

## SPMP (SOFTWARE PROJECT MANAGEMENT PLAN) PARA PROYECTO DE GRADO

### TÍTULO

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

### MODALIDAD: Proyecto de investigación

### ESTUDIANTE(S)

#### Cesar Alejandro Guayara Rodríguez

Documento	Celular	Teléfono fijo	Correo Javeriano
cc. 1015438560	316-529-6135	7319482	<a href="mailto:c_guayara@javeriana.edu.co">c_guayara@javeriana.edu.co</a>

#### Erika Jennifer Harker Gutiérrez

Documento	Celular	Teléfono fijo	Correo Javeriano
cc. 1018471803	301-341-2386	7599794	<a href="mailto:eharker@javeriana.edu.co">eharker@javeriana.edu.co</a>

#### Juan Manuel Sánchez Lozano

Documento	Celular	Teléfono fijo	Correo Javeriano
cc. 1013665691	314-471-2446	-	<a href="mailto:jsanchez.l@javeriana.edu.co">jsanchez.l@javeriana.edu.co</a>

#### Juan Miguel Gómez Ganem

Documento	Celular	Teléfono fijo	Correo Javeriano
cc. 1020822419	304-574-9104	-	<a href="mailto:jm-gomez@javeriana.edu.co">jm-gomez@javeriana.edu.co</a>

#### Luis David Zárate Castillo

Documento	Celular	Teléfono fijo	Correo Javeriano
cc. 1014282537	304-382-3335	-	<a href="mailto:zarate.luis@javeriana.edu.co">zarate.luis@javeriana.edu.co</a>

### DIRECTOR

#### Leonardo Flórez Valencia

Documento	Celular	Teléfono fijo	Correo Javeriano	Empresa donde trabaja y cargo
cc. -	314-893-2219	-	<a href="mailto:florez-l@javeriana.edu.co">florez-l@javeriana.edu.co</a> ;	Pontificia Universidad Javeriana; Profesor Departamento de Sistemas

## Contenido

<b>LISTA DE TABLAS E ILUSTRACIONES .....</b>	<b>4</b>
LISTA DE TABLAS .....	4
LISTA DE ILUSTRACIONES .....	4
<b>1 PREFACIO.....</b>	<b>5</b>
<b>2 VISTA GENERAL DEL PROYECTO .....</b>	<b>6</b>
2.1 VISIÓN DEL PRODUCTO .....	6
2.2 PROPÓSITO .....	6
2.3 ALCANCE .....	6
2.4 OBJETIVOS .....	6
2.4.1 <i>Objetivo general</i> .....	6
2.4.2 <i>Objetivos Específicos</i> .....	6
2.5 SUPUESTOS Y RESTRICCIONES.....	7
2.5.1 <i>Supuestos</i> .....	7
2.5.2 <i>Restricciones</i> .....	7
2.6 ENTREGABLES.....	7
<b>3 GLOSARIO .....</b>	<b>9</b>
<b>4 CONTEXTO DEL PROYECTO .....</b>	<b>10</b>
4.1 MODELO DE CICLO DE VIDA .....	10
4.1.1 <i>Fases del Proyecto</i> .....	10
4.1.2 <i>Metodología del proyecto</i> .....	10
4.2 ORGANIGRAMA Y DESCRIPCIÓN DE ROLES .....	11
<b>5 ADMINISTRACIÓN DEL PROYECTO.....</b>	<b>13</b>
5.1 LENGUAJES Y HERRAMIENTAS .....	13
5.1.1 <i>Estándares de Desarrollo</i> .....	13
<b>6 MONITOREO Y CONTROL DEL PROYECTO.....</b>	<b>14</b>
6.1 ADMINISTRACIÓN DE REQUERIMIENTOS .....	14
6.1.1 <i>Priorización de los Requerimientos</i> .....	14
6.1.2 <i>Estados del requerimiento</i> .....	15
6.2 ACEPTACIÓN DE PRODUCTO .....	15
<b>7 PROCESOS DE SOPORTE.....</b>	<b>16</b>
7.1 ANÁLISIS DE RIESGOS .....	16
7.2 MÉTRICAS Y PROCESO DE MEDICIÓN .....	17

7.3 CONTROL DE CALIDAD.....	18
7.3.1 Estrategia de pruebas .....	18
7.3.2 Niveles de pruebas .....	18
7.3.3 Tipos de pruebas .....	19
7.3.4 Técnicas de diseño de pruebas.....	19
<b>8 REFERENCIAS .....</b>	<b>21</b>

## Lista de tablas e ilustraciones

### Lista de tablas

<i>Tabla 1. Entregables del Proyecto en cada fase .....</i>	8
<i>Tabla 2. Entregables del Proyecto en cada fase [5] .....</i>	9
<i>Tabla 3. Priorización de requerimientos [7].....</i>	14

### Lista de ilustraciones

<i>Ilustración 1. Organigrama del proyecto .....</i>	12
<i>Ilustración 2. Niveles de pruebas .....</i>	18

## 1 Prefacio

Este documento presenta el plan de gestión de proyectos de software del proyecto de investigación “Análisis del procesamiento de imágenes médicas pulmonares para el diagnóstico y tratamiento del SDRA”; dirigido a todo aquel interesado en conocer la solución propuesta, entorno al procesamiento de imágenes, para el estudio de la problemática que se vive actualmente a la hora de tratar y diagnosticar a un paciente que sufre del síndrome de dificultad de respiración aguda.

El síndrome de dificultad de respiración aguda o SDRA es un proceso de edema pulmonar no hidrostático, el cual es el resultado de un desequilibrio entre las fuerzas que ocasionan la entrada de líquido en los alvéolos y los mecanismos para retirarlo, e hipoxemia [1]. El tratamiento de este síndrome toma tiempo, lo que es esencial para tratar a un paciente con este, de lo contrario se puede ver reflejado en la vida del paciente. A pesar de que actualmente existen herramientas, principalmente enfocadas en el procesamiento de imágenes, que pueden apoyar el diagnóstico toma demasiado tiempo para capacitarse en su uso y no siempre arrojan resultados con información relevante.

El proyecto de investigación en primera instancia desea realizar un análisis entorno a las principales herramientas en el mercado enfocadas en el procesamiento de imágenes médicas y los algoritmos que estas contiene, con el fin de escoger los más adecuados, basados en aspectos técnicos. Con los algoritmos una vez seleccionados se tiene como objetivo realizar un sistema que permita integrar al menos dos de estos algoritmos y de esta forma brindar en un solo punto varios y distintos resultados, representados en una nueva imagen que brinde información extra que pueda ayudar al estudio y tratamiento del SDRA.

## 2 Vista general del proyecto

Crear un sistema que permita integrar distintos algoritmos de procesamiento de imágenes médicas, con el fin de brindar un soporte para el estudio y tratamiento del SDRA por medio de la segmentación de TACs pulmonares. De esta forma se desea brindar un punto de acceso que reúna distintos recursos que ataquen este problema y así dar la posibilidad de tener resultados variados desde diferentes aproximaciones con el fin de que la información que estos resultados traigan facilite el estudio y tratamiento del SDRA.

### 2.1 Visión del Producto

Brindar una interfaz que permita al usuario procesar distintas imágenes pulmonares a través de varios algoritmos de procesamiento de imágenes médicas, ya sea por cada una de estas o por alguna en específico dependiendo de las opciones de selección que se exponen en la interfaz.

### 2.2 Propósito

Este proyecto es desarrollado teniendo en cuenta las necesidades de aquellas personas cuyo trabajo es diagnosticar y estudiar el SDRA, los cuales se verían beneficiados de la información y detalle que el aplicativo brinda, teniendo en cuenta que la prioridad es bajar los tiempos de diagnóstico de dicho síndrome de días a horas y que la curva de aprendizaje de la herramienta para el usuario sea más accesible que las demás herramientas en el mercado. Este proyecto ofrecerá la integración de varios algoritmos al servicio del usuario para que, según las imágenes ingresadas, se entregue una serie de imágenes procesadas que aporten mayor información para poder facilitar su análisis por de los expertos.

### 2.3 Alcance

Crear un prototipo que integre al menos dos algoritmos de procesamiento de imágenes médicas, que proporcionen una variedad de resultados en forma de nuevas imágenes que permitan agilizar el estudio y tratamiento del SDRA por medio de la información extra que se pueda identificar al procesar los TACs pulmonares originales.

### 2.4 Objetivos

#### 2.4.1 Objetivo general

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

#### 2.4.2 Objetivos Específicos

1. Realizar una investigación de los algoritmos de procesamiento de imágenes médicas que existen actualmente.

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

## 2.5 Supuestos y Restricciones

### 2.5.1 Supuestos

- **Claridad en las imágenes:** Se entiende que las imágenes a procesar serán lo suficiente claras para que los algoritmos que las usaran puedan funcionar correctamente.

### 2.5.2 Restricciones

- **Peso de las imágenes:** En vista que los TACs son imágenes sumamente llenas de información y al estar compuesta de diversos cortes, se generan imágenes muy pesadas y en vista que toda la información allí presente es importante, el proyecto no se puede permitir muchos reajustes a estas imágenes iniciales, ya que se podría estar información necesaria para un mejor procesamiento de las TACs.
- **Tiempo de procesamiento:** Esta restricción está ligada al tiempo que determinados algoritmos les pueden llegar a tomar el procesar las imágenes iniciales, ya que si el proceso toma demasiado el resultado obtenido puede ser muy tarde para cuando se necesitaba, hablando estrictamente en el momento de apoyar a un médico con esta herramienta para el tratamiento del SDRA.

## 2.6 Entregables

Fase	Entregable	Descripción
Configuración e investigación	1ra versión del SRS	Versión inicial del SRS, donde se muestra la primera especificación de los requerimientos principales (funcionales y no funcionales)
Análisis de requerimientos y diseño	2da versión del SRS	Versión final del SRS, donde se muestran los requerimientos del sistema que se van a implementar, y requerimientos asociados que se pueden desarrollar en trabajos futuro
	SDD	Descripción de la arquitectura del sistema y su diseño detallado

Desarrollo y plan de pruebas	Prototipo	Producto final que va a entregar el proyecto, en donde se implementaran los requerimientos principales del sistema
	Manual de usuario	Documento que explica en uso del producto y el mantenimiento de este
	Documentación de pruebas	Documento que muestra las pruebas realizadas sobre el sistema y los resultados de esta
	Memoria del proyecto	Documento final del proyecto

*Tabla 1. Entregables del Proyecto en cada fase*



### 3 Glosario

- Sistema: Producto que se va a desarrollar en el proyecto
- Cliente: Persona inmediata que va a recibir el producto
- Usuario: Persona o sistema que va a utilizar el producto
- Entregables: Productos que, en un cierto estado, se intercambian entre los clientes y los desarrolladores a lo largo de la ejecución del proyecto informático.
- SDRA: Síndrome de Dificultad Respiratoria Aguda
- Procesamiento de imágenes: es mejorar el aspecto de las imágenes con el fin de facilitar la búsqueda de información dentro de esta y denotar detalles relevantes para el observador [2].
- Segmentación de imágenes: técnica de tratamiento digital de imágenes que permite extraer información de los objetos a partir de unas escenas dadas y plasmar esta información en un sistema estructurado, que en muchas ocasiones consta de una única estructura [3]
- TAC: Tomografía Axial Computarizada. Técnica radiológica que se sustenta en la obtención de imágenes por planos, basada en la emisión de rayos X y un algoritmo computarizado para reconstruir la imagen.[4]
- Anomalía: Cualquier cosa observada en la documentación o el funcionamiento del software que se desvía de las expectativas basadas en verificaciones previas de los productos de software o documentos de referencia.
- Abortar: La terminación de un proceso antes de su correcta finalización

En la siguiente tabla se presentan la lista de acrónimos para el este documento.

Acrónimo	Significado
API	Aplication Programming Interface
CRUD	Create, Retrieve, Update, Delete
DBMS	Data Base Management System
JDBC	Java DataBase Connectivity
JVM	Java Virtual Machine
LAN	Local Area Network
RFC	Request For Comments
SDD	Software Design Description
SQL	Structured Query Language
SRS	Software Requirement Specification
IEEE	Institute of Electrical and Electronics Engineers
V&V	Verificación y Validación (Verification and Validation)

*Tabla 2. Entregables del Proyecto en cada fase [5]*

## 4 Contexto del proyecto

### 4.1 Modelo de Ciclo de Vida

#### 4.1.1 Fases del Proyecto

El proyecto se dividirá en tres fases. (ver “3. Proceso” en la propuesta de proyecto de grado)

1. Configuración e investigación: En esta fase se realizará el objetivo específico 1. Va a estar enfocada a la investigación de programas que realizan segmentación de imágenes médicas. Aquí se incluye el análisis de características técnicas, así como las características cualitativas que los médicos le dan a estas y se analiza los algoritmos que estas proveen a nivel de resultados.
2. Análisis de Requerimientos y Diseño: En esta fase se realizará los objetivos específicos 2 y 3. Una vez se tenga la lista de programas que van a integrar, se analiza los requerimientos técnicos, algunos ya evaluados en la fase anterior para la selección de herramientas, necesarios para la implementación del sistema. También, se definen los requerimientos formalmente y se validan con el profesor Leonardo Flórez. A continuación, se realiza el diseño arquitectural del sistema. Por último, una vez aprobada la arquitectura, se realiza el diseño detallado.
3. Desarrollo y Plan de Pruebas: En esta fase se realizará los objetivos específicos 4 y 5. Se realiza el desarrollo del sistema para generar el prototipo, las pruebas sobre los requerimientos principales y la documentación del prototipo

Aunque las fases se presentan en forma de cascada, cada fase se realizará de forma evolutiva incremental, con el fin de disminuir el impacto de los errores y tener mayor flexibilidad en el proyecto.

#### 4.1.2 Metodología del proyecto

Para el desarrollo del proyecto se utilizará dos metodologías, una enfocada en la investigación (1ra fase) y otra enfocada en el diseño y desarrollo del sistema (2da y 3ra fase). Para esto se utilizará la metodología de comparación y scrum respectivamente. Además, como se mencionó anteriormente el proceso de requerimientos, diseño y desarrollo se va a realizar de forma incremental. En estas fases (2da y 3ra) se utilizará principalmente elementos de Scrum. (ver “3. Proceso” en la propuesta de proyecto de grado).

La metodología de comparación es un procesamiento de búsqueda de similitudes y comparaciones sistemáticas que sirve para la verificación de hipótesis con el objeto de encontrar parentescos y se basa en la documentación de múltiples casos para realizar análisis comparativos.

A continuación, se explica las características seleccionadas de Scrum para el desarrollo del proyecto, puesto que no se va a utilizar la metodología de Scrum en su totalidad:

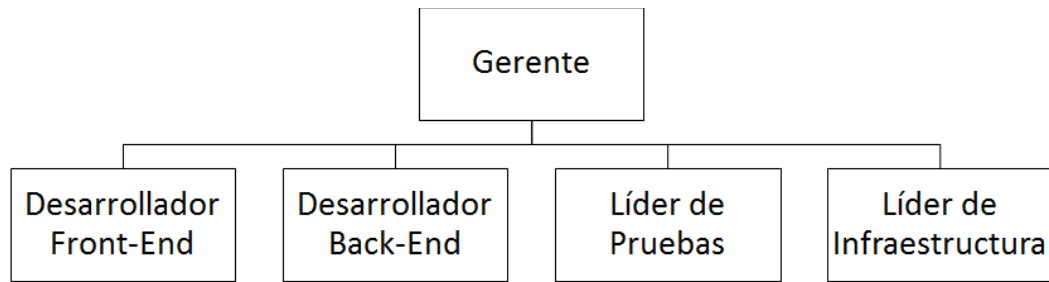
- Se van a realizar *entregas parciales y regulares* cada 2 semanas. Se cambiará este tiempo si es necesario previa reunión con el profesor Leonardo Flórez, de acuerdo con el avance que se haya realizado en el proyecto

- Se tendrá un “*Product backlog*” en donde se tendrán los objetivos y alcance del proyecto, al igual que las actividades necesarias para el cumplimiento de estos, priorizando según la importancia para el cliente y el coste de recursos para el grupo de trabajo [6].
- *Sprint planner*: Se planeará cómo será el siguiente Sprint refinando y definiendo las tareas [6].
- *Sprint*: Se realizan las tareas propuesta “Sprint planner” y se realiza una entrega al cliente (Ing. Leonardo Flórez). A continuación, el cliente aprueba o sugiere cambios sobre la entrega. Estos cambios se priorizan y se introducen en el Product backlog. Así, el cliente conoce el avance [6].
- Se realizará el *Sprint review* y *Sprint retrospective* en una sola reunión donde se encontrarán por lo menos un representante del grupo de trabajo y el cliente. Aquí se analizará los problemas que se generaron en el Sprint, enfocándose en especialmente en el proceso. A continuación, se plantea soluciones y si es necesario cambios en la forma como se está llevando el proceso [6].
- A diferencia de Scrum, no se van a realizar las reuniones diarias a causa de la disponibilidad del grupo de trabajo. Sin embargo, se realizará por lo menos reuniones semanales para mirar el avance de las tareas del Sprint. De ser necesario se programarán más reuniones durante la semana en horarios en donde la mayoría de los integrantes estén presentes
- Durante la entrega del producto parcial en cada Sprint, no va a haber redefinición de los requerimientos por cambios de objetivos. Solo se redefinirá si existe errores sobre estos.
- No se tendrá en cuenta los roles que establece Scrum. Internamente en el grupo se acordaron roles según las necesidades del proyecto (ver sección “7.2. Organigrama y Descripción de Roles”)

## 4.2 Organigrama y Descripción de Roles

Los roles dentro del grupo de trabajo fueron acordados por el grupo teniendo en cuenta las necesidades del producto y su alcance. Teniendo en cuenta que la base del proyecto es la programación Back-End, los roles establecidos no son fijos, es decir que una persona puede desarrollar varios roles a la vez o cambiar de rol durante el desarrollo del proyecto, según las necesidades que se presenten en este. El único rol que siempre lo tendrá la misma persona es el de Gerente. A continuación, se muestra los roles que se van a ejecutar durante el proyecto:

- *Gerente*: Encargado de hacer seguimiento al proceso de desarrollo y del cumplimiento de los objetivos
- *Desarrollador Front-End*: Encargado del desarrollo de la interfaz de usuario y administrar los requerimientos asociados y su respectivo diseño.
- *Desarrollador Back-End*: Encargado del desarrollo de la lógica de negocio y administrar los requerimientos asociados (funcionales y no funcionales) y su respectivo diseño (arquitectura y diseño detallado).
- *Lider de Pruebas*: Encargado del diseño e implementación de las pruebas.
- *Lider de Infraestructura*: Encargado del soporte sobre cual el sistema correrá, la escalabilidad y rendimiento de este.



*Ilustración 1. Organigrama del proyecto*

## 5 Administración del Proyecto

### 5.1 Lenguajes y Herramientas

Para la infraestructura del sistema se desarrollará como una plataforma stand-alone, donde se considera que el computador del usuario final es el que cumple con todos los requerimientos para que las herramientas a utilizar corran sin ningún problema.

#### 5.1.1 Estándares de Desarrollo

##### 5.1.1.1. *Estándar en la Segmentación de Imágenes Médicas*

Para este tipo de estándar, se va a seguir las indicaciones de integración de las tres herramientas a utilizar. Estas herramientas son:

- TurtleSeg
- PTK
- MITK

##### 5.1.1.2. *Estándar en el Estilo del Código Fuente*

Para este estándar, se van a seguir las indicaciones dadas por la organización Mozilla para la correcta implementación del código fuente. Para ello se tendrán en cuenta principalmente tres indicadores:

- Indentación
- Formato
- Nombres

##### 5.1.1.3. *Estándar en la Gestión del Código Fuente*

Para este estándar, se van a seguir las indicaciones de Git-Flow para correcta gestión de configuración del código fuente. Para ello se tendrán en cuenta una de las siguientes guías:

- Feature Branches
- Release Branches

## 6 Monitoreo y Control del Proyecto

### 6.1 Administración de Requerimientos

#### 6.1.1 Priorización de los Requerimientos

Para priorizar los requerimientos, se tendrá en cuenta 4 aspectos que afectan cada requerimiento.

- Dificultad técnica (35%): Se tienen en cuenta aspectos como cambio de capas y conocimiento del lenguaje y las tecnologías necesario para implementar el requerimiento (curva de aprendizaje).
- Recursos (15%): Se refiere al número de recursos físicos o de personas que se necesitan para la implementación de un requerimiento en específico.
- Importancia para el cliente (20%): Los requerimientos serán priorizados teniendo en cuenta los criterios de los médicos, las investigaciones que se hayan realizado en la Universidad de Lyon y la opinión del profesor Leonardo Flórez
- Número de dependencias (30%): Es el número de dependencias que se necesitan para la implementación del requerimiento, es decir, las dependencias que tiene un requerimiento específico con otros requerimientos.

Teniendo en cuenta la priorización, los requerimientos se clasifican como se muestra en la siguiente tabla.

Medida de prioridad	Descripción
1	Si no se incluye en las primeras versiones del producto no se espera un impacto significativo en la satisfacción del cliente.
2	No afecta la satisfacción del cliente ni retrasa las versiones iniciales del producto.
3	El no incluir este tipo de requerimiento puede afectar la satisfacción del cliente, pero no se debe retrasar las primeras versiones del producto por la ausencia de uno de ellos.
4	Afecta la satisfacción del cliente y retrasa parcialmente versiones iniciales del producto.
5	Es un requerimiento básico, y necesario que debe incluirse en versiones finales del producto. Si no se implementa puede afectar la satisfacción del cliente y el correcto funcionamiento del sistema.

*Tabla 3. Priorización de requerimientos [7]*

### **6.1.2 Estados del requerimiento**

Cada requerimiento tiene 4 estados posibles [8]:

- Especificación: Se recolecta el requerimiento y se especifica, teniendo en cuenta las características de un buen requerimiento (consistente, completo, verificable, priorizables, trazable, correcto y no ambiguo)
- Diseño: es el cómo se va a implementar la funcionalidad en el sistema. Los requerimientos se encuentran en la etapa de arquitectura o diseño detallado
- Implementación: Es la elaboración de la funcionalidad en el sistema del requerimiento mencionado.
- Pruebas: Las pruebas son la última fase del requerimiento, en donde se verifica y se valida

### **6.2 Aceptación de producto**

Para que el sistema sea considerado como terminado se deben cumplir las siguientes condiciones en su totalidad:

- El sistema debe mostrar o generar imágenes posiblemente superpuestas con la TAC original que faciliten el proceso de detallar el estado del pulmón.
- El sistema debe integrar como mínimo dos herramientas de procesamiento de imágenes del conjunto de herramientas analizadas en la primera fase del proyecto.

## 7 Procesos de Soporte

### 7.1 Análisis de Riesgos

1. Solo se puede “Integrar” un algoritmo: El objetivo principal del sistema es integrar diferentes algoritmos de segmentación de imágenes médicas. Sin embargo, existe el riesgo que por las dificultades técnicas y las herramientas analizadas solo se logre utilizar un solo algoritmo. Dado que solo es un algoritmo ya no sería una integración.
  - a. *Disparador*: En la lista las herramientas seleccionadas en la 1ra fase no existe dos algoritmos que se pueden integrar a la vez o no se encontraron tecnologías que realicen esta función.
  - b. *Pérdida Asociada*: No lograr el objetivo del proyecto: Integrar algoritmo s
  - c. *Tratamiento*
    - i. *Preventivo*: Entre los criterios de selección tener uno que se relacione directamente con la integración
    - ii. *Mitigación*: Tener una lista de herramientas secundaria que tenga otras herramientas que se puedan integrar
    - iii. *Correctivo*: Volver a la primera fase y enfocar la búsqueda en dos algoritmos que se pueda integrar en el mismo sistema.
2. Coste de tiempo de pruebas muy alto: Dado que una de la problemática es el tiempo de procesamiento de imágenes médicas al momento de diagnosticar el SDRA, realizar pruebas sobre las imágenes de prueba puede tardar horas o días, por lo que el consumo de tiempo en las pruebas es muy alto.
  - a. *Disparador*: Complejidad de los algoritmos muy altas; interfaces de integración y comunicación lentos; Pruebas muy complejas.
  - b. *Pérdida Asociada*: El tiempo de comprobar la efectividad de un algoritmo es alto. Menos tiempo para corregir errores.
  - c. *Tratamiento*
    - i. *Preventivo*: Diseño de pruebas eficientes. Identificación de pruebas críticas con el fin de no desperdiciar tiempo en pruebas innecesarias.
    - ii. *Mitigación*: Rediseño de pruebas, selección de pruebas que puedan gas el menor tiempo posible
    - iii. *Correctivo*: Eliminar la etapa de pruebas. Solo realizar pruebas unitarias
3. Requerimientos técnicos muy altos: El procesamiento de imágenes requiere ciertos requerimientos técnicos mínimos para poder realizar el proceso. Se puede dar el caso que para lograr los requerimientos de desempeño se requiera requerimientos técnicos que la infraestructura que se tenga en el momento no alcance.
  - a. *Disparador*: Requerimientos no funcionales mal definidos. Tecnología mal seleccionada



- b. *Pérdida Asociada*: Adquisiciones innecesarias. Tiempo de búsqueda y adquisición de nuevos recursos.
  - c. *Tratamiento*
    - i. *Preventivo*: Realizar validación de los requerimientos no funcionales y de la arquitectura con el cliente y asesores externos (otros profesores de la Universidad)
    - ii. *Mitigación*: Adquirir equipos con especificaciones técnicas superiores a las necesarias según los requerimientos no funcionales y la arquitectura.
    - iii. *Correctivo*: Reducir los requerimientos no funcionales a las especificaciones técnicas de los equipos que se tiene al momento de que se presente el riesgo.
4. Herramientas con una curva de aprendizaje lenta: No solo es necesario aprender sobre las herramientas de desarrollo para realizar el proyecto. Es necesario conocer el funcionamiento de las herramientas de procesamiento de imágenes médicas para desarrollar un sistema que tenga una manipulación ante el usuario similar.
- a. *Disparador*: Encontrar herramientas que no se tenga el conocimiento para ser las evaluaciones de la 1ra fase; en el momento de la integración, no conocer las interfaces que proveen las herramientas seleccionadas
  - b. *Pérdida Asociada*: Tiempo que se invierte en el aprendizaje de las herramientas no se puede invertir en el desarrollo del proyecto. Si se presente en la fase de desarrollo (3ra fase) no existirá flexibilidad en los tiempos.
  - c. *Tratamiento*
    - i. *Preventivo*: Antes de comenzar el proyecto, realizar capacitación sobre las principales herramientas que se van a integrar y se van a utilizar para el desarrollo del proyecto.
    - ii. *Mitigación*: Realizar cursos y capacitaciones durante las 3 fases del proyecto.
    - iii. *Correctivo*: Realizar cursos como tiempo extra y enfocar las búsquedas al problema específico.

## 7.2 Métricas y Proceso de Medición

Las métricas que se tendrán en cuenta son basadas en el estándar para calidad de software ISO/9126. Estas métricas son [9]:

- **Funcionalidad**: Conjunto de atributos que miden la existencia de funcionalidades que cumplan con el objetivo. La funcionalidad se medirá por medio de que se haya integrado al menos dos algoritmos, funcionando de manera adecuada.
- **Usabilidad**: este atributo mide la capacidad del sistema para ser entendido, aprendido y usado. Se medirá con la cantidad de horas de aprendizaje que le tome a un usuario objetivo nuevo en aprender a usar el sistema sin ayuda de un experto.
- **Mantenibilidad**: conjunto de atributos que miden el esfuerzo necesario para hacer modificaciones. Este atributo se medirá por medio del impacto que tiene un cambio a

nivel de código, a través de la cantidad de dependencias que se den entre diferentes componentes

- **Eficiencia:** conjunto de atributos que miran la relación entre el nivel de rendimiento del software y la cantidad de recursos usados, dado ciertas circunstancias. Esto se medirá por medio del tiempo de respuesta del sistema comparado con el tiempo de respuesta de los softwares investigados
- **Portabilidad:** conjunto de atributos que representan la habilidad del software de ser movido de un ambiente a otro. Dado a que el sistema siempre será usado en un computador de escritorio, y trabajará como un sistema stand-alone, este atributo no se tomará en cuenta.
- **Confiabilidad:** conjunto de atributos relacionados con la habilidad del software de mantener su nivel de rendimiento dado ciertas circunstancias bajo un determinado tiempo. En vista de que los algoritmos a usar ya están implementados y se consideran tolerantes a fallas dado entradas correctas, la confiabilidad no se medirá.

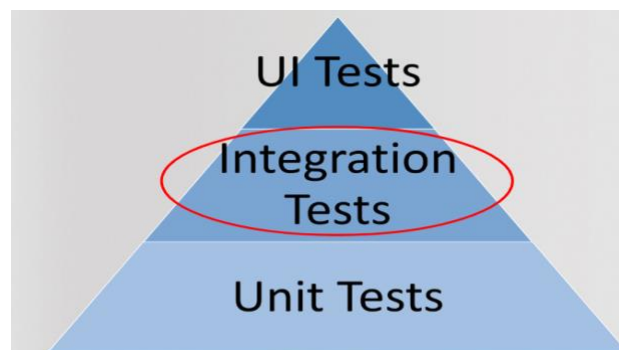
## 7.3 Control de Calidad

### 7.3.1 Estrategia de pruebas

El test strategy o la estrategia de pruebas expresa los requisitos genéricos para probar uno o más proyectos de software, proporciona detalles sobre cómo se realiza la prueba, los objetivos y tipos de pruebas, en general el acercamiento [10]. Siguiendo estas bases y para mantener el control de calidad se plantean los siguientes componentes a tener en cuenta para desarrollar en el proceso de pruebas del software del proyecto.

### 7.3.2 Niveles de pruebas

En la siguiente figura se puede observar los niveles básicos de pruebas de software, estos niveles buscan enfocar las pruebas en componentes específicos del software, partiendo desde las pruebas unitarias del código hasta llegar a la simulación de la interacción con el usuario por medio de las distintas interfaces del software y validar las funcionalidades, completas, del sistema. Para el desarrollo y validación del proyecto se hará especial énfasis en las pruebas de integración con las herramientas de procesamiento de imágenes seleccionadas.



*Ilustración 2. Niveles de pruebas*

- **Unit testing:** También llamado nivel de pruebas unitarias, es donde proceso de pruebas de software se enfoca en las unidades individuales de un software.
- **Integration testing:** O pruebas de integración es un nivel del proceso de pruebas del software donde las unidades individuales se combinan y se prueban como un grupo. El propósito de este nivel de prueba es exponer fallas en la interacción entre unidades integradas.
- **UI testing:** Este nivel busca probar el sistema completo o por funcionalidades precisas a partir de las interfaces expuestas por el software. El propósito de esta prueba es evaluar el cumplimiento del sistema con los requisitos especificados.

### 7.3.3 Tipos de pruebas

- **Pruebas funcionales:** Estas pruebas son realizadas con el fin de evaluar la conformidad de un componente o sistema con los requisitos funcionales [10], como los casos de uso del software que exponen una acción directa del usuario con el sistema. El proceso también se enfocará en probar la interoperabilidad del sistema que evalúa la capacidad del software para interactuar con uno o más componentes o sistemas específicos.
- **Pruebas no funcionales:** Estas pruebas son realizadas con el objetivo de evaluar el cumplimiento de un componente o sistema con requisitos no funcionales, como disponibilidad, rendimiento, portabilidad, entre otros [10]. Para el caso del proyecto se establecerá un foco en los tiempos de respuesta que le tome a cada una de las herramientas seleccionadas procesar determinadas imágenes, de esta manera tener un criterio para poder reajustar la selección de dichas herramientas.

### 7.3.4 Técnicas de diseño de pruebas

- **Black-box:** También definidas por el International Software Testing Qualifications Board (ISTQB) como técnicas basadas en la especificación, son un procedimiento para derivar y / o seleccionar casos de prueba basados en un análisis de la especificación, ya sea funcional o no funcional, de un componente o sistema sin referencia a su estructura interna [10].

Es decir, en esta técnica no se contempla el análisis de código sino por el contrario, se toman componentes donde se brinda un conjunto de entradas y se espera una o varias respuestas, sin hacer un seguimiento del proceso que los componentes involucrados estén efectuando.

- **Tabla de decisiones:** Esta técnica especifica cómo diferentes combinaciones de condiciones producen diferentes resultados [10]. De esta forma se puede identificar condiciones (datos de entradas) y acciones resultantes (datos de salida) del sistema, así se podrá tener un control de los posibles flujos que se generen para el desarrollo del sistema.

- **Casos de uso:** Cada caso de uso especifica algún comportamiento que un sujeto puede realizar en colaboración con uno o más actores. Un caso de uso puede describirse mediante interacciones y actividades, así como condiciones previas y condiciones posteriores [10]. Con esta técnica basada en los casos de uso se planea seleccionar las funcionalidades específicas a evaluar, teniendo como base aquellas que mayor integración con otros componentes contemplen.

## 8 Referencias

- [1] «ARDS Symptoms, Causes & Risk Factors». [En línea]. Disponible en: <http://www.lung.org/lung-health-and-diseases/lung-disease-lookup/ards/symptoms-causes-risk-factors.html>. [Accedido: 11-ago-2018].
- [2] G. Dougherty, *Digital image processing for medical applications*. California, United State of America: Cambridge University Press, 2009.
- [3] CARLOS GABRIEL GAVIDIA CALDERÓN, «SEGMENTACIÓN DE IMÁGENES MÉDICAS MEDIANTE ALGORITMOS DE COLONIA DE HORMIGAS». 2014.
- [4] «¿Qué es tomografía axial computarizada?» [En línea]. Disponible en: <https://www.cun.es/diccionario-medico/terminos/tomografia-axial-computarizada>. [Accedido: 24-oct-2018].
- [5] M. Torres, «Plantilla SRS», 2015. [En línea]. Disponible en: [http://uvirtual.javeriana.edu.co/webapps/portal/frameset.jsp?tab\\_group=courses&url=%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2Ffile%3Fcmd%3Dview%26content\\_id%3D\\_322100\\_1%26course\\_id%3D\\_7108\\_1%26framesetWrapped%3Dtrue](http://uvirtual.javeriana.edu.co/webapps/portal/frameset.jsp?tab_group=courses&url=%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2Ffile%3Fcmd%3Dview%26content_id%3D_322100_1%26course_id%3D_7108_1%26framesetWrapped%3Dtrue).
- [6] K. Schwaber y J. Sutherland, «The Scrum Guide, The Definitive Guide to Scrum: The Rules of the Game», nov-2017.
- [7] A. P. García, *Software Requirements Specification*. Bogotá, 2013.
- [8] S. Alexandre, *Paquete de Despliegue Análisis de Requerimientos de Software Perfil Básico*. Belgica: Luis García – Universidad Peruana de Ciencias Aplicadas (Perú), 2013.
- [9] «ISO/IEC 9126-1:2001 - Software engineering -- Product quality -- Part 1: Quality model». [En línea]. Disponible en: <https://www.iso.org/standard/22749.html>. [Accedido: 31-oct-2018].
- [10] International Software Testing Qualifications Board, «Certified Tester: Foundation Level Syllabus», 2018. [En línea]. Disponible en: <https://www.istqb.org/downloads/send/51-ctfl2018/208-ctfl-2018-syllabus.html>.