

Avance 1

Carolina Campos, Juan Manuel Guevara, Jose Ricardo Rivera

I. PROPÓSITO DEL SISTEMA

El propósito del sistema es desarrollar una aplicación web que permita automatizar el conteo celular en una serie de imágenes, para que el mismo pueda ser usado en el Laboratorio de Quimiosensibilidad tumoral de la Universidad de Costa Rica.

II. ALCANCE DEL SISTEMA

El alcance estará dado de la siguiente manera:

- El usuario podrá cargar una imagen (en un formato definido) digital de alguna dirección del ordenador.
- Aplicar la convolución a una imagen.
- Aplicar el filtro gaussiano a una determinada imagen.
- Aplicar la ecualización de histogramas para mejorar el contraste de una imagen.
- Aplicar un filtro bilateral a una imagen.
- Medir el tiempo de ejecución de una determinada función o del procesamiento de imagen en general.
- Generar una imagen preprocesada.

El sistema será capaz de poder analizar imágenes de células en un formato o formatos definidos, dichas imágenes pasaran por una serie de análisis para que al final quede lo más nítida posible. Se espera que con el desarrollo de la aplicación se agilice el proceso de procesamiento de imágenes para el Laboratorio de Quimiosensibilidad tumoral de la Universidad de Costa Rica y así tengan un software que realice solamente lo que ellos ahí necesitan, sin funcionalidades extra y que sea web, no como la que tienen actualmente que es de escritorio.

III. CONTEXTO DEL SISTEMA

Entre los elementos principales del sistema se encuentran el componente de la carga de imágenes, el aplicador de la convolución a una imagen dada, el filtro gaussiano y bilateral y la ecualización de histogramas requerida. Los usuarios que interactuarán con el sistema, los cuales serán los doctores Steve Quirós, Rodrigo Mora y otro grupo de personas que forman parte del equipo del laboratorio. Estos elementos se interactúan de la siguiente manera: el usuario iniciará el proceso cargando la imagen respectiva que desea que se analice, luego de que la imagen se carga correctamente se procede a procesarla, aplicándole la convolución, el filtro gaussiano y bilateral y la ecualización de histogramas, el usuario también podrá personalizar parámetros como la desviación estándar y el tamaño de la ventana, al final se mostrará la imagen ya procesada y el tiempo que tardó en ejecutarse todos los algoritmos para obtener el resultado deseado, el usuario decide si quiere descargar la imagen o no.

IV. FUNCIONES DEL SISTEMA

El sistema será capaz de qué, dada una imagen cargada por el usuario se pueda realizar una serie de algoritmos para quitar todos los obstáculos o ruido que estas presenten. Entre las limitaciones que presentará el sistema se tiene que al ser una aplicación web se dependerá completamente de internet, en caso de que no haya conexión no se puede hacer uso de la misma.

V. CARACTERÍSTICAS DE USUARIO

La aplicación será para un grupo de personas que trabajan en la Universidad de Costa Rica, pero al ser web queda expuesta a que la puedan utilizar quienes lo necesiten. Se debe de tomar en cuenta las siguientes características de usuario a la hora de crear la aplicación:

- Conoce poco de computación.
- Tiene alto conocimiento en el dominio.

Las características anteriores son importantes para que el usuario pueda adaptarse al sistema con facilidad, se debe crear una aplicación que se comprenda fácilmente y que el usuario no demore mucho en comprender (al menos lo más básico) para hacer uso de esta.

VI. REQUERIMIENTOS FUNCIONALES

En los requerimientos funcionales del sistema se encuentran los siguientes puntos:

ID	Requerimiento	Prioridad
01	El sistema será capaz de cargar la imagen deseada por el usuario.	Alta
02	Visualizar la imagen cargada.	Media
03	Aplicar el filtro gaussiano.	Alta
04	Aplicar la ecualización de histogramas.	Alta
05	Aplicar el filtro bilateral.	Alta
06	Mostrar el tiempo de ejecución de todos los algoritmos.	Media
07	Generar la imagen con todos los algoritmos.	Alta
08	Mostrar el resultado (la imagen procesada)	Alta
09	Opción de descargar la imagen procesada.	Alta
10	Reportar los errores producidos en la ejecución del programa	Baja
11	Capacidad de identificar cuando sucede un error en el tiempo de ejecución.	Baja
12	Opción de personalizar los parámetros como desviación estándar y tamaño de la ventana.	Alta

VII. REQUISITOS DE USABILIDAD

La idea es que el sistema sea fácil de usar, que no presente ninguna complejidad que demore tiempo en entender cómo funciona, por lo que se definieron los siguientes requerimientos:

ID	Requerimiento	Prioridad
01	Los usuarios (con experiencia) deberán utilizar las funcionalidades del sistema en un tiempo promedio de 1 minuto.	Alta
02	Los usuarios nuevos podrán aprender a usar el sistema en un tiempo promedio de 5 minutos.	Alta
03	Realizar un sitio web que tenga instrucciones claras y concisas.	Media
04	Realizar un sitio web en el cual el diseño sea amigable para el usuario y facilite la comprensión de su uso.	Media
05	Implementación de un manual de usuario, para que así el usuario pueda consultar cualquier duda respecto a alguna funcionalidad de la aplicación.	Alta

VIII. REQUISITOS DE DESEMPEÑO

En los requerimientos de desempeño del sistema se encuentran los siguientes:

ID	Requerimiento	Prioridad
01	El tiempo de respuesta de la página web debe ser muy bajo, se espera, que sea aproximadamente 1 segundo.	Alta
02	Se contratará un técnico en mantenimiento (en caso necesario) para que pueda operar todo lo relacionado a la mantenibilidad del sistema o alguna situación en específico.	Baja

IX. INTERFACES DEL SISTEMA

Para el sitio web se utilizará el protocolo Hypertext Transfer Protocol (HTTP) el cual permite la transferencia de información en la World Wide Web.

X. REQUERIMIENTOS DE INTEGRACIÓN DEL SISTEMA HUMANO

El sistema será diseñado para ser usado por cualquier tipo de usuario, esto porque no presentará ninguna complejidad en su operación y además de eso se hará un sistema que sea de fácil uso y por ende de fácil aprendizaje para los usuarios nuevos.

XI. MANTENIBILIDAD

En cuanto a la mantenibilidad se han definido los siguientes requerimientos:

ID	Requerimiento	Prioridad
01	Se espera que el sistema esté en línea un 90 % del tiempo, tomando en cuenta que son 365 días por año, se establece que 328 días estará en perfecto funcionamiento.	Alta
02	Se contratará un técnico en mantenimiento (en caso necesario) para que pueda operar todo lo relacionado a la mantenibilidad del sistema o alguna situación en específico.	Baja

XII. CONFIABILIDAD

Siendo un fallo a cualquier defecto, físico o lógico, en cualquier componente hardware o software del sistema. Existen fallos que no se tomaran en cuenta, los cuales son los siguientes: contactos accidentales entre conductores eléctricos, cortes en los mismos, defectos en los componentes, variaciones en el funcionamiento de los elementos electrónicos debidas a perturbaciones externas. Para los fallos no mencionados que se

incluye en el sistema se tiene previsto la prevención y de ser necesario la recuperación con el objetivo de reducir la latencia de los fallos. Los cuales se categorizan como funcionamiento incorrecto de la aplicación se pueden mencionar el usuario ingresando información incorrecta, la información siendo mal procesada o la mala manipulación de los datos.

XIII. ESTADOS Y MODOS DEL SISTEMA

Se adjuntan en los anexos los siguientes diagramas:

- Figura 1. Diagrama de componentes.
- Figura 2. Diagrama de clases.

XIV. CARACTERÍSTICA FÍSICAS

Al ser un sistema implementado con características web lo único necesario para darle uso a este sistema ser a un computador o computadores los cuales deberán ser proveídos por el laboratorio de Quimiosensibilidad tumoral, de la Universidad de Costa Rica. No será necesario colocar algún tipo de servidor físico ya que el mismo será montado sobre la plataforma de Microsoft Azure, la cual es un servicio en la nube que nos ayudara a almacenar y montar la base de datos en la nube. Microsoft Azure ofrece en los niveles de servicio Básico, Estándar y Premium, se usará el servicio Estándar, el cual es el más ideal para el sistema que se está implementando, una denición de este servicio es la siguiente: Estándar: Es la opción predilecta para la mayoría de aplicaciones en la nube, admite varias consultas simultaneas. Entre los ejemplos se incluyen aplicaciones web o para grupos de trabajo.

XV. REQUERIMIENTOS FÍSICOS

El sistema está diseñado para poder ser extendible, para poder agregar funcionalidades en un futuro, entre los requerimientos se encuentran:

ID	Requerimiento	Prioridad
01	El sistema debe implementar patrones de diseño que permitan la adaptación de nuevas interfaces internas o externas.	Alta
02	. El sistema debe estar disponible para cualquier explorador de internet utilizado en el laboratorio.	Alta

XVI. CONDICIONES AMBIENTALES

No aplica. Ninguna variable ambiental entre las cuales se menciona variables naturales, medios inducidos y medio auto-inducidos serán tomadas en cuenta en el sistema debido que el equipo ni la empresa se responsabiliza por algún daño o alteración del sistema debido a las citadas anteriormente, así se aclara que el Laboratorio de Quimiosensibilidad tumoral, de la Universidad de Costa Rica, deberá tomar las medidas adecuadas para resolver algún problema debido a las condiciones ambientales.

XVII. SISTEMA DE SEGURIDAD

La seguridad de un sistema forma parte de unas de las características más relevantes de un sistema, y al ser un sistema construido para una entidad de la Universidad de Costa Rica muy importante, la seguridad de este sistema debe de ser controlada de la siguiente manera:

- Proteger la información de los datos obtenida de las imágenes cargadas por el usuario. Validar que los usuarios que ingresan al sistema sean usuarios con los privilegios correctos, en este caso solamente puede ser ingresado por los miembros del Laboratorio de Quimiosensibilidad tumoral, de la Universidad de Costa Rica.
- Crear y validar que el acceso a bases de datos y robo o corrupción de datos personales o condenciales no pueda ser realizado por ninguna persona sin los permisos necesarios.
- Crear protocolos de seguridad que permita alertar si hay alguna Intercepción de datos personales y condenciales.
- Uso de herramientas que permitan controlar los Ataques de denegación de servicio (DoS) que deshabilitan la disponibilidad de los servicios.

XVIII. GESTIÓN DE LA INFORMACIÓN

La información del sistema va a ser manejada por medio de la interacción entre un cliente-servidor el cual va ser manejado por el Protocolo de transferencia de hipertexto (HTTP) el cual esta diseñado para manejar este tipo de comunicaciones. Por lo tanto, se van a realizar dos métodos comúnmente utilizados para una petición-respuesta entre un cliente y el servidor, estos son:

- Get - pide datos del servidor.
- Post - los datos se envían al servidor.

Además, toda esta información es manejada por medio de un tipo de archivo llamado JSON (JavaScript Object Notation) el cual es un formato para el intercambio de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos que van a ser intercambiados por el cliente y el servidor. Asimismo, la información que va a ser enviada y recibida por el servidor y el cliente van a seguir un protocolo de seguridad de información más especificada en la sección de seguridad del sistema.

XIX. POLÍTICAS Y REGLAMENTOS

Se dene las siguientes políticas tomadas en consideración:

- Existen solo una serie especifica de usuarios que puede acceder al sistema.
- De manera automática obtener las imágenes procesadas por el usuario.
- Generar un informe con el lote de imágenes procesadas y que procedimiento se utilizó para cada una de ellas
- De manera automática cuando tardara el procesar un lote de imágenes anteriormente seleccionadas.
- Generar un informe con las imágenes que no se han podido procesar y dar un resultado.

XX. SOSTENIMIENTO DEL CICLO DE VIDA

Es importante destacar que en el proceso de desarrollo del sistema se mantendrá organizado el progreso por medio del control de versiones, además se tendrá el estándar definido por la empresa para tener el nivel de documentación correctos. Además se realizarán actividades de evaluación al finalizar cada sprint con el fin de aumentar la calidad que se pueda ofrecer, estas actividades se mencionan en detalle en la sección Actividades de ACS.

XXI. EMBALAJE, MANIPULACIÓN, ENVÍO Y TRANSPORTE

No aplica. Ya que nuestro sistema no deberá de tener algún tipo de manipulación y entrega física, ya que la mayor parte de este sistema es manejado en la nube.

XXII. VERIFICACIÓN

Las verificaciones que se realizarán al sistema ya fueron realizadas en otro documento, en dicho documento se explican cuáles son las métricas que se usarán para verificar que la calidad de los elementos dentro del sistema sea la mejor, se puede encontrar esta sección dentro de Anexos: Métricas.

XXIII. SUPOSICIONES Y DEPENDENCIAS

Dentro de las suposiciones contamos con los siguientes puntos:

- El laboratorio de Quimiosensibilidad tumoral de la UCR tendrá el equipo necesario para el uso del sistema dentro de su organización.
- El laboratorio de Quimiosensibilidad tumoral cuenta con el ancho de banda de internet requerido.
- La imagen de entrada deberá tener el formato correcto.
- Se contará con una base de datos sobre la plataforma de Microsoft Azure para contar con un registro de los datos analizados.
- El laboratorio de Quimiosensibilidad tumoral cuenta con los recursos necesarios para hacer los pagos de los servicios de Microsoft Azure.

Dentro de las dependencias contamos con los siguientes puntos:

- El uso del sistema dependerá de la estabilidad del servicio de Microsoft Azure
- El laboratorio de Quimiosensibilidad tumoral de la UCR debe contar con el servicio eléctrico estable para el buen funcionamiento de las computadoras que utilizan el sistema.

XXIV. JUSTIFICACIÓN DEL ESTÁNDAR DE CODIFICACIÓN

XXIV-A. Introducción

Gracias a que se debe mantener un código legible y entendible a travez de los sistemas de software, surge la importancia de seguir un estándar de codificación que asegure una edición o modificación fácil y accesible para los desarrolladores que analizarán el código en algún momento dado, y que en muchos casos será otro programador. Por lo tanto, el equipo tendrá que seguir los siguientes estándares de codificación presentados a continuación.

XXIV-B. Estandar de codificación Java

El estándar elegido para la creación del software es el establecido por la empresa Google, el cual expresa ampliamente la forma de codificación en el lenguaje Java. La guía propuesta toma aspectos no encontrados en otros estándares como lo son los try and catch o Javadoc, los cuales son importantes para lograr medir por ejemplo la documentación interna del código. El estándar de Google se acomoda a las necesidades del proyecto ya que sigue principios de código limpio como la declaración adecuada de variables con nombre significativos. Gracias a esto, al seguir esta guía, se garantiza la calidad del código del producto para su mejor mantenimiento. Referencia: Google Java Style Guide

XXIV-C. Estándar de codificación Javascript

Se analizaron las guías más populares incluyendo la presentada por Google y Crockfords. En comparación a todas las observadas Airbnb se distingue debido a su completitud y facilidad de aprendizaje del estándar. Los ejemplos que se encontraron en el repositorio en el cual se encuentra el estilo, muestran explícitamente la forma incorrecta de escribir el código adjuntando seguidamente la correcta, lo que simplifica identificar si el desarrollador está realizando la codificación correctamente, de esta manera se llegó a la conclusión de que esta era la característica más relevante para la decisión tomada. Por otro lado, este estándar utiliza como referencia en su elaboración el estilo de Google e Idiomatic JavaScript, el cual se puede obtener referencia para la búsqueda de frameworks que permitan la generación de test unitarios como lo son QUnit o Jest. Referencia: Airbnb JavaScript Style Guide

XXV. DIAGRAMA DE COMPONENTES

En este diagrama de componentes se representa la distribución y conexión entre los componentes que integran el sistema. De parte del cliente, debido a la naturaleza de la aplicación web, únicamente se contiene el navegador por el cual se accede a la aplicación. Por otro lado, dentro del servidor se contendrá el web service que el cliente podrá consumir para lograr procesar una imagen. Este service ofrecerá la página de interfaz index.jsp donde se realizarán a través de angularJS las peticiones Get y Post correspondientes para mostrar una aplicación dinámica. Además, se accederán a los servlets por el mismo service para el procesamiento del video mediante openCV. Se crea un componente especial para el ingreso de los usuarios autorizados. Como último componente relevante del diagrama se coloca la base de datos, la cual se encontrará dentro del mismo servidor. Se anexa la imagen con el diagrama: Diagrama de componentes de sistema

XXVI. DIAGRAMA DE CLASES

El diagrama de clases representa en su mayoría la parte del Back-end que se implementará para el proyecto. Se utilizará una clase para preprocesar la imagen, la cual devolverá una imagen ya "arreglada" para que el procesador aplique los filtros requeridos para la generación de la imagen final, esta clase estará conectada a la clase principal "Procesador" que

se encargará de todos los métodos de aplicación de métodos sobre la imagen. La selección del método se hará a través de una clase tipo enumeration "Metodo" y se utiliza el patrón MVC en el proyecto para mantener un orden en el código. Se anexa la imagen con el diagrama: Diagrama de clases del sistema

XXVII. TABLA DE ACTIVIDADES DE ACS

Actividad	Responsable
Revisión de la construcción de los requerimientos del sistema establecidos con el estándar ISO/IEC/IEEE 29148-2011	Juan Manuel Guevara, Jose Rivera, Carolina Campos
Revisión y análisis de los requerimientos del cliente	Jose Rivera
Revisión de la construcción de los requerimientos del sistema basado en los atributos del ISO-9126	Juan Manuel Guevara
Evaluación de la implementación de código que siga los estándares de codificación	Carolina Campos, Juan Manuel Guevara
Revisión y evaluación de la herramienta de control de cumplimiento de estándares de codificación	Juan Manuel Guevara, Carolina Campos, Jose Rivera
Identificar los objetivos que cumplen el estándar de código según la orientación del sistema	Juan Manuel Guevara, Jose Rivera
Valoración de los principios de diseño acorde al diagrama de clases	Juan Manuel Guevara
Evaluación de la relación del diseño de componentes según los requerimientos planteados	Jose Rivera
Construir una validación de ingreso de usuarios válidos del sistema	Jose Rivera
Analizar los riesgos técnicos que se pueden experimentar	Carolina Campos
Inspeccionar la integración de los patrones de diseño implementados	Juan Manuel Guevara, Jose Rivera
Revisión y evaluación de los diagramas UML que detallan la arquitectura del sistema	Carolina Campos
Crear pruebas para estresar el componente a su máximo nivel	Juan Manuel Guevara
Construir un conjunto de pruebas unitarias para la validación de funcionamientos específicos del sistema	Carolina Campos, Juan Manuel Guevara
Revisar y analizar las respuestas obtenidas de las pruebas unitarias del sistema	Juan Manuel Guevara, Jose Rivera
Construir un conjunto de pruebas de integración para la validación de módulos en conjunto del sistema	Juan Manuel Guevara, Carolina Campos, Jose Rivera
Revisar y analizar las respuestas obtenidas de las pruebas de integración del sistema	Carolina Campos, Juan Manuel Guevara
Construir un conjunto de pruebas de aceptación para la validación de la funcionalidad completa del sistema	Juan Manuel Guevara, Carolina Campos, Jose Rivera
Revisar y analizar las respuestas obtenidas de las pruebas de aceptación del sistema	Juan Manuel Guevara
Realizar una revisión de los tiempos de respuesta de la página web sean bajos	Carolina Campos, Juan Manuel Guevara, Jose Rivera

XXVIII. ANEXOS

XXVIII-A. Anexo A: Diagrama de clases

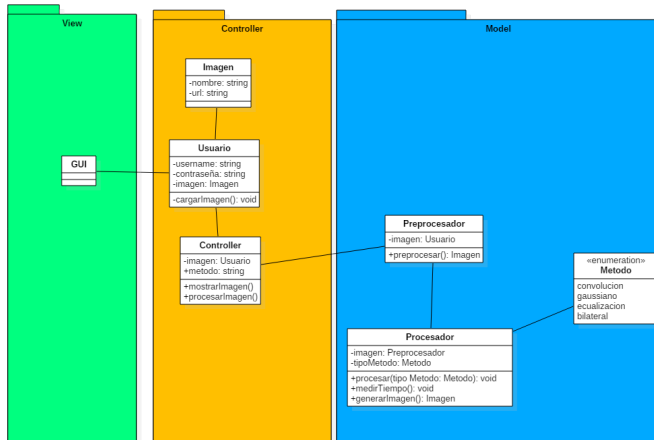


Figura 1: Diagrama de clases del sistema

XXVIII-B. Anexo A: Diagrama de componentes

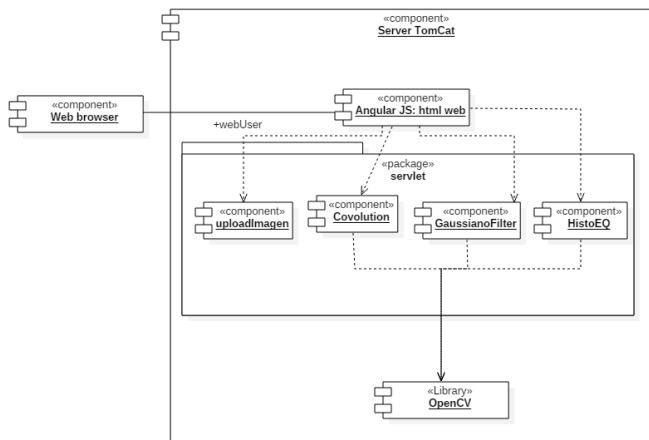


Figura 2: Diagrama de componentes del sistema

XXIX. MÉTRICAS

XXIX-A. Categorías

XXIX-A1. Funcionabilidad:

XXIX-A1a. *Seguridad*: La métrica es Interna: Utilizada para analizar el código fuente desarrollado, para así determinar si existen errores en el mismo.

El nivel requerido es: No debería de existir ningún error o descontento crítico.

La herramienta necesaria es: Fortify software.

XXIX-A1b. *Adecuación*: La métrica es Interna: Evaluar el promedio de cantidad de tareas que fueron asignadas, de acuerdo a los requerimientos más importantes del sistema.

El nivel requerido es: Deben existir 3 o más tareas relacionadas con el requerimiento.

La herramienta necesaria es: Zoho ReqView.

XXIX-A1c. *Exactitud*: La métrica es Interna: Se obtendrá el porcentaje según la cantidad de funcionalidades correctas en un conjunto de 10 funcionalidades.

El nivel requerido es: El porcentaje deberá ser mayor al 85 %.

La herramienta necesaria es: GroundTruth

XXIX-A2. Eficiencia:

XXIX-A2a. *Comportamiento con el tiempo*: La métrica es Interna: El tiempo promedio que tarda el sistema en dar respuesta al usuario sobre una petición específica.

El nivel requerido es: El tiempo promedio debe ser menor a dos segundos.

La herramienta necesaria es: Jmeter.

XXIX-A2b. *Utilización de recursos*: La métrica es Interna: La cantidad de líneas de código usadas para generar una determinada función, debe ser lo menos posible.

El nivel requerido es: No excederse en las líneas de código, entre menos (pero funcional y eficiente) mejor.

La herramienta necesaria es: EclEmma.

XXIX-A2c. *Conformidad del reconocimiento*: La métrica es Interna: Analizar el código realizado para verificar que no existan olores de software.

El nivel requerido es: No deberá existir ningún error que sea clasificado como crítico o mayor.

La herramienta necesaria es: Sonarqube

XXIX-B. Fiabilidad

XXIX-B0a. *Tolerancia a fallos*: La métrica es Externa: Se le debe informar al usuario en caso de algún error que se presente en tiempo ejecución o en la medida de lo posible tratar de solucionarlos internamente.

El nivel requerido es: No cerrar la ejecución del programa a menos que el usuario así lo indique.

La herramienta necesaria es: Junit.

XXIX-B0b. *Madurez*: La métrica es Externa: El sistema presenta mínimas o en la medida de lo posible ningún falló durante su ejecución.

El nivel requerido es: El sistema funcione perfectamente sin errores e interrupciones.

La herramienta necesaria es: Junit.

XXIX-B0c. *Cumplimiento de fiabilidad*: La métrica es Externa: Al desarrollar el programa se cumplieron los estándares que se propusieron.

El nivel requerido es: La documentación generada cumple con las características propuestas

La herramienta necesaria es: Investigación de documentación.

XXIX-C. Usabilidad

XXIX-C0a. *Facilidad de aprendizaje*: La métrica es Externa: Una de las métricas de calidad externas en la facilidad de aprendizaje es hacer una medición de los tiempos que

duran los miembros del laboratorio de quimiosensibilidad tumoral de la Universidad de Costa Rica por cada una de las funcionalidades de la página web por primera vez.

El nivel requerido es: Los tiempos de aprendizaje sean bajos, entre un rango de 10 a 20 minutos.

La herramienta necesaria es: Cualquier herramienta con un cronómetro.

XXIX-C0b. Operabilidad: La métrica es Externa: Una de las métricas de calidad externas en la operabilidad es hacer una encuesta a los miembros del laboratorio de quimiosensibilidad tumoral de la Universidad de Costa Rica y saber que tan difícil les fue utilizar todas las funcionalidades del sistema.

El nivel requerido es: Que haya un nivel de satisfacción alto (Rango del 1 al 10) en la operabilidad del sistema.

La herramienta necesaria es: Cualquier herramienta para hacer encuestas.

XXIX-C0c. Atracción: La métrica es Externa: Una de las métricas de calidad externa en la atracción es que la página web es hacer una encuesta a los miembros del laboratorio de quimiosensibilidad tumoral de la Universidad de Costa Rica y saber que tan fácil les fue leer la información del análisis del imágenes en la página web.

El nivel requerido es: Que haya un nivel de alto en la lectura de los datos (Rango del 1 al 10) en la atracción del sistema.

La herramienta necesaria es: Cualquier herramienta para hacer encuestas.

XXIX-D. Portabilidad

XXIX-D0a. Adaptabilidad: La métrica es Externa: Una de las medidas de calidad externas en la adaptabilidad es que todos los miembros del laboratorio de quimiosensibilidad tumoral de la Universidad de Costa Rica tengan la facilidad de poder visualizar las funcionalidades del sistema desde sus computadoras, en este caso la aplicación web. Por lo tanto que puedan usar cualquier navegador web y puedan visualizar estas funcionalidades con facilidad.

El nivel requerido es: Visualización de la aplicación en todos los navegadores web.

La herramienta necesaria es: Cualquier navegador web: Google Chrome, Firefox, Safari, entre otros.

XXIX-D0b. Capacidad de instalación: La métrica es Externa: Una de las medidas de calidad externas en la capacidad de instalación es ver la facilidad en que los miembros del laboratorio de quimiosensibilidad tumoral de la Universidad de Costa Rica pudieron instalar el sistema en sus computadoras de los laboratorios.

El nivel requerido es: Poder instalar el sistema sin ningún tipo de problema, en un tiempo de no más de 10 minutos.

La herramienta necesaria es: Cualquier herramienta para hacer encuestas.

XXIX-D0c. Coexistencia: La métrica es Externa: Una de las medidas de calidad internas en la coexistencia es que el servidor de la página web pueda funcionar con una

herramienta que pueda detectar y bloquear cualquier tipo de ataque DDOS.

El nivel requerido es: Que la coexistencia de otras herramientas no afecten cualquier funcionamiento del sistema.

La herramienta necesaria es: Log2ban es una herramienta que permite la detección y bloqueo de IPs que participan en ataques DDOS.

XXIX-E. Mantenibilidad

XXIX-E0a. Estabilidad: La métrica Externa: El sistema mantiene su integridad si se modifica algún componente. La métrica Interna: La implementación presenta alta cohesión y bajo acoplamiento entre sus componentes.

El nivel requerido es: La modificación no afecta el funcionamiento de otros componentes.

La herramienta necesaria es: Sonarqub

XXIX-E0b. Capacidad de instalación: La métrica es Interna: El diseño implementado cumple con los principios SOLID relacionados con la programación orientada a objetos.

El nivel requerido es: Se observa la posibilidad de extensión del diseño para la integración de nuevos requerimientos.

La herramienta necesaria es: Visual Paradigm 14.0

XXIX-E0c. Análisis: La métrica es Interna: El código escrito presenta los siguientes olores relacionados con software: Código duplicado o “muerto”, poca cohesión a nivel de clases y paquetes, alto acoplamiento, escritura ambigua, poca documentación.

El nivel requerido es: El código presente cumple con la mayoría de buenas prácticas de programación.

La herramienta necesaria es: Sonarqub.