se puede evidenciar Kinect posee las suficientes características para brindar a un médico la capacidad de interactuar con modelos tridimensionales, como el del pulmón, de una manera muy precisa. Así mismo, cabe recalcar que es importante para los hospitales utilizar tecnologías que no sean de alto costo Kinect cumple con esta premisa si se compara con tecnologías médicas ya existentes basadas en NUI.

Debido a lo mencionado a lo largo de esta sección, en los últimos años Kinect ha comenzado a introducirse en el campo de la medicina con el objetivo de facilitar la manipulación de imágenes clínicas como tomografías y resonancias magnéticas. A continuación, se presenta un trabajo en el cual se emplea el Kinect para la manipulación y deformación de una masa, y dos trabajos en donde se utiliza el Kinect específicamente en la manipulación de imágenes médicas.

En el trabajo *Evaluating organic 3D sculpting using Natural User Interfaces with the Kinect* (Wesson & Wilkinson, 2013) se empleó el Kinect para modificar una masa en el espacio. Además de la detección de gestos se utilizó el reconocimiento de voz que provee el dispositivo, sin embargo, este reconocimiento no funcionó como se esperaba puesto que muchas acciones se tenían que repetir varias veces para ser detectadas.

Al utilizar el Kinect se esperaría enfocarse únicamente en la detección de gestos, puesto que es la característica más robusta del dispositivo. Al añadir el reconocimiento de voz se incrementa la complejidad en el desarrollo de tareas debido a la falta de precisión que Kinect presenta en este sensor.

Además, dicho trabajo propone el reconocimiento de las manos y de los dedos para realizar ciertas acciones como, por ejemplo, agarrar algún elemento de la interfaz gráfica. No obstante, los autores hacen énfasis en la dificultad que tuvieron las personas al utilizar esta característica, esto causado por la baja precisión que se tiene sobre esta parte del cuerpo. Por ende, recomiendan utilizar guantes de datos para estas acciones por su precisión, sin embargo, para para el mejoramiento de modelos tridimensionales del pulmón resultaba intrusivo utilizar esta

tecnología: Se pretendía que el médico estuviese libre de dispositivos físicos en la manipulación de imágenes.

El primer trabajo de investigación relacionado con la manipulación de imágenes médicas que se abordará es dado a conocer en el artículo *Natural Interactive 3D Medical Image Viewer Based on Finger and Arm Gestures* (Tuntakurn, Thongvigitmanee, Sa-Ing, Hasegawa, & Makhanov, 2013). En este trabajo se propone un software de visualización de imágenes médicas tridimensionales con una interacción natural basada en el Kinect de Microsoft. Este software fue desarrollado en C++ e hizo uso de la librería VTK, además empleó el SDK de Kinect para la detección de gestos con los brazos, y finalmente se desarrolló un algoritmo para la detección de gestos con los dedos y las manos.

Para el uso del software se fijaron siete gestos básicos: rotar, enfocar/desenfocar, desplazamiento sobre la imagen, seleccionar corte coronal, seleccionar corte axial, seleccionar corte sagital y cambiar corte. Cada gesto fue sometido a pruebas de precisión y de rendimiento. En la prueba de precisión se le pidió a un grupo de diez personas realizar cada uno de los gestos diez veces. Para la prueba de rendimiento se le solicitó al mismo grupo realizar los siete gestos para interactuar con el software, dos semanas después las personas realizaron esta misma tarea. El rendimiento se determinó comparando el tiempo que tardaron en ejecutar las acciones en las dos sesiones.

Tras las pruebas se concluyó que todos los gestos se realizaron en promedio con una precisión del 93%. Los gestos en los que se presentaron menor precisión fueron los de selección de cortes, sin embargo, esto pudo ser debido a que el sistema era inestable cuando las manos permanecían muy cerca al cuerpo. Finalmente, en cuanto al rendimiento, se evidenció que el tiempo necesario para realizar los siete gestos disminuyó en la segunda sesión de la prueba, lo cual puede indicar que los gestos eran claros y sencillos.

El segundo trabajo de investigación se encuentra en el artículo *Controller-free exploration of medical image data: experiencing the Kinect* (Gallo, Placitelli, & Ciampi, 2011), los autores desarrollaron un sistema que permitía a un cirujano manipular imágenes TAC, de resonancia magnética (IRM) y de tomografías por emisión de positrones (PET). El código fue escrito completamente en C++, junto con OpenCV y MITO para el procesamiento gráfico.

En este software era posible navegar a través de diversas imágenes médicas e inclusive examinar una pequeña área de interés, más no un órgano en particular. No obstante, no deja de ser una tarea tediosa examinar imagen por imagen para determinar cuál de todas es la determinante en la cirugía. Cabe recalcar que los autores desarrollaron nueve gestos para suplir las funcionalidades del software, sin embargo, debido al exceso de funcionalidades, los gestos debían diferenciarse mucho el uno del otro para evitar ambigüedad en la detección.

Del primer trabajo que utiliza Kinect para la manipulación de imágenes médicas se puede concluir que el dispositivo es una herramienta adecuada para el área de la medicina por su facilidad de uso. Por otra parte, el segundo trabajo deja una lección muy importante: un software que emplea una interacción natural de usuario no debe ser muy robusto en cuanto a funcionalidades, ya que conlleva a que los gestos sean difíciles y poco intuitivos.

## **2.2. Justificación**

Un mal diagnóstico sobre el pulmón puede poner en riesgo la vida de un paciente, al ser un órgano vital para el funcionamiento del cuerpo humano. Es por esto que un médico requiere una imagen médica clara y precisa del pulmón para detectar la presencia de anomalías (Beutel, Kundel, & Van Metter, 2000).

El error de la imagen tridimensional del pulmón puede alterar el diagnóstico, esto debido a que los algoritmos de segmentación presentan una gran dificultad a la hora de discernir entre lo que es o no tejido pulmonar: En ocasiones el modelo del pulmón resulta incompleto o con partes de otros órganos, lo que impide un buen diagnóstico (Brant & Helms, 2012).

Para realizar el mejoramiento de la segmentación del pulmón, el médico debe interactuar con una máquina que le permita eliminar secciones sobrantes, o, adicionar partes del pulmón que pudieron haber sido eliminadas por los algoritmos. Como se mencionó en la [sección 2.1.5](#), existen diversos estilos de interacción. WIMP, puede ser una solución para este problema, pero para un médico esta interacción puede considerarse compleja.

Por el contrario, la interacción natural le brindaría al médico la capacidad de manipular y disminuir el error del pulmón de una forma sencilla y eficiente. La curva de aprendizaje para

este estilo es más corta porque el modo de interacción es intuitivo. Adicionalmente la interfaz gráfica resulta minimalista debido a que no se requiere el uso de componentes como menús, que imponen complejidad sobre el manejo de la herramienta.

Actualmente, solo algunos hospitales en el mundo cuentan con software basado en la interacción natural para actividades como la visualización de imágenes médicas. Esto se debe a que dichas aplicaciones requieren de sensores muy costosos como cámaras estereoscópicas (Kipshagen, Graw, Tronnier, Bonsanto, & Hofmann, 2009).

Sin embargo, hoy en día ya existe una larga lista de sensores que permiten llevar a cabo las mismas funcionalidades y son de muy bajo costo. En la tabla 1 se describen los aspectos más relevantes de los sensores de bajo costo que se consideraron para el desarrollo de este proyecto.

Como se puede observar el dispositivo Kinect posee características que lo hace el más apropiado para el desarrollo del proyecto, algunas de estas son:

- Kinect posee diferentes sensores que permiten una captura de información muy detallada en cuanto a los gestos que se realizan.
- El usuario no está atado a un controlador, en otras palabras, no debe tener contacto directo con un dispositivo para transmitir la información de los movimientos. Esto es un aspecto muy importante, ya que en el área de la medicina tener contacto físico con un dispositivo puede afectar procesos médicos en donde la esterilidad es primordial. (Tuntakurn, Thongvigitmanee, Sa-Ing, Hasegawa, & Makhanov, 2013).
- El costo del sensor es muy bajo en relación a sus funcionalidades (Villaroman, Rowe, & Swan, 2011).

Además, vale la pena destacar que actualmente existen varias empresas y trabajos de investigación que han integrado la tecnología de Kinect a software de visualización de imágenes médicas ("TEDCAS | Natural user interfaces for healthcare", 2016). Esto nos lleva a concluir que muchos expertos en el área de la medicina ya están familiarizados con esta tecnología. Sin embargo, aún no se ha empleado la tecnología de Kinect para el mejoramiento del modelo tridimensional de pulmón, lo cual abre paso a una gran oportunidad en el campo de la investigación.

Nombre	Costo (USD)	Sensor	Contacto	Detección	Rango de uso
<b>Kinect</b>	\$189.99	<ul style="list-style-type: none"> <li>• Cámara RGB</li> <li>• Sensor de profundidad</li> <li>• Micrófonos</li> </ul>	No	<ul style="list-style-type: none"> <li>• Detección de cuerpo (Esqueleto)</li> <li>• Detección facial</li> <li>• Reconocimiento de voz</li> </ul>	Hasta 4 metros
<b>Guantes de datos 5DT</b>	\$1,000.00 - \$5,000.00	<ul style="list-style-type: none"> <li>• Acelerómetro</li> </ul>	Si	<ul style="list-style-type: none"> <li>• Detección de flexibilidad y abducción entre dedos</li> </ul>	No aplica
<b>Wiimote</b>	\$189.99	<ul style="list-style-type: none"> <li>• Acelerómetro</li> <li>• Sensor óptico</li> <li>• Brújula electrónica</li> </ul>	Si	<ul style="list-style-type: none"> <li>• Aceleración del dispositivo a lo largo de los tres ejes</li> <li>• Ubicación del dispositivo en los tres ejes</li> </ul>	Hasta 10 metros
<b>Leap motion</b>	\$48.88	<ul style="list-style-type: none"> <li>• Sensor de profundidad</li> </ul>	No	<ul style="list-style-type: none"> <li>• Detección de dedos</li> <li>• Detección de muñeca</li> </ul>	Entre 25-600 milímetros
<b>MYO</b>	\$199.99	<ul style="list-style-type: none"> <li>• Sensores de electromiografía</li> <li>• Sensor IMU</li> </ul>	Si	<ul style="list-style-type: none"> <li>• Detección de actividad eléctrica de los músculos (5 gestos)</li> <li>• Detección de movimiento, orientación y rotación</li> </ul>	Hasta 15 metros
<b>Phantom</b>	\$1,000.00	<ul style="list-style-type: none"> <li>• Sensor de presión</li> </ul>	Si	<ul style="list-style-type: none"> <li>• Presión</li> </ul>	No aplica

**Tabla 1. Sensores de interacción natural**

Con lo expuesto previamente, se plantea un esquema de interacción que facilite la manipulación del modelo del pulmón en tres dimensiones. Lo cual conlleva a que la interacción se aproxime a la de un objeto en el mundo real. Esta situación permitiría al médico examinar y mejorar el modelo del pulmón de una forma sencilla.

Adicionalmente vale la pena destacar que, en el prototipo desarrollado para sustentar la hipótesis de este proyecto de investigación, no se presenta un modelo tridimensional de un pulmón sino un modelo tridimensional de una esfera. Esto es debido a que los dos modelos son homotópicos, es decir que uno puede deformarse continuamente hasta convertirse en el otro. En otras palabras, ambos modelos resultan ser topológicamente equivalentes (Crossley, 2005).

## III - DESCRIPCIÓN DE LA SOLUCIÓN

En las siguientes secciones se puede observar el resultado de la segunda y tercera fase de la metodología empleada para el desarrollo del proyecto.

Las secciones [3.2](#) y [3.3](#) corresponden al desarrollo de la segunda fase metodológica, en estas se presentan las funcionalidades del prototipo derivadas de las historias de usuario y el diseño del mismo.

Desde la sección [3.4](#) hasta la sección [3.7](#) se presenta el desarrollo de la tercera fase metodológica, a excepción de las pruebas. La sección [3.4](#) corresponde a las iteraciones de diseño de historias de usuario e implementación de gestos. La sección [3.5](#) corresponde a la iteración de implementación VTK. La sección [3.6](#) refiere a la iteración de implementación ITK. Por último, la sección [3.7](#) describe la iteración de integración ITK, VTK y gestos.

### 3.1. Limitaciones y dependencias del prototipo

La principal limitación del prototipo desarrollado es que el usuario debe contar con sus dos extremidades superiores y estar de pie. De igual manera para que el prototipo funcione correctamente, el dispositivo Kinect debe ubicarse al menos a 2 metros de distancia del usuario, y a una altura entre 0,6 y 2 metros del suelo.

Por otra parte, una vez el sistema final sea desarrollado, se debe suponer que el usuario es un experto en el área de la medicina que conoce la estructura correcta de modelos tridimensionales del pulmón.

Vale la pena destacar que el prototipo puede ejecutarse únicamente sobre el sistema operativo Windows 7 o versiones superiores. Así mismo, para su funcionamiento, este depende del SDK de Kinect versión 1.8 y de las librerías ITK y VTK.

## **3.2. Funcionalidades del prototipo**

Para llevar a cabo la interacción con el modelo tridimensional se definieron 4 funcionalidades básicas con las que el prototipo debía contar, las cuales se definieron a partir de las historias de usuario. Estas son presentadas a continuación.

### **3.2.1. Acercar y alejar**

El prototipo permite realizar la acción de acercar y alejar el modelo tridimensional en pantalla, para que el médico pueda ser capaz de analizar un área específica y así deformar con mayor precisión aquellas áreas que presenten errores.

### **3.2.2. Trasladar**

El prototipo permite trasladar el modelo tridimensional de manera horizontal y vertical para ajustarlo a la pantalla

### **3.2.3. Rotar**

Es necesario observar el modelo tridimensional desde diferentes ángulos, es por esto que el prototipo permite rotar el modelo vertical y horizontalmente.

### **3.2.4. Deformar**

Esta es la acción más importante del prototipo, dado que el propósito de la investigación es verificar si Kinect es la herramienta más adecuada para que un médico pueda mejorar el modelo del pulmón. La funcionalidad de deformar, entonces, permite modificar la estructura topológica del modelo.

## **3.3. Diseño**

### **3.3.1. Diseño del prototipo**

El diseño que se utilizó en el prototipo está basado en el patrón arquitectural Modelo-Vista-Controlador (MVC), esto con el fin de que la aplicación pueda ser implementada a futuro en diferentes plataformas o utilizando otras librerías. En el caso de Kinect se esperaría que en

algún momento se desarrolle una herramienta que permita la detección de gestos, no sólo en Windows sino en otros sistemas operativos como Linux. En el caso de la visualización, en lugar de utilizar VTK se podría emplear OGRE, que es una librería diseñada para la visualización tridimensional. Y en el caso ITK, emplear otras librerías de segmentación como MITO, que fue utilizada en trabajos relacionados ([ver sección 2.1.5](#)).

En la figura 8 se presentan los diferentes componentes que conforman el patrón MVC en el prototipo.

Los componentes están definidos de la siguiente manera: ***Kinect*** es el controlador, ***Deformación*** el modelo y ***Visualización*** la visualización. ***Kinect*** está encargado de recibir y procesar los datos capturados a través del dispositivo Kinect y decidir qué gesto está realizando el usuario para delegar a la ***Visualización*** o la ***Deformación***, según corresponda.

***Kinect*** se comunica con ***Deformación*** para realizar cambios topológicos sobre el modelo tridimensional, para ello envía las coordenadas (X, Y y Z) que representan el punto en el que se debe deformar el modelo. Por otro lado, ***Visualización*** está encargada de realizar las transformaciones geométricas del modelo tridimensional, para ello ***Kinect*** invoca los diferentes métodos que posee ***Visualización***, como por ejemplo: *moverHorizontal()*, *rotarVertical()*, entre otros.

Adicionalmente, ***Kinect*** utiliza la clase auxiliar ***ManejadorGestos*** para conocer el estado en el que se encuentra cada gesto, y la enumeración ***GESTO*** para definir el gesto que se está llevando a cabo en un momento determinado. Por otra parte, ***Deformación*** hace uso de la clase ***ITKaVTK***, que está encargada de la transformación de datos entre ITK y VTK.

En las secciones [3.4](#), [3.5](#), [3.6](#) y [3.7](#) se describe con mayor detalle la comunicación entre los componentes mencionados anteriormente.

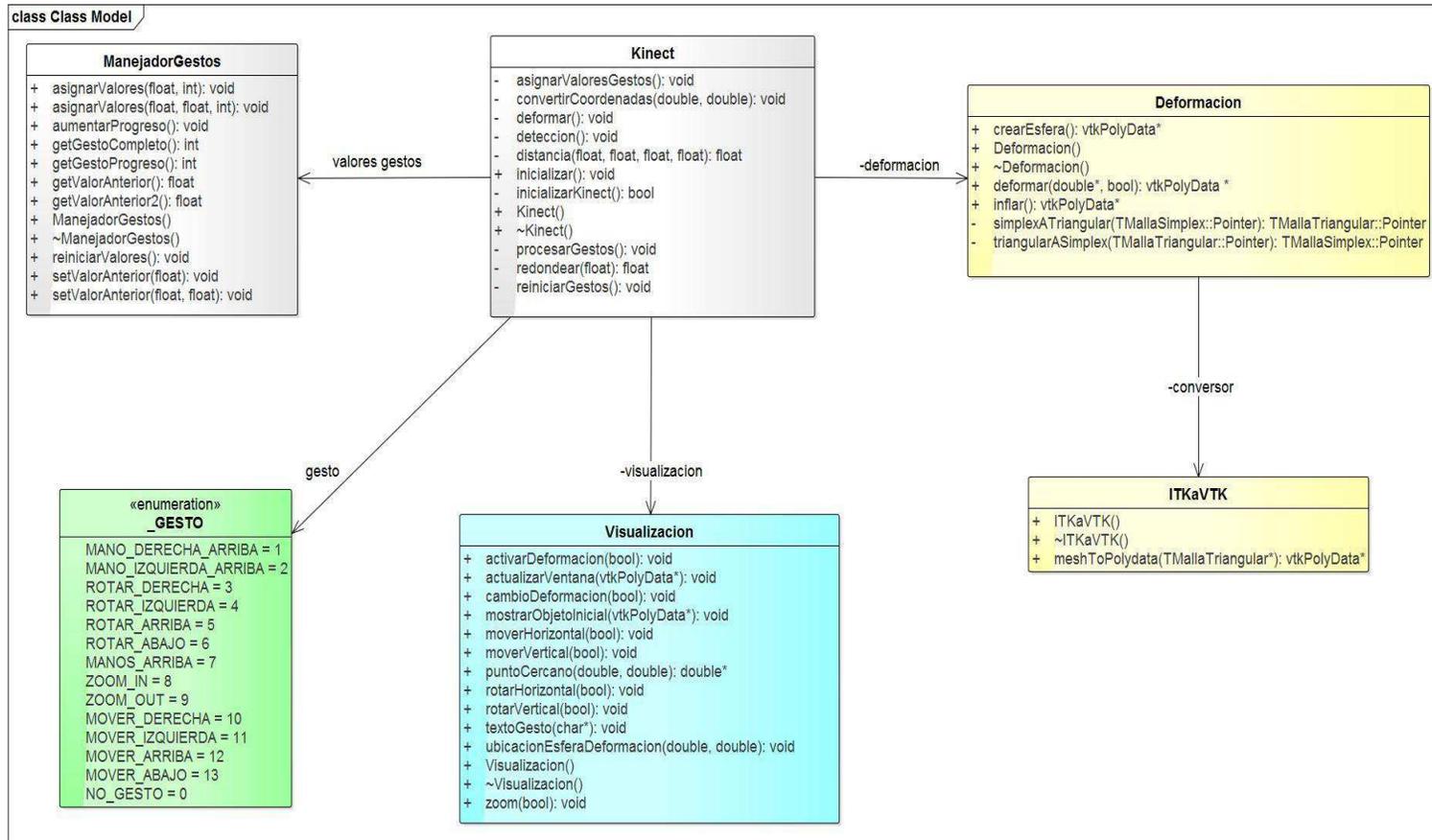


Figura 8. Diseño del prototipo

## **3.4. Desarrollo de gestos**

La detección de gestos es el componente más importante del prototipo desarrollado. Para lograr esto, se hizo uso del dispositivo Kinect de Microsoft y su respectivo SDK en la versión 1.8 ([ver sección 2.1.5](#)).

En primera instancia, el prototipo accede al sensor de profundidad del Kinect. A partir de esto el SDK toma las coordenadas de los pixeles de la imagen de profundidad y las mapea a un esqueleto, que representa al usuario. Dicho esqueleto está conformado por veinte nodos que representan las diferentes partes del cuerpo, como lo son la cabeza, las manos, los codos entre otros. Cada uno de estos puntos posee una coordenada en el espacio (X, Y y Z), la cual es dada a conocer por el Kinect 30 veces por segundo.

Así pues, la detección de gestos se hace posible al detectar la posición de algún nodo por un tiempo determinado (fotogramas por segundo). La detección de los gestos se dividió en dos fases: la precondition y la acción. La primera hace referencia al gesto inicial que el usuario debe realizar para poder empezar una acción. Y la segunda, refiere al comando que se quiere ejecutar sobre el prototipo.

En las siguientes subsecciones se describe de manera detallada como se logró la detección de cada uno de los gestos.

### **3.4.1. Alejar y acercar**

#### **Precondición**

Para dar inicio al gesto de acercar y alejar, el usuario debe levantar la mano izquierda y posteriormente la mano derecha, estas deben permanecer levantadas. Se estableció que una mano se encuentra levantada cuando su coordenada en Y es superior a la coordenada en Y del nodo cadera centro.

**Acción**

Para realizar la acción de alejar el usuario debe acercar las manos. La distancia en  $X$  entre los nodos de las manos debe disminuir a medida que transcurre el tiempo para reconocer este gesto.

Para realizar la acción de acercar, el usuario debe alejar las manos, este gesto de igual manera se reconoce a través de la distancia entre los nodos, pero en este caso se determina si la distancia va aumentando.

La decisión de asociar estos gestos a las funcionalidades de acercar y alejar, se basó en la forma en la cual se realizan estas acciones actualmente en los dispositivos móviles táctiles, así como en el trabajo relacionado *Controller-free exploration of medical image data: experiencing the Kinect* (Gallo, Placitelli, & Ciampi, 2011).

Diagrama de estados

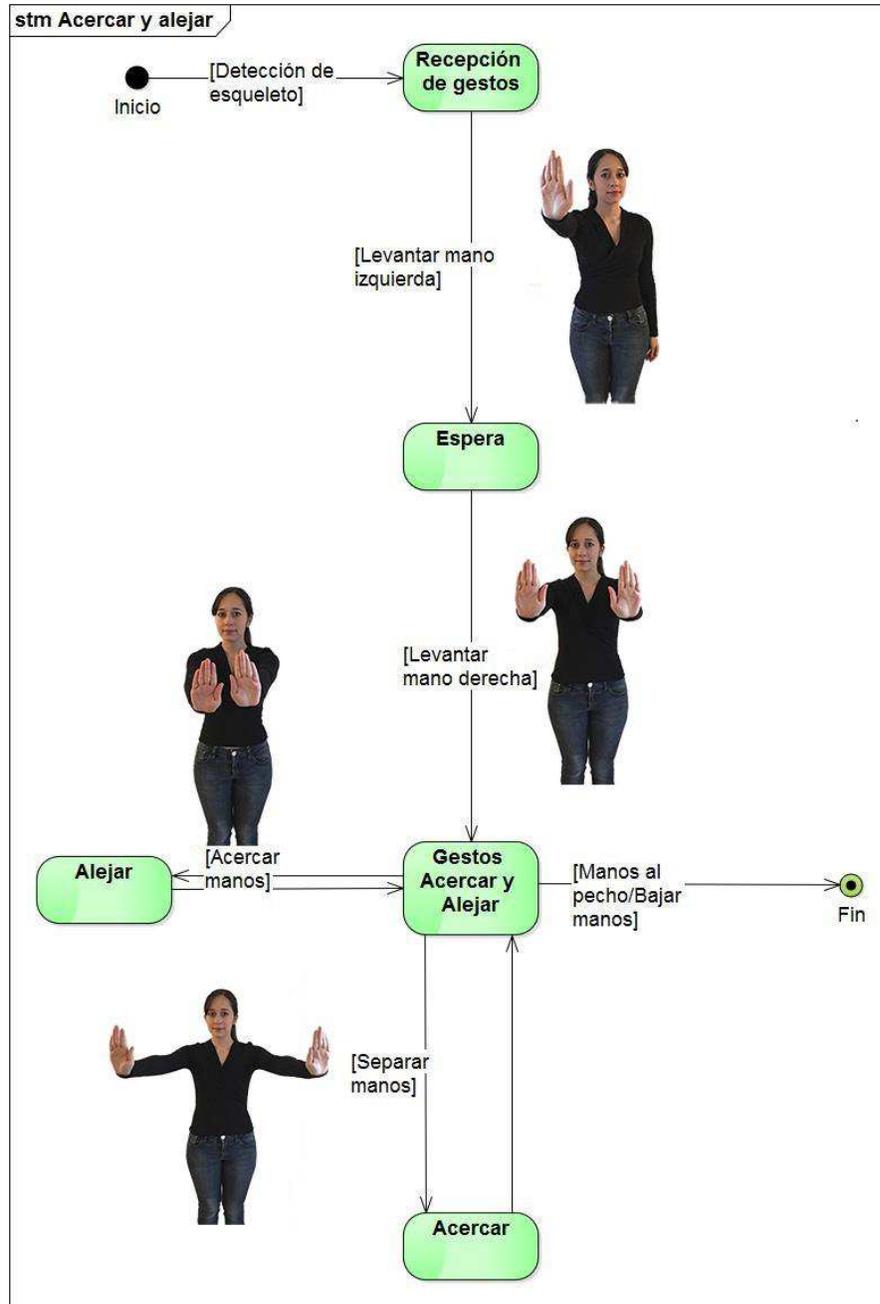


Figura 9. Diagrama de estados acercar y alejar

### **3.4.2. Trasladar**

#### **Precondición**

Para comenzar el gesto de trasladar el usuario debe levantar y ubicar sus dos manos en frente de su cuerpo y mantenerlas en esa posición.

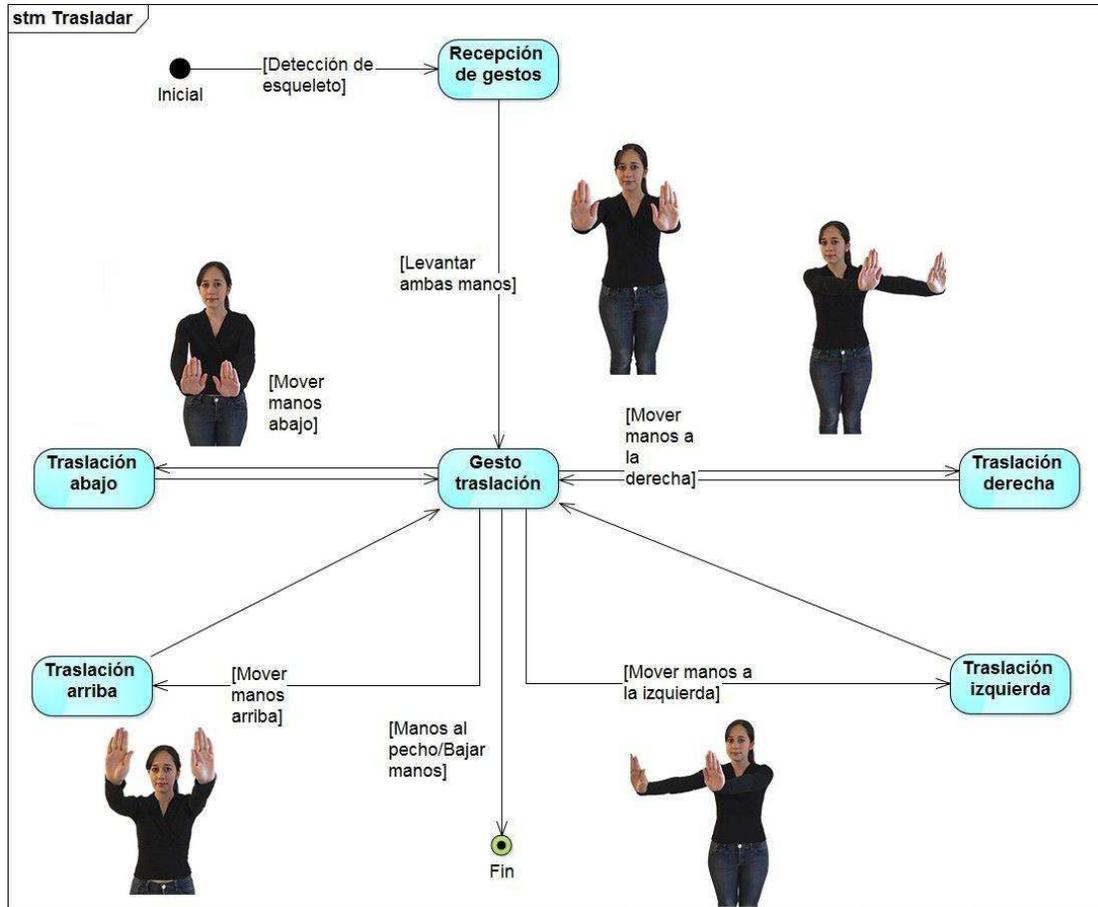
#### **Acción**

El usuario puede trasladar el modelo tridimensional de manera horizontal o vertical.

Para realizar una traslación horizontal, el usuario debe mover sus manos a la derecha o a la izquierda. Este gesto se detecta al determinar si las coordenadas en X de las dos manos aumentan o disminuyen en el transcurso del tiempo.

En la traslación vertical el usuario debe mover sus manos hacia arriba o hacia abajo. El gesto se detecta de la misma manera que la traslación horizontal, pero usando como referencia las coordenadas en Y.

**Diagrama de estados**



**Figura 10. Diagrama de estados trasladar**

**3.4.3. Rotar**

**Precondición**

El gesto de rotar inicia cuando el usuario levanta su mano izquierda y la ubica frente a su cuerpo.

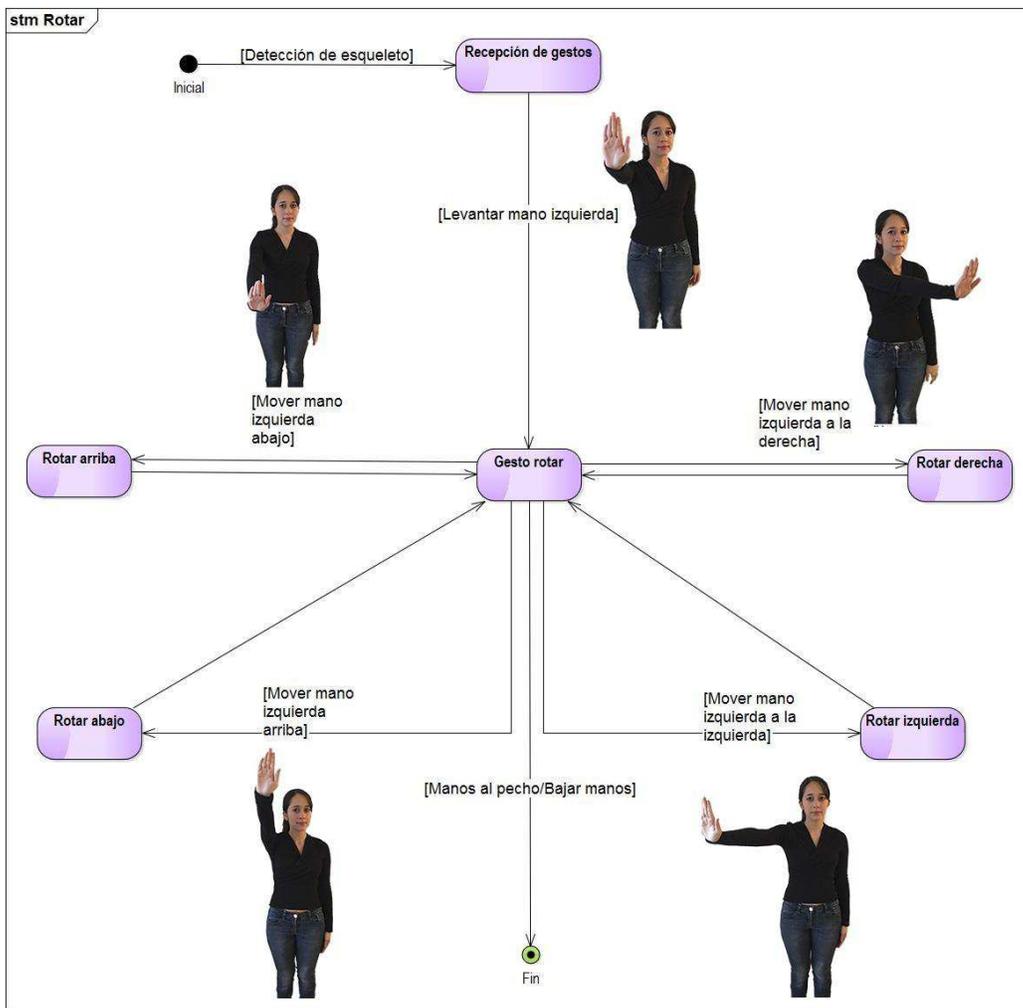
**Acción**

El usuario puede rotar el modelo tridimensional de manera horizontal o vertical.

La rotación horizontal se logra cuando el usuario mueve su mano izquierda a la derecha o a la izquierda. Este gesto se detecta al determinar si la coordenada en X de la mano aumenta o disminuye en el transcurso del tiempo.

Para realizar una rotación vertical, el usuario debe mover su mano izquierda hacia arriba o hacia abajo. El gesto se detecta al determinar si la coordenada en Y de la mano aumenta o disminuye en el transcurso del tiempo.

**Diagrama de estados**



**Figura 11. Diagrama de estados rotar**

### **3.4.4. Deformar**

#### **Precondición**

El gesto de deformar inicia cuando el usuario levanta su mano derecha y la mantiene en esa posición.

#### **Acción**

El usuario puede deformar el modelo de dos modos: atraer o repeler.

Para realizar la acción de deformar con el modo de repeler, el usuario debe mover su mano derecha a través del espacio, este movimiento se verá reflejado en la deformación del modelo tridimensional. Esto se determina al hallar constantemente las coordenadas en X y en Y de la mano derecha.

Para realizar la acción de deformar con el modo de atraer, el usuario debe adicionalmente levantar la mano izquierda y posteriormente mover su mano derecha a través del espacio.

Diagrama de estados

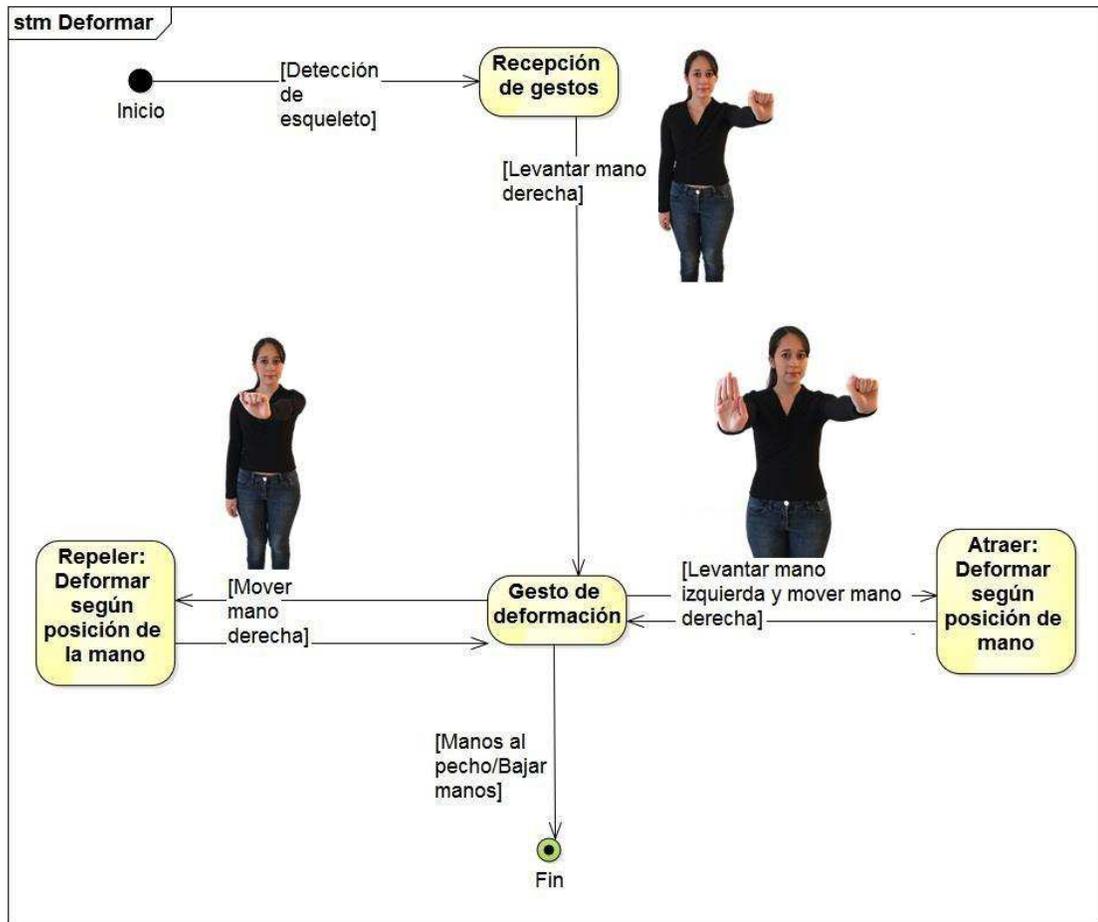


Figura 12. Diagrama de estados deformar

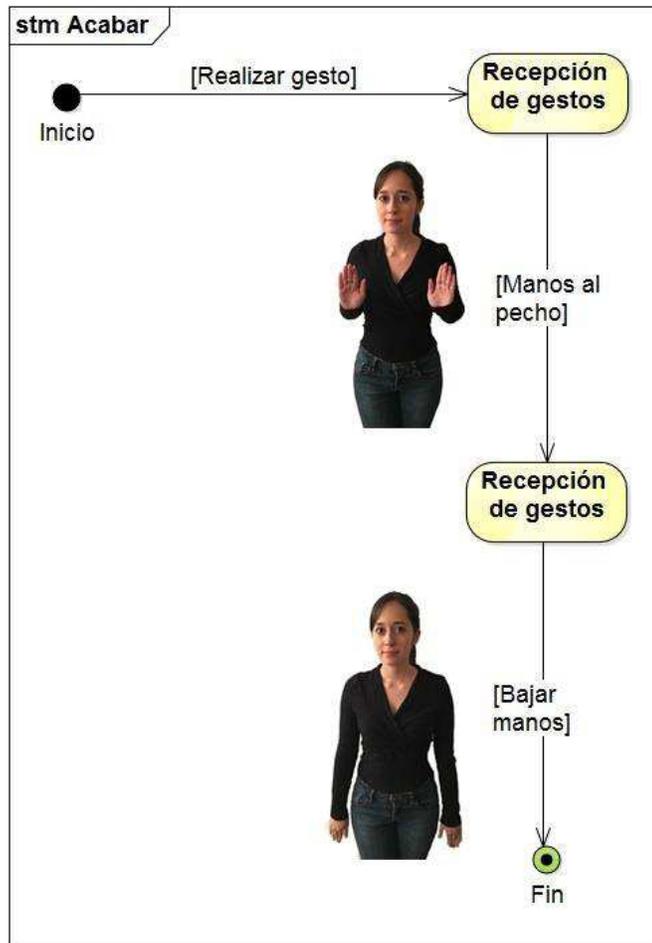
3.4.5. Finalizar gesto

Precondición

Para realizar este gesto el usuario debe estar realizando alguno de los gestos presentados anteriormente.

**Acción**

Para realizar la acción de finalizar gesto el usuario debe acercar sus manos al cuerpo y posteriormente bajarlas. Para detectar este gesto se comparan las coordenadas en Z de las manos y del nodo hombro centro. Después se comparan las coordenadas en Y de las manos con las del nodo cadera centro.



**Figura 13. Finalizar gesto**

### 3.4.6. Estado neutro

El gesto de estado neutro define un estado de inactividad, es decir que no se está realizando alguna funcionalidad del prototipo. Para ello, el usuario debe tener ambas manos abajo como se indica en la siguiente figura.



**Figura 14. Gesto posición neutra**

Este gesto se detecta cuando la coordenada en Y de ambas manos es menor que la coordenada en Y del nodo cadera centro.

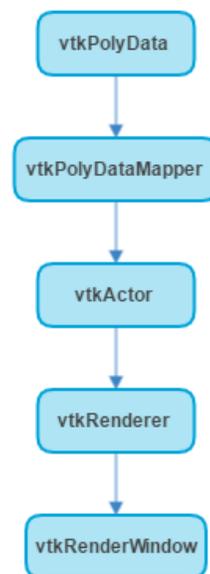
## 3.5. Desarrollo VTK

La librería VTK (*Visualization Toolkit*) en su versión 7.0.0 fue empleada para el desarrollo del componente de visualización del prototipo.

Como se mencionó en la [sección 2.1.4](#) VTK está conformado por nueve objetos básicos necesarios para desplegar en pantalla el modelo tridimensional que se va a manipular y deformar. En el prototipo se presenta una esfera que representa el modelo tridimensional de un pulmón. Estos dos modelos son homotópicos por lo cual son topológicamente equivalentes al deformarse. Una Adicionalmente, para el prototipo, dicho modelo tridimensional es representado a través de clase *vtkPolyData*, que a su vez está compuesta por puntos, líneas y polígonos.

Los pasos necesarios para desplegar en pantalla el *vtkPolyData* se presentan a continuación:

1. Crear el objeto básico *vtkPolyDataMapper* y asignarle como entrada el *vtkPolyData*
2. Crear el objeto básico *vtkActor* y asignarle el *vtkPolyDataMapper* creado previamente.
3. Crear el objeto básico *vtkRenderer* y adicionarle el *vtkActor*.
4. Crear el objeto básico *vtkRenderWindow* y adicionarle el *vtkRenderer* que ya fue creado.



**Figura 15. Pasos para desplegar un *vtkPolyData***

La librería VTK además de hacer posible la visualización del modelo tridimensional, cuenta con una serie de métodos que permite llevar a cabo con mayor facilidad las funcionalidades del prototipo. En las siguientes secciones se presentan dichas funcionalidades y los métodos de VTK que fueron empleados.

### 3.5.1. Alejar y acercar

Para alejar y acercar el modelo tridimensional en pantalla, es necesario crear el objeto básico *vtkCamara*. Este objeto cuenta con el método *Zoom*, el cual en modo paralelo modifica la escala paralela por un factor. Cuando dicho factor es mayor a 1 se realiza la acción acercar, y cuando es menor a 1 realiza la acción alejar.

### 3.5.2. Trasladar

La traslación se logra a través del *vtkActor* que tiene asociado el modelo tridimensional. Este objeto básico cuenta con el método *GetPosition*, el cual retorna la coordenada en X, Y y Z del centro del modelo.

Para trasladar de manera horizontal el modelo, la coordenada en X obtenida previamente es modificada. A esta coordenada se le suman 4 unidades si se requiere trasladar el modelo a la derecha o se le restan 4 unidades si se requiere trasladar a la izquierda.

La traslación vertical se logra al modificar la coordenada en Y obtenida. Si se requiere trasladar hacia arriba el modelo se le suman 4 unidades a esta coordenada, de lo contrario se le restan 4 unidades.

Hechas las modificaciones, se recurre nuevamente al *vtkActor*, este objeto cuenta con el método *SetPosition*, el cual modifica la posición actual del modelo tridimensional al recibir la nueva coordenada en X, Y y Z que fueron creadas previamente.

### 3.5.3. Rotar

Para llevar a cabo la rotación del modelo, se debe acceder al *vtkActor* que este tiene asignado. La rotación horizontal y vertical se logra a través del método *RotateWXYZ*. Este método recibe como parámetro un ángulo de rotación de  $45^\circ$  o  $-45^\circ$  según el sentido de rotación que se requiera.

---

### 3.5.4. Deformar

Para la deformación es necesario crear dos objetos nuevos: un cursor en pantalla y un *vtkKdTree*. El cursor está representado a través de una esfera pequeña, que indica la posición en la cual el usuario desea deformar el modelo tridimensional. Hecho esto, se construye un *vtkKdTree* a partir de los puntos del modelo tridimensional. Esto se hace con el objetivo de que la búsqueda de puntos en el modelo sea de menor complejidad algorítmica.

Al momento de deformar se accede al *vtkActor* del cursor, y a través de método *GetPosition* se obtiene constantemente la coordenada de su posición. Luego, se hace uso del método *FindClosestPoint* del objeto *vtkKdTree*. Este método retorna el punto más cercano con respecto a un punto dado, en este caso la posición del cursor. Ahora bien, cuando la distancia entre el cursor y el punto del modelo es de una unidad o menos, se indica que este punto es el que se debe deformar.

## 3.6. Desarrollo ITK

La librería ITK (*Insight Segmentation and Registration Toolkit*) en su versión 4.4 fue empleada para el desarrollo del componente de deformación.

En el prototipo se empleó una esfera para representar el modelo tridimensional, sin embargo, el prototipo se puede extender para que sea posible manipular y deformar el modelo de un pulmón.

El modelo tridimensional se encuentra representado inicialmente a través de una malla triangular, debido a esto es necesario transformarlo a una malla simplex ([ver sección 2.1.3](#)). Para ello se hace uso del filtro *TriangleMeshToSimplexMeshFilter* de la librería ITK. Hecho esto, se crea un objeto de la clase *itk::PointsLocator*, el cual forma un KDtree con los puntos de la malla simplex. A partir del objeto creado, se hace uso del método *FindClosestPoint* para obtener la coordenada en X,Y y Z del punto que se va deformar.

Para finalizar, la deformación se logra al modificar dichas coordenadas, a estas se les debe sumar o restar un vector cuyo valor en X, Y y Z es uno.

### 3.7. Integración

A continuación, se describe como se llevó a cabo la comunicación entre los componentes del prototipo.

#### 3.7.1. Integración de gestos con VTK

La comunicación entre las clases Kinect y Visualización del prototipo se puede apreciar claramente cuando se llevan a cabo las funcionalidades de alejar/acercar, trasladar y rotar.

A continuación, se presentan los diagramas de secuencia de las funcionalidades mencionadas.

##### 3.7.1.1. Alejar y acercar

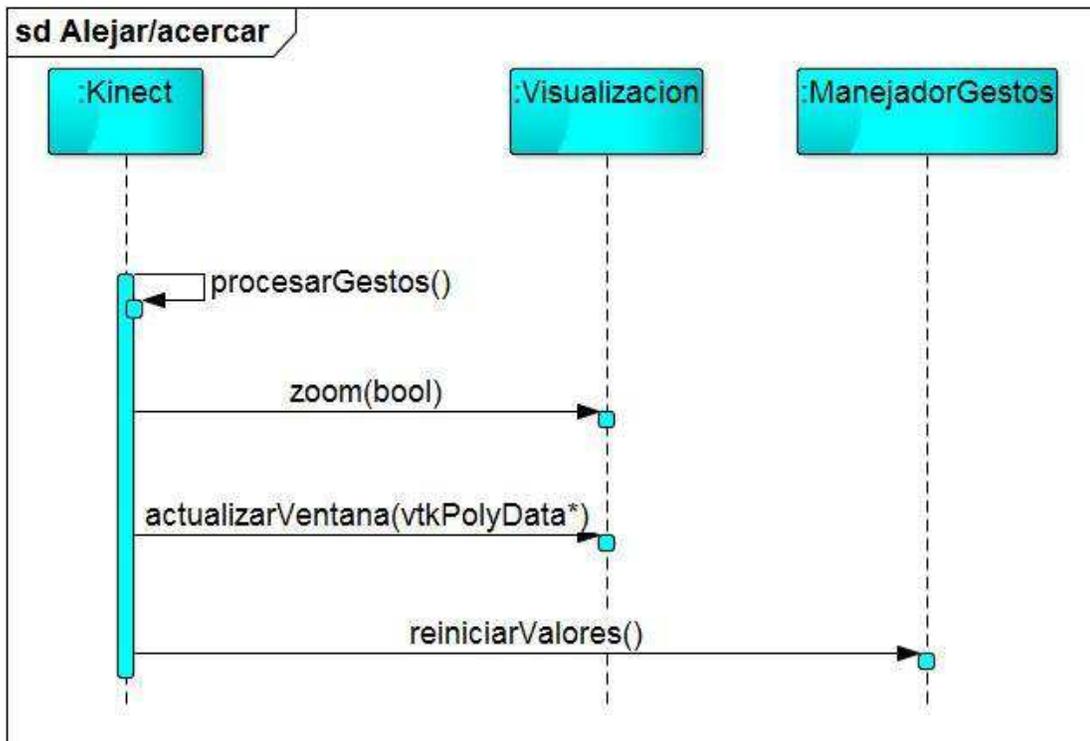


Figura 16. Diagrama de secuencia de acercar y alejar

Como se puede observar la comunicación entre la clase *Kinect* y *Visualizacion* para las funcionalidades de acercar y alejar, se hace a través del método *zoom* de la clase *Visualizacion*,

el cual recibe como parámetro un booleano que indica si debe acercar o alejar el modelo tridimensional ([ver sección 3.5.1](#)).

### 3.7.1.2. Trasladar

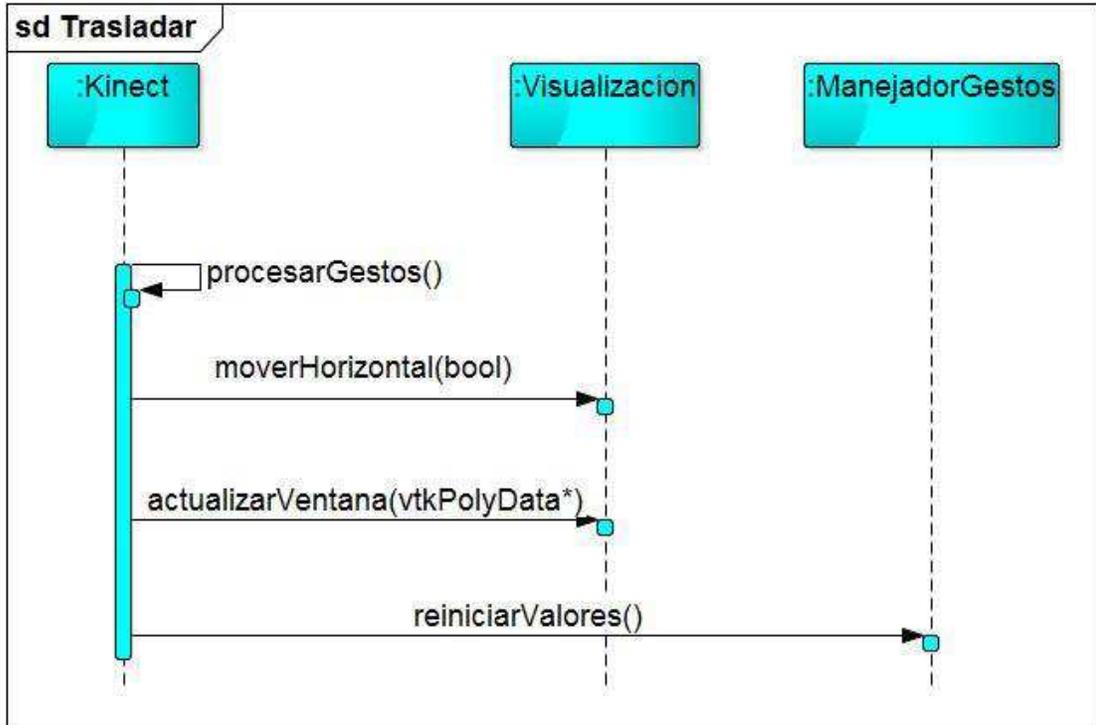


Figura 17. Diagrama de secuencia de traslación

La comunicación entre la clase *Kinect* y *Visualizacion* para la funcionalidad de trasladar se hace a través de los métodos *moverHorizontal* y *moverVertical* de la clase *Visualizacion*. En el diagrama presentado anteriormente se observa la secuencia al detectar un gesto de traslación horizontal. Dicho método recibe como parámetro un booleano que indica la dirección (derecha o izquierda) en la que el modelo tridimensional se debe mover ([ver sección 3.5.2](#)).

### 3.7.1.3. Rotar

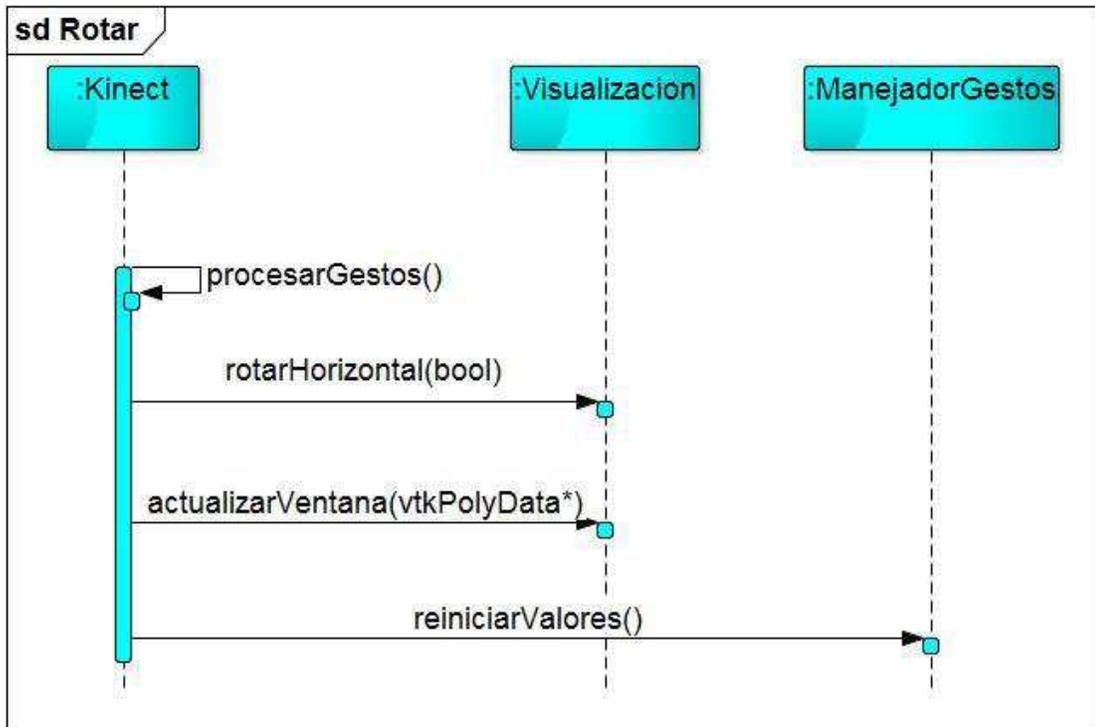


Figura 18. Diagrama de secuencia de rotación

La comunicación entre la clase *Kinect* y *Visualizacion* para la funcionalidad de rotar se hace a través de los métodos *rotarHorizontal* y *rotarVertical* de la clase *Visualizacion*. Al detectar el gesto rotar, *Kinect* llama alguno de estos métodos mencionados, en este caso, el diagrama presenta la secuencia al rotar de manera horizontal. Este método recibe como parámetro un booleano que indica la dirección en la que el modelo tridimensional se debe rotar ([ver sección 3.5.3](#)).

### 3.7.2. Integración de ITK con VTK

La comunicación entre la clase *Deformacion* y *Visualizacion* se hace a través de una clase adicional llamada *ITKaVTK*. Esta clase se encarga de convertir el modelo tridimensional, que está representado como malla triangular dentro de la clase *Deformacion*, a un objeto de tipo *vtkPolyData* que puede ser plasmado en pantalla por medio de la clase *Visualizacion*.

---

### 3.7.3. Integración de gestos con ITK y VTK

La comunicación entre los tres componentes del prototipo se puede evidenciar en la funcionalidad de deformar (ver figura 19).

La secuencia de esta funcionalidad comienza en el momento en que la clase **Kinect** detecta el gesto de deformar dentro del método *procesarGestos*, inmediatamente **Kinect** accede al método *activarDeformación* de la clase **Visualizacion**, la cual permite mostrar en pantalla el cursor de deformación. Posteriormente, mediante el método *convertirCoordenadas* La clase **Kinect** obtiene las coordenadas de la posición de la mano derecha del usuario según el dispositivo Kinect. Al tener estos datos, el método convierte las coordenadas obtenidas en nuevas coordenadas, proporcionales a la ventana de visualización de VTK y las asigna al cursor con el método *ubicaciónEsferaDeformación* de la clase **Visualizacion**.

A continuación, **Kinect** llama el método *puntoCercano* que devuelve el punto del modelo tridimensional más cercano al cursor ([ver sección 3.5.4](#)) y después llama el método *deformar* de la clase **Deformacion**, que modifica la topología del modelo de acuerdo a el punto obtenido previamente ([ver sección 3.6](#)).

Seguido a esto, la clase **Deformacion** convierte el modelo que se encuentra representado como malla simplex a una malla triangular con el método *simplexATriangular*. Para terminar, **Deformacion** accede al método *meshToPolydata* de la clase **ITKaVTK** para convertir la malla triangular en un objeto de tipo *vtkPolydata*. Dicho objeto es retornado a la clase **Kinect** y es enviado nuevamente a la clase **Visualizacion** para finalmente ser presentado en pantalla ([ver sección 3.5](#)).

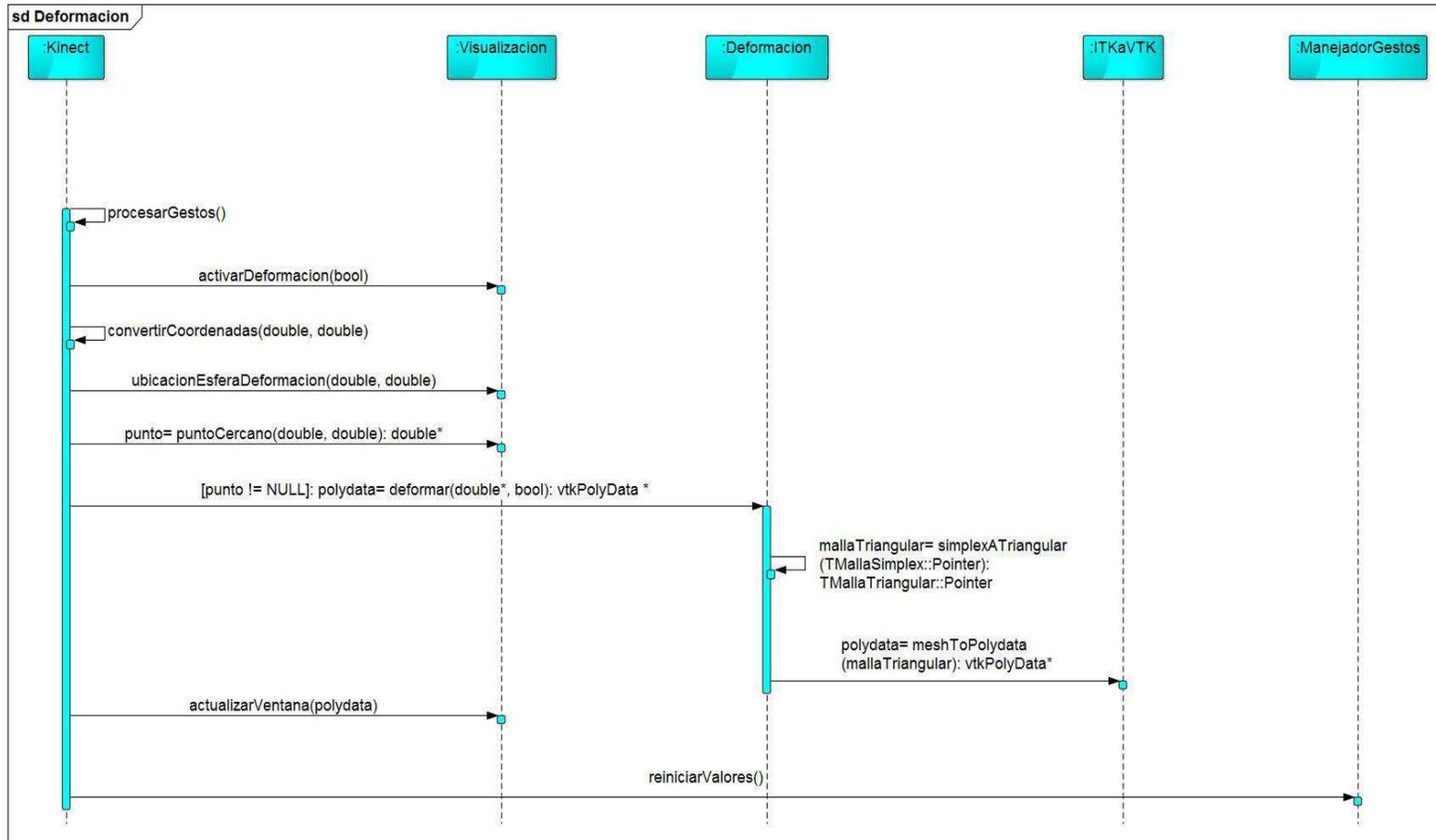


Figura 19. Diagrama de secuencia de deformación

## IV - PRUEBAS

Esta sección hace referencia a la última actividad de la tercera fase metodológica que corresponde a la iteración de pruebas. Estas pruebas se enfocaron en determinar si el esquema de interacción propuesto, facilitaba la manipulación y la deformación de modelos tridimensionales, que para este resultaba ser una esfera que representaba el modelo tridimensional de un pulmón.

### 4.1. Proceso de pruebas

En primera instancia, vale la pena destacar que se descartó un proceso de validación sobre el prototipo. Esto, es debido a que la validación requería de un gran número de personas expertas en el campo de la medicina y al menos 6 meses más para realizar pruebas de recordación sobre los gestos. Como ninguna de las dos fue posible se realizaron pruebas piloto sobre el prototipo.

Las pruebas desarrolladas se enfocaron en la usabilidad, esto permitió determinar si al usuario le resultaba sencillo efectuar los gestos descritos en la [sección 3.4](#). El objetivo de las pruebas estuvo centrado en dos aspectos: precisión y rendimiento.

Las pruebas se realizaron sobre tres personas con los siguientes perfiles:

- Profesor de planta de ingeniería de sistemas de la Pontificia Universidad Javeriana, con doctorado en ingeniería de sistemas y computación, experto en procesamiento de imágenes médicas.
- Estudiante de ingeniería de sistemas de la Pontificia Universidad Javeriana con énfasis en computación gráfica.
- Estudiante de ingeniería de sistemas de la Pontificia Universidad Javeriana con énfasis en construcción de software.

Las pruebas se desarrollaron de la siguiente manera:

1. Al iniciar la prueba se realizó una corta presentación sobre los gestos y la respectiva funcionalidad de cada uno de ellos.

2. Luego se pidió a los sujetos que interactuaran libremente con el prototipo para que se familiarizaran con la interfaz y los gestos.
3. A continuación, se explicó en qué consistía la prueba: En pantalla aparecería un gesto aleatorio y el sujeto debía realizar tres veces ese gesto para poder avanzar. La prueba finalizaba luego de que se realizarán 12 gestos que incluían: Acercar, alejar, deformar y, trasladar y rotar en todas las direcciones.
4. Cada sujeto debía realizar tres veces la prueba mencionada con el objetivo de evaluar la precisión y el rendimiento de los gestos.

Durante las pruebas se calculó el tiempo que le tomó a cada sujeto realizar cada uno de los gestos. Con el resultado de los tiempos de la primera y última prueba se obtuvo el aspecto de rendimiento que se había especificado en el objetivo de las pruebas. Para evaluar el rendimiento, se determinó el porcentaje de reducción del tiempo en la realización de cada gesto.

$$\text{Porcentaje de rendimiento: } \frac{\text{Tiempo inicial} - \text{Tiempo final}}{\text{Tiempo inicial}} \times 100$$

Como cada gesto se realizó nueve veces al finalizar las tres pruebas, se determinó cuántas de esas nueve veces se había hecho el gesto erróneamente; sobre este resultado se calculó el porcentaje de precisión de la siguiente manera:

$$\text{Porcentaje de precisión: } \frac{9 - \text{número de errores}}{9} \times 100$$

Adicionalmente, se pidió a cada sujeto que hiciera una retroalimentación sobre su experiencia al interactuar con el prototipo. Esta retroalimentación se basó en una encuesta que se presenta a continuación:

		1	2	3	4	5
1	Compleitud en las funcionalidades para manipular objetos tridimensionales					
2	Los gestos resultan intuitivos con respecto a las funcionalidades					
3	Los gestos fueron detectados fácilmente por el dispositivo Kinect					
4	La respuesta de la interfaz es acorde a la acción que se está realizando					
5	¿Qué gestos le presentaron mayor dificultad?					
6	¿Agregaría alguna funcionalidad al prototipo presentado?					
	Observaciones					

Finalmente, vale la pena destacar que las pruebas de código se despreciaron, esto se debe a que se utilizaron librerías externas para llevar acabo la visualización y deformación del modelo tridimensional. Se asume que los desarrolladores de ITK y VTK ya realizaron pruebas sobre las funcionalidades de las librerías.

## 4.2. Resultados

A partir de las pruebas realizadas se determinó que existieron errores en el proceso de diseño de gestos. Esto, debido a que los sujetos presentaron dificultades al momento de realizar ciertas funcionalidades del prototipo.

En la siguiente tabla se puede observar el porcentaje promedio de precisión de cada uno de los gestos.

Gesto	Promedio de precisión
Deformar repeler	100%
Deformar atraer	96%
Rotar Derecha	100%
Rotar Izquierda	100%
Rotar Arriba	100%
Rotar Abajo	100%
Acercar	81%
Alejar	70%
Mover Derecha	100%
Mover Izquierda	100%
Mover Arriba	100%
Mover Abajo	100%

**Tabla 2. Resultados porcentaje de precisión**

A través de los resultados presentados en la tabla 1 se puede observar que los gestos de acercar y alejar contaron con el menor porcentaje de precisión. Esto se debió principalmente a que las personas presentaban una gran dificultad a la hora de recordar los movimientos que definían estos dos gestos. En cuanto a los demás gestos, se puede observar que en su mayoría se logró una precisión del 100%, lo cual indica que eran intuitivos y fueron diseñados correctamente.

Por otro lado, se tomaron en cuenta los porcentajes de reducción de tiempo entre la primera y la última prueba, este resultado se encuentra en la columna *Rendimiento* de la siguiente tabla *promedio*. Adicionalmente se presenta el tiempo promedio de cada una de las pruebas.

Gesto	Promedio tiempo 1 (ms)	Promedio tiempo 2 (ms)	Promedio tiempo 3 (ms)	Rendimiento promedio
Deformar atraer	7473,67	3468,33	1900,00	75%
Deformar repeler	15951,33	6836,67	6408,33	60%
Mover Derecha	11130,67	6494,67	4692,33	58%
Rotar Arriba	11490,50	6221,67	4900,33	57%
Mover Izquierda	7263,00	4325,67	3352,67	54%
Mover Arriba	11897,67	8553,67	5695,00	52%
Rotar Derecha	12016,67	6006,67	5779,33	52%
Rotar Abajo	10783,00	6627,00	5210,00	52%
Mover Abajo	12491,67	6311,67	7738,00	38%
Rotar Izquierda	11361,33	9614,00	8120,67	29%
Alejar	11875,33	10436,33	9363,33	21%
Acercar	7505,00	7145,00	6005,00	20%
<b>Total tiempo prueba</b>	<b>131239,83</b>	<b>82041,33</b>	<b>69165,00</b>	<b>47%</b>

**Tabla 3. Resultados porcentaje de rendimiento**

A partir de los datos de la tabla 2 se puede observar que el tiempo promedio de las pruebas disminuyó en un 47%, lo cual indica que en general los gestos eran fáciles de recordar (excluyendo los gestos de alejar y acercar, como se mencionó anteriormente).

Al analizar de manera individual los tiempos de cada gesto en las tres pruebas, es evidente que el tiempo del gesto rotar hacia la izquierda disminuyó muy poco a comparación de los demás gestos de rotar (derecha, arriba y abajo). Sin embargo, este resultado no se debió a que el gesto

no fuera intuitivo, sino por un error en la detección. Cuando los sujetos realizaban este gesto, extendían mucho el brazo hacia la izquierda, lo cual conllevaba a que la mano tuviera la misma coordenada en Z del pecho y se terminara el gesto (manos al pecho, [ver sección 3.4.5](#)).

Por último, los resultados de las encuestas se muestran en la tabla 4. Con la información que se presenta, se pudo concluir que el Kinect resulta ser una herramienta adecuada para la manipulación de objetos tridimensionales. Sin embargo, la interfaz gráfica que se desarrolló para este prototipo debe mejorarse en trabajos futuros; en pantalla se debe mostrar de una manera más adecuada la retroalimentación de los gestos que está llevando a cabo el usuario.

	Pregunta	Resultado
1	Completitud en las funcionalidades para manipular objetos tridimensionales	4,33
2	Los gestos resultan intuitivos con respecto a las funcionalidades	4,33
3	Los gestos fueron detectados fácilmente por el dispositivo Kinect	4,67
4	La respuesta de la interfaz es acorde a la acción que se está realizando	4,00

**Tabla 4. Resultados encuesta**

En el caso de las preguntas abiertas, por un lado, todos estuvieron de acuerdo que los gestos de acercar y alejar resultaron ser los más complicados. Por otro lado, uno de los sujetos propuso que la esfera deformadora debía tener la capacidad de aumentar o disminuir su tamaño para que proporcionalmente aumentara o disminuyera la deformación sobre el objeto tridimensional.

## V - CONCLUSIONES

### 5.1. Análisis de Impacto

Desde el punto de vista académico, se espera que este trabajo de investigación sirva como base para la creación de un sistema completo que permita manipular y mejorar los modelos tridimensionales de los pulmones. Una vez el sistema sea instaurado en los hospitales, se tendrá un gran impacto a nivel social. Se espera que este sistema permita a los médicos generar

diagnósticos más acertados y de una manera más rápida con respecto al método de diagnóstico tradicional. A nivel económico, es importante mencionar que este trabajo de investigación propone el uso de un sensor muy económico, lo cual generaría un gran ahorro en los hospitales a comparación de los sensores que se emplean actualmente.

## 5.2. Conclusiones del proyecto

- Tras finalizar este trabajo de investigación, se pudo concluir que el Kinect efectivamente cuenta con la precisión necesaria para manipular y modificar la estructura topológica de una esfera. Vale la pena destacar que existe homotopía entre el modelo tridimensional de un pulmón y una esfera, es decir que uno puede deformarse continuamente hasta convertirse en el otro, por lo cual pueden llegar a ser topológicamente equivalentes.

El Kinect de Microsoft cuenta con un sensor de profundidad que facilitó la ubicación del usuario en el espacio. Esto era un aspecto primordial debido a que los movimientos que el usuario realizaba en el mundo real, debían verse reflejados en el espacio tridimensional que se plasmaba en pantalla.

- Desde un principio se esperaba que el prototipo pudiese ser ejecutado tanto en Windows como en sistemas operativos Unix. Sin embargo, al momento de llevar a cabo la fase de exploración se evidenció que la librería necesaria para desarrollar el prototipo en Unix, ya no se encontraba disponible. Esto se debe a que la compañía desarrolladora había sido absorbida por la empresa Apple Inc. Se esperaría que, en un futuro estas librerías vuelvan a estar disponibles, o se desarrollen nuevas que permitan la compatibilidad entre el Kinect y otros sistemas operativos.
- Las librerías ITK y VTK que se emplearon en el prototipo, resultaron suficientes para el desarrollo de la visualización y el procesamiento. Sin embargo, se encontró que en la librería ITK los filtros de deformación de las mallas simplex presentaban errores al momento de desconectar el flujo de datos, por lo que se debía crear una nueva malla simplex para cada deformación.

- A partir de las pruebas realizadas se pudo concluir que el 83% de los gestos asociados a las funcionalidades del prototipo fueron diseñados correctamente, ya que eran sencillos e intuitivos para el usuario. Sin embargo, los gestos de acercar y alejar resultaron en algunas ocasiones confusos, por lo que en trabajos futuros se deberían rediseñar estos gestos.

Adicionalmente, como trabajo futuro se esperaría realizar una validación completa sobre el prototipo, ya que por falta de tiempo y de recursos en este trabajo únicamente se pudo llevar a cabo pruebas piloto de usabilidad.

- En los últimos años se ha evidenciado una gran innovación en el área de la interacción hombre-máquina, especialmente en la interacción natural de usuario. Esto se ha visto reflejado en la creación de múltiples sensores y dispositivos como lo es el Kinect de Microsoft. A pesar de estos grandes avances, la industria de los videojuegos es la única que ha sacado provecho de estas tecnologías.

Es claro que no se ha explotado todo el potencial de estas nuevas tecnologías, es por esto que empleamos el Kinect como parte fundamental de este trabajo de investigación. Como conclusión, se puede afirmar que esta clase de tecnología se debería aplicar en mayor medida a diferentes áreas del conocimiento, con el objetivo de aumentar la productividad de las personas.

## VI - REFERENCIAS Y BIBLIOGRAFÍA

- Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1), 25–30.
- Brown, A. & Wilson, G. (n.d.). The Architecture of Open Source Applications: Elegance, Evolution, and a Few Fearless Hacks, volume I. aosabook. org, March 2012. *Online at Http://aosabook. Org.*
- Crossley, M. D. (2005). *Essential Topology*. Springer.
- Darmanayagam, S. E., Harichandran, K. N., Cyril, S. R. R. & Arputharaj, K. (2013). A novel supervised approach for segmentation of lung parenchyma from chest CT for computer-aided diagnosis. *Journal of Digital Imaging*, 26(3), 496–509. doi:10.1007/s10278-012-9539-6
- Das, C. J., Seith, A. & Mukhopadhyay, S. (2007). Thoracic application of multi-detector CT. *Indian J Chest Dis Allied Sci*, 49(1), 29–36.
- De Nunzio, G., Tommasi, E., Agrusti, A., Cataldo, R., De Mitri, I., Favetta, M., ... Oliva, P. (2011). Automatic lung segmentation in CT images with accurate handling of the hilar region. *Journal of Digital Imaging*, 24(1), 11–27. doi:10.1007/s10278-009-9229-1
- Delingette, H. (1994). Simplex meshes: a general representation for 3D shape reconstruction. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on* (pp. 856–859).
- Doi, K. (2006). Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology. *Physics in Medicine and Biology*, 51(13), R5–27.
- Dong, H., Xia, L., Zhang, J. & Cai, A. (2013). Medical Image Reconstruction Based on ITK and VTK. In *Computer Sciences and Applications (CSA), 2013 International Conference on* (pp. 642–645).

Gallo, L., Placitelli, A. P. & Ciampi, M. (2011). Controller-free exploration of medical image data: Experiencing the Kinect. In *Computer-based medical systems (CBMS), 2011 24th international symposium on* (pp. 1–6).

Gonzalez, R. C. & Woods, R. E. (2002). *Digital image processing* (2nd ed.). Upper Saddle River, N.J: Prentice Hall.

Heathcote, P. M. (2003). *“A” Level ICT*. Payne Gallway.

Iyer, P. W., Levin, B. J. & Shea, M. A. (2006). *Medical Legal Aspects of Medical Records*. Lawyers & Judges Publishing Company.

Jana, A. (2012). *Kinect for Windows SDK Programming Guide*. Packt Publishing Ltd.

Johnson, H. J., McCormick, M. M. & Ibanez, L. (2014). *The ITK Software Guide Book 1: Introduction and Development Guidelines Fourth Edition Updated for ITK version 4.7*.

Khoshelham, K. & Elberink, S. O. (2012). Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12(2), 1437–1454. doi:10.3390/s120201437

Kipshagen, T., Graw, M., Tronnier, V., Bonsanto, M. & Hofmann, U. (2009). Touch-and marker-free interaction with medical software. In *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany* (pp. 75–78).

Kumar. (2005). *Human Computer Interaction*. Firewall Media.

Ladzinski, A. (2012). Development of 3D image manipulation software utilizing the microsoft kinect.

Laramee, R. S. (2008). Comparing and evaluating computer graphics and visualization software. *Software: Practice and Experience*, 38(7), 735–760.

Larman, C. (2004). *Agile and iterative development: a manager's guide*. Boston, Mass.: Addison-Wesley.

Li, X., Wang, X., Dai, Y. & Zhang, P. (2015). Supervised recursive segmentation of volumetric CT images for 3D reconstruction of lung and vessel tree. *Computer Methods and Programs in Biomedicine*.

Lo, P. C. P. (2010). *Segmentation of Lung Structures in CT*.

Martins, M. G. (2004). TOMOGRAFÍA AXIAL COMPUTADA EN EL DIAGNÓSTICO DE PATOLOGÍAS PULMONARES.

Mathew, A. R., Al Hajj, A. & Al Abri, A. (2011). Human-computer interaction (hci): An overview. In *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on* (Vol. 1, pp. 99–100).

McAuliffe, M. J., Lalonde, F. M., McGarry, D., Gandler, W., Csaky, K. & Trus, B. L. (2001). Medical image processing, analysis and visualization in clinical research. In *Computer-Based Medical Systems, 2001. CBMS 2001. Proceedings. 14th IEEE Symposium on* (pp. 381–386).

Nakata, H., Kimoto, T., Nakayama, T., Kido, M., Miyazaki, N. & Harada, S. (1985). Diffuse peripheral lung disease: evaluation by high-resolution computed tomography. *Radiology*, 157(1), 181–185.

Of Radiology, E. S. (2013). Breathe Easy.

Ogiela, M. R. & Hachaj, T. (2015). *Natural User Interfaces in Medical Image Analysis: Cognitive Analysis of Brain and Carotid Artery Images*. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=795508>

Schroeder, W. J., Martin, K. M. & Lorensen, W. E. (1996). The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In *Proceedings of the 7th conference on Visualization'96* (p. 93–ff).

Schaefer-Prokop, C. (2013). An over view of chest imaging. *Breath Easy*, 4-7.

Sommerville, I. & Galipienso, M. I. A. (2005). *Ingeniería del software*. Pearson Educación.

*TEDCAS | NATURAL USER INTERFACES FOR HEALTHCARE*. (2016). Tedcas.com. Consultado el 23 Mayo de 2016, en <http://www.tedcas.com>

Thimbleby, H. (2007). *User-centered methods are insufficient for safety critical systems*. Springer.

Tuntakurn, A., Thongvigitmanee, S. S., Sa-Ing, V., Hasegawa, S. & Makhanov, S. S. (2013). Natural interactive 3D medical image viewer based on finger and arm gestures. In *Biomedical Engineering International Conference (BMEiCON), 2013 6th* (pp. 1–5).

Van den Boomen, M. (2009). *Digital material: Tracing new media in everyday life and technology* (Vol. 2). Amsterdam University Press.

Wei, Y., Shen, G. & Li, J. (2013). A fully automatic method for lung parenchyma segmentation and repairing. *Journal of Digital Imaging*, 26(3), 483–95. doi:10.1007/s10278-012-9528-9

Wesson, B. & Wilkinson, B. (2013). Evaluating organic 3D sculpting using natural user interfaces with the Kinect. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration* (pp. 163–166).

Withey, D. & Koles, Z. (2007). Medical image segmentation: Methods and software. In *Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging, 2007. NFSI-ICFBI 2007. Joint Meeting of the 6th International Symposium on* (pp. 140–143).

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2), 4–10.

## VII - ANEXOS

### **Anexos**

Los anexos se encuentran disponibles en <http://pegasus.javeriana.edu.co/~CIS1610IN02>, estos son:

- Glosario
- Manual de usuario y de instalación
- Artículo
- Video de demostración del prototipo
- Código fuente y documentación del prototipo