

Aseguramiento de la Calidad del Software

Avance #1

André Arroyo Piedra
2015073657

Bryan Vargas
2015011562

Juan Villacis
2016201681

13 de septiembre del 2018

1. Introducción

En el presente documento se describen aspectos relacionados con los requerimientos, el estilo, la estructura y una pequeña funcionalidad del sistema. Con respecto a los requerimientos se realiza el SRS, que describe las necesidades del Laboratorio de quimiosensibilidad tumoral de la Universidad de Costa Rica. En este documento se especifican de una manera detallada para los desarrolladores estas necesidades, lo que ayudará a desarrollar un sistema que sea útil para esa entidad.

El estilo que tendrá el código escrito para la herramienta también se establece, luego de una comparación de los principales estilos de codificación existentes. Esto ayudará a incrementar la mantenibilidad del sistema en el futuro. Los diagramas de componentes y de UML elaborados ayudan a describir de una mejor manera el sistema, lo que será útil tanto en las etapas de desarrollo como de mantenimiento de la herramienta. Por último se describen las pruebas unitarias realizadas para la funcionalidad de carga y presentación de imágenes.

2. Repositorio

Es posible tener acceso al código del proyecto en la siguiente dirección web <https://github.com/4a75616e/segmentator>

3. Documento de Requerimientos

3.1. Propósito del proyecto

La razón para crear este proyecto es brindar una herramienta que permita a los investigadores del laboratorio de quimiosensibilidad tumoral tener una ayuda en el proceso de segmentación de células. Esa segmentación tiene como objetivo determinar si existe la enfermedad del cáncer en el tejido analizado; y si existe entonces poder determinar el grado de avance.

3.2. Alcance del proyecto

El sistema que se desarrollará, de nombre "Herramienta para la segmentación de células", tiene como objetivo solucionar las dificultades que se le presentan a los microbiólogos del laboratorio de quimiosensibilidad tumoral de la UCR al tener que analizar las imágenes de tejidos celulares manualmente en busca de la existencia del cáncer. Los inconvenientes con el sistema actual radican en las dificultades de continuar con los estudios y atención a los pacientes que se generan por la gran cantidad de imágenes a analizar.

El sistema implementará un pipeline que permita realizar eficientemente este proceso. Primeramente se encargará de cargar las imágenes de las células y de contabilizar las células de cada imagen, mediante técnicas de aprendizaje automático y preprocesamiento, además comparará cada imagen con su predecesora, para poder determinar la existencia de malignidades. Los resultados del proceso mostrarán las células segmentadas, el centroide y el área de cada célula en un archivo .CSV generado al final del análisis, y añadirá etiquetas a las células identificadas. Para determinar la eficiencia se mostrará el tiempo de ejecución de la imagen y también por lote de imágenes.

Las metas del proyecto son claras, ya que se basan en alcanzar un beneficio al área biomédica, mejorando y agilizando su trabajo al analizar células.

3.3. Resumen del sistema

3.3.1. Contexto del sistema

El sistema se desarrolla en el contexto de una aplicación de segmentación de células para el laboratorio de quimiosensibilidad tumoral de la Universidad de Costa Rica. Dentro del proyecto se pueden identificar varios subsistemas, que son los siguientes

- Base de datos
- Módulo de segmentación
- Aplicación

La base de datos se encargará de almacenar la información de los usuarios, direcciones de imágenes y resultados. El módulo de segmentación es la parte del programa que se encargará de todos los aspectos relacionados con las imágenes, su procesamiento y su segmentación. Por último, la aplicación se encargará de la comunicación con el cliente y el despliegue de datos. Las relaciones entre estos módulos se puede ver en el mapa de ecosistema de la figura 1

El diagrama de contexto en la figura 2 muestra el universo de la aplicación y las interacciones con el mundo exterior

3.3.2. Funciones del sistema

Las principales funcionalidades del sistema son las siguientes:

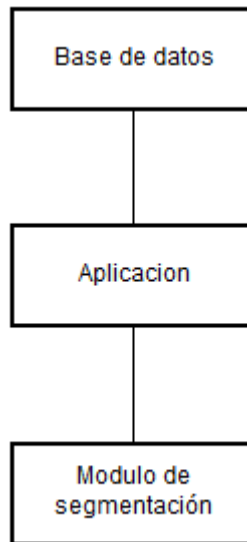


Figura 1: Mapa de ecosistema del proyecto

- Carga de imagenes: Se debe permitir que un usuario seleccione un grupo de imagenes que desea cargar. Estas imagenes serán usadas para realizar la segmentación
- Envio de imagenes a la segmentación: Las imagenes cargadas deben poder ser enviadas al modulo de segmentación para que sean procesadas
- Segmentación de imagenes: Se deben identificar y segmentar las celulas en cada imagen. Esta segmentación debe realizarse rápidamente en relación con el tiempo que duraría un humano en hacer el proceso.
- Etiquetado: Se deben etiquetar las distintas celulas presentes en la imagen
- Identificación de centros y areas: En las celulas identificadas se deben identificar los centroides y el area correspondiente para cada celula
- Presentación de resultados: Se deben agrupar todos los resultados obtenidos en un archivo .CSV para que el usuario pueda observarlos
- Descarga de archivos: Se debe poder descargar el archivo generado con los resultados

3.3.3. Características del usuario

La herramienta solo tendrá una clase de usuario, médicos de la UCR. Esto se debe a que la aplicación está hecha para el laboratorio de quimiosensibilidad tumoral de la UCR. Las características de estos usuarios son:

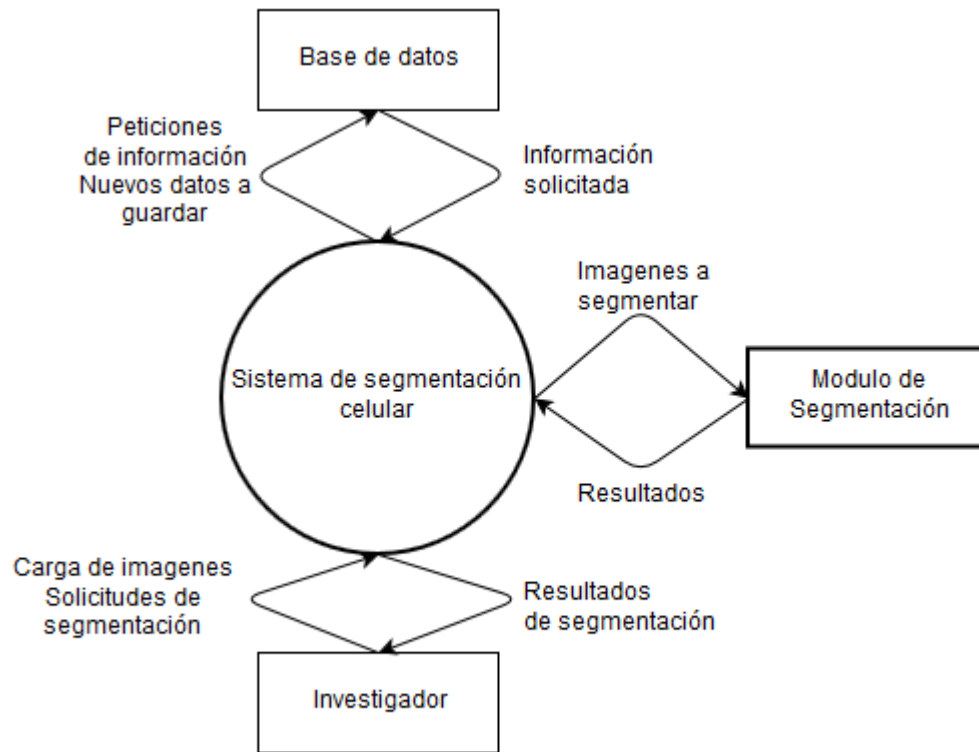


Figura 2: Diagrama de contexto del proyecto

- Espera que el sistema sea exacto y preciso, que mejore el trabajo manual de un humano
- Conocimiento computacional de nivel medio o bajo
- Serán los encargados de usar la mayoría de funcionalidades del sistema
- La mayoría de los requerimientos son destinados a satisfacer sus necesidades

En la figura 4 se puede observar la manera en la que interactuarán con el sistema.

3.4. Requerimientos Funcionales

- REQ-1: Debe poder cargar las imagenes que se desean segmentar, más de una a la vez

- REQ-2: El sistema de cargar los modelos con los cuales se van a procesar las imagenes
- REQ-3: Debe generar un archivo csv con los resultados de la segmentación
- REQ-4: Debe realizar la segmentación de las celulas en las imagenes cargadas

3.5. Requerimientos de Usabilidad

- REQ-1: El sistema deberá ser sencillo de aprender para un usuario con poco conocimiento de sistemas computacionales. No le deberá tomar más de una hora el poder familiarizarse con las principales funcionalidades y poder hacer un uso efectivo de estas.
- REQ-2: El sistema deberá ser fácil de entender, con una interfaz de usuario amigable y sin necesidad de consultar la ayuda del sistema para ser utilizada.
- REQ-3: El programa debe de ser operado sin dificultad alguna, creando un ambiente fluido para poder realizar las funciones sin dificultades innecesarias.

3.6. Requerimientos de Desempeño

Se pueden identificar los siguientes requerimientos de desempeño

- REQ-1: El sistema debe tener una elevada eficiencia temporal, que permita obtener los resultados de la segmentación rápidamente
- REQ-2: El sistema debe ser capaz de soportar grandes volúmenes de imagenes con los que pueda trabajar y manejarlas de manera eficiente, debe tener una elevada eficiencia temporal.

3.7. Interfaces del sistema

El sistema tendrá que comunicarse con dos elementos externos.El primer elemento externo con el que se tendrá que comunicar son los usuarios. Para comunicarse con estos se usará una interfaz gráfica y el protocolo TCP/IP. Esta interfaz tiene los siguientes requerimientos

- REQ-1: La interfaz deberá ser minimalista, clara y ordenada
- REQ-2: Debe permitir acceder a todas las funcionalidades del sistema
- REQ-3: Debe mostrar todos los datos y resultados que sean necesarios para el usuario
- REQ-4: El sistema deberá hacer uso del protocolo HTTP para realizar la transferencia de información

El segundo elemento con el que se deberá conectar la aplicación es la base de datos donde se almacena la información importante para la ejecución del sistema. Los requerimientos asociados a esto son

- Se deberá conectar mediante un conector de base de datos para Python

3.8. Operaciones del sistema

3.8.1. Requisitos de integración del sistema humano

Las partes del sistema que necesitan mayor concentración de recursos humanos es la de la elaboración, entrenamiento y validación de los modelos, ya que estos son la funcionalidad base del sistema y también pueden ocasionar consecuencias catastróficas en los pacientes si no funcionan correctamente y de acuerdo a las necesidades de los investigadores del laboratorio de quimiosensibilidad tumoral.

Otras áreas que pueden necesitar una concentración de recursos humanos significativos son el etiquetado de las imágenes y la presentación de resultados, ya que si no se realizan correctamente y hay errores entonces todas las otras funcionalidades del sistema no serían útiles.

3.8.2. Mantenibilidad

- REQ-1: El sistema debe ser capaz de ser analizado, de poder planificar de una manera sencilla, como quedará el sistema en caso de requerir modificaciones.
- REQ-2: Debe contener un funcionamiento estable, en el cual no hayan particularidades inesperadas con el transcurso del tiempo.
- REQ-3: El programa debe ser fácil de modificar, en donde si se necesita cambiar algo en el código no requiera mucho esfuerzo.

3.8.3. Confiabilidad

Los requerimientos de confiabilidad del sistema son los siguientes

- REQ-1: El sistema debe ser capaz de mantenerse activo sin mostrar fallas y mostrar resultados exactos
- REQ-2: Ya que el sistema es pequeño, debería completar las tareas y obtención de datos de manera eficiente.

Estas necesidades se tienen en un ambiente de uso normal, en el que se espera que no se tengan más de 5 usuarios en un momento dado, donde cada uno está haciendo tareas normales para la aplicación, como la carga de datos, la obtención de resultados o el envío de consultas.

3.9. Características físicas

3.9.1. Requerimientos físicos

Los requerimientos físicos del sistema son los siguientes

- REQ-1: El sistema deberá desplegarse en una plataforma de computación en la nube
- REQ-2: Se deberá tener suficiente poder computacional en la plataforma de la nube para poder procesar todas las necesidades de los usuarios
- REQ-3: Se deberá tener suficiente espacio de almacenamiento en la nube para poder almacenar las imágenes cargadas

3.9.2. Requerimientos de adaptabilidad

Se pueden identificar los siguientes requerimientos de adaptabilidad

- REQ-1: Se debe mantener el presupuesto suficiente por si el sistema necesita actualizaciones o nuevas funciones que requiera el usuario.
- REQ-2: Será desplegado en una plataforma de computación en la nube que se encargará de las instalaciones de crecimiento del sistema.

3.10. Condiciones ambientales

Debido a la naturaleza del sistema las condiciones ambientales no tienen ningún efecto en el sistema.

3.11. Seguridad del sistema

Se pueden identificar los siguientes requerimientos de seguridad para el sistema

- REQ-1: El sistema debe ser seguro permitiendo que solo usuarios registrados puedan ingresar o utilizar las funciones de este.
- REQ-2: No permite que se viole la privacidad de los pacientes ya sea por el ingreso de una persona externa al sistema y que pueda robarse la información.

3.12. Manejo de Información

Los requerimientos del manejo de información para el sistema son los siguientes

- REQ-1: Soportar grandes cantidades de imágenes ya que es su principal manejo de información.

- REQ-2: Debe generar un archivo csv en el cual se dará el resultado del procesamiento de imagenes.

El sistema generalmente trabaja con imágenes y ese es su principal manejo de información, aparte del resultado del analisis el cual es un archivo csv con los datos del resultado final de cada prueba.

3.13. Políticas y regulaciones

Este sistema debe estar siempre apegado a las legislaciones pertinentes que se rigen en Costa Rica y en las universidades publicas en el área de sistemas computacionales, aparte se debe manejar un concepto de confidencialidad máxima para no exponer información de ningun paciente.

3.14. Sostenibilidad del ciclo de vida del sistema

Se debe estar realizando un conteo y un analisis de los resultados que registra el software en cada prueba, y con esto saber si se deben realizar cambios o mejoras en algun momento de vida del sistema. Ya que la aplicación se encontrará en la nube, no hace falta evaluar una situación de posteriores instalaciones.

3.15. Empaquetado, manejo, envío y transporte

- REQ-1: El sistema debe permitir la carga de imágenes ya que esto es primordial para completar la función principal de este sistema.
- REQ-2: Se debe crear, enviar y mostrar el archivo csv correctamente al usuario.

3.16. Verificación

La manera de comprobar si el sistema está dando los resultados correctos, es realizando comparaciones de los resultados finales de cada prueba haciendo las mismas pruebas en otro sistema similar y que esté comprobado que este funcione. Es recomendable hacer esta prueba con varios sistemas y no solo otro más.

3.17. Suposiciones y dependencias

Se tienen las siguientes suposiciones y dependencias para el correcto funcionamiento del sistema

1. Se deberá tener acceso a Internet.
2. Se deberá tener un navegador web como Mozilla Firefox u Opera

4. Diagramas del proyecto

4.1. Diagramas UML

En esta sección se adjuntarán los diagramas de componentes y UML de la primera iteración del proyecto, con el fin de mejorar la comprensión y planificación del proyecto.

Para mejorar el diseño del proyecto se decidió utilizar un patrón de diseño creacional, el Singleton [4]. Con esto se asegura de que solo se tendrá una instanciación del controlador y todos tendrán acceso a el.

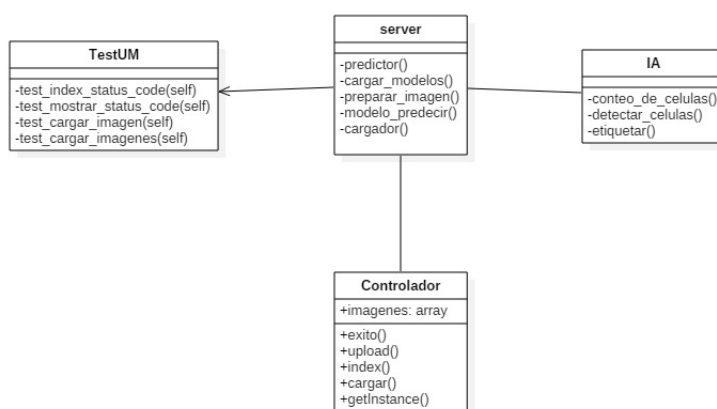


Figura 3: Diagrama de clases del sistema

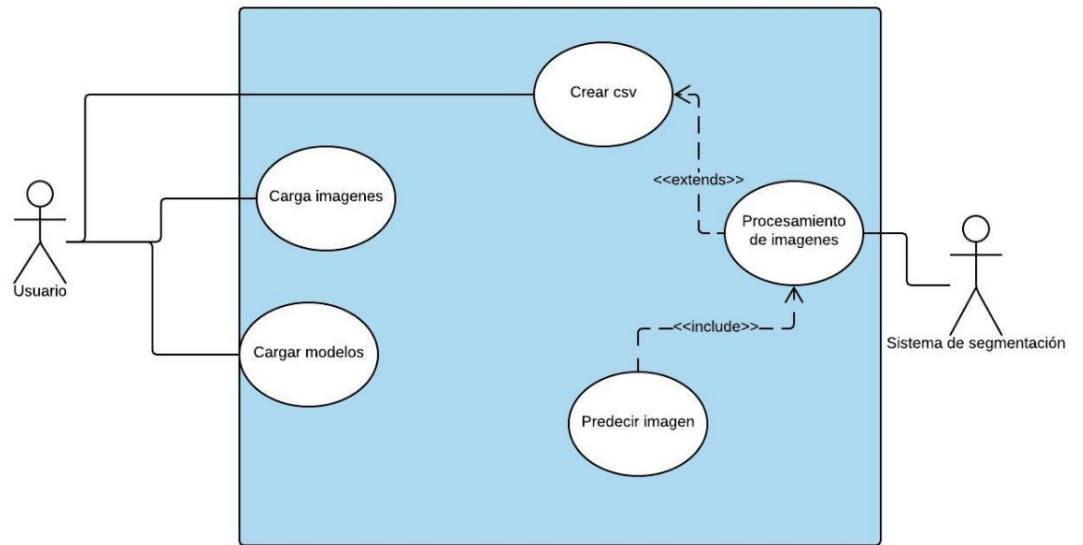


Figura 4: Diagrama de casos de uso del sistema

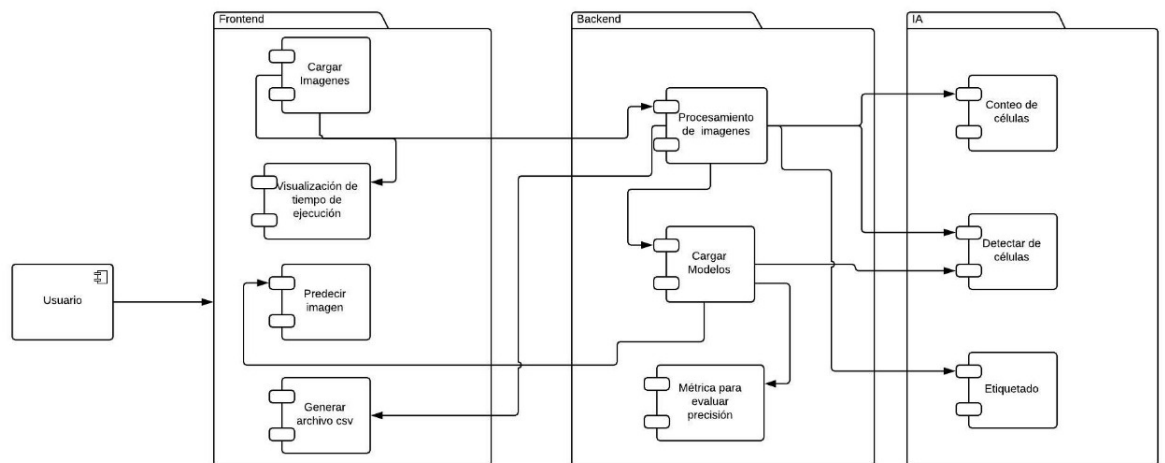


Figura 5: Diagrama de componentes del sistema

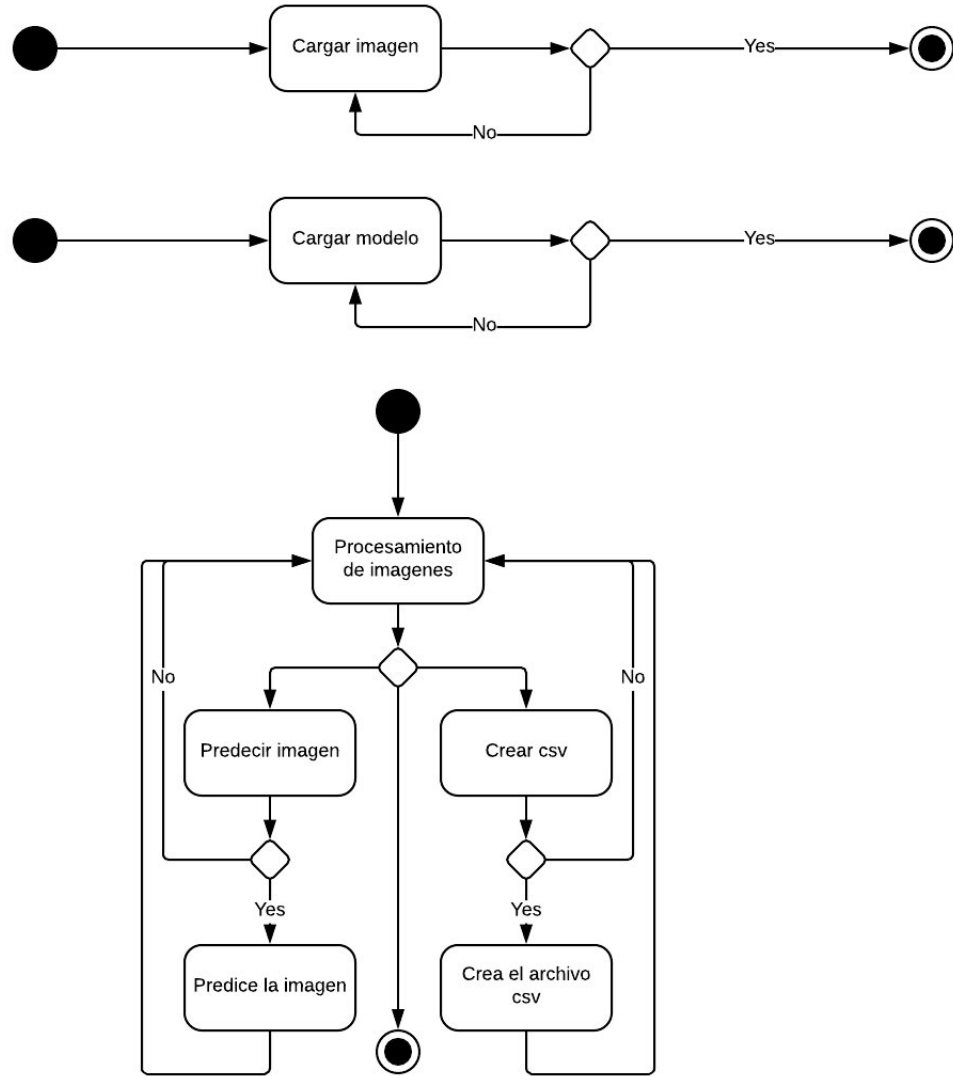


Figura 6: Diagramas de actividad del sistema

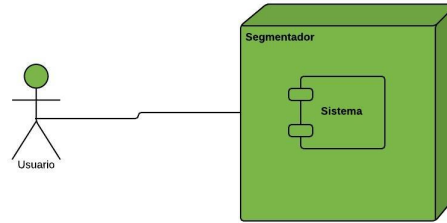


Figura 7: Diagrama de despliegue del sistema

5. Estándar de documentación

Para esta sección se compararán los estándares de

- MediaWiki [2]
- Google [1]
- PEP [3]

Se usará el acrónimo NM para representar características no mencionadas por una guía de estilo

Característica y Estilo	MediaWiki	Google	PEP
Indentación	4 espacios	4 espacios	4 espacios
Variables Globales	Evitarlas	Evitarlas	Evitarlas, solo para módulos
Comentarios de bloque	Se ponen antes de partes complicadas del código, no se describe el código. Se separan con 2 líneas en blanco de la parte que explican	NM	Inician con # seguido de un espacio, indentados al mismo nivel del código
Comentarios de línea	Se ponen después de líneas cuyo funcionamiento no es obvio	NM	Usarlos escasamente
Largo máximo de una línea	80 caracteres	Sugiere menos de 80 caracteres, siempre tomando en cuenta la legibilidad	79 caracteres para código, 72 para comentarios
Líneas en blanco	Dos líneas en blanco entre definiciones de alto nivel, una línea en blanco entre la definición de métodos	NM	Padding de 2 líneas para clases y funciones de alto nivel, de 1 línea para métodos en clase

Nombres de funciones	Misma que en [3]	Cortos y descriptivos	En minúscula con _ separando las palabras
Nombres de clases	Misma que en [3]	Cortos y descriptivos	En CamelCase
Nombres de variables	Misma que en [3]	En minúscula con _ separando las palabras	
Constantes	En mayúscula y si son varias palabras con barras abajo	NM	Nombres en mayúscula
Espaciados alrededor de comas	Después y no antes	Un espacio después mas no antes	Después mas no antes
Espaciado alrededor de operadores	Antes y después	Un espacio antes y después	Añadir un espacio antes y después de los operadores binarios, considerar usar espacios para denotar las prioridades de los operadores
Espaciado alrededor de paréntesis	No antes y tampoco después	Un espacio después mas no antes	Antes no, después tampoco, solo en casos específicos se pueden agregar espacios antes
Argumentos de funciones	Misma convención que para los nombres de variables, se permiten los valores por defecto en la mayoría de casos	Misma convención que para los nombres de variables	Misma convención que para los nombres de variables
Strings	4 espacios	Favorece "	No hace recomendaciones, ' y " son equivalentes
Imports	Solo usarlos para paquetes y módulos	Misma convención que la de [3]	En distintas líneas, a menos de que se use from * import. Siempre en la parte superior del documento. Separados en 3 grupos: librerías estándar, imports de terceros, imports locales. Cada grupo debe estar separado por una línea en blanco

Al poder comparar todas estas características se ha decidido usar el estándar de PEP, ya que este provee una mayor definición que el estándar de Google y se asemeja más al estilo de los miembros del grupo. Otra razón por la que se escogió el estándar de PEP es que el oficial y propuesto por el creador de Python, por lo que se espera que sea el más usado y por lo tanto el estilo del código se

asemeje lo más posible al de otros proyectos. El estándar de MediaWiki no será usado debido a sus obvias limitaciones.

6. Actividades de aseguramiento de la calidad

Sprint	Actividad	Encargado	Descripción
Sprint del avance 1	Desarrollo de pruebas unitarias para la función de carga de imágenes	Juan Villacis	Se planean y codifican las pruebas unitarias para los trozos de código implementados en este sprint
	Revisión de cumplimiento de estándares de codificación	Juan Villacis	Se revisa el código para determinar si cumple con el estándar de codificación seleccionado
	Establecimiento del proceso de aseguramiento de la calidad del software	André Arroyo y Bryan Vargas	Se desarrollan las etapas de las que se compondrá el proceso de aseguramiento de la calidad
	Evaluar conformidad de diagramas con requerimientos	Bryan Vargas	Se comparan los diagramas generados con los requerimientos para determinar
Sprint del avance 2	Desarrollo de pruebas unitarias para el código desarrollado en el sprint	Juan Villacis	Se planean y codifican las pruebas unitarias para los trozos de código implementados en este sprint
	Revisión de cumplimiento de estándares de codificación	Juan Villacis	Se revisa el código para determinar si cumple con el estándar de codificación seleccionado
	Revisión de conformidad del diseño con los requerimientos	André Arroyo y Bryan Vargas	Se analiza el diseño elaborado para determinar el grado de cumplimiento de los aspectos pertinentes con los requerimientos
	Evaluar conformidad del código desarrollado con los requerimientos	Juan Villacis	Se compara el código con los requerimientos para determinar si los aspectos pertinentes cumplen con los requerimientos establecidos

	Evaluación del conocimiento de los desarrolladores	André Arroyo	Se mide el grado de conocimiento de los desarrolladores en las tecnologías que se usarán para determinar su nivel de experiencia y recomendar temas por mejorar
	Desarrollo de pruebas de integración de la herramienta	Bryan Vargas	Se desarrollan las pruebas para medir el grado de integración de los distintos componentes del sistema
	Desarrollo de pruebas de aceptación de la herramienta	André Arroyo	Se desarrollan las pruebas para medir la aceptación de la herramienta por parte del cliente
Sprint del avance 3	Desarrollo de pruebas unitarias para el código desarrollado en el sprint	Juan Villacis	Se planean y codifican las pruebas unitarias para los trozos de código implementados en este sprint
	Revisión de cumplimiento de estándares de codificación	Juan Villacis	Se revisa el código para determinar si cumple con el estándar de codificación seleccionado
	Evaluar conformidad de la implementación con los requerimientos	Juan Villacis	Se analiza la implementación contra los requerimientos para determinar en qué grado los satisface
	Evaluar conformidad de la implementación con el diseño	Juan Villacis	Se evalúa la implementación para determinar su grado de apego al diseño
	Evaluar aceptabilidad de la implementación	André Arroyo	Se evalúa la aceptabilidad de la implementación de acuerdo al plan desarrollado
	Evaluar integración de la implementación	Bryan Vargas	Se evalúa el grado de integración de acuerdo al plan desarrollado
	Evaluar el cumplimiento del plan de calidad	André Arroyo	Se analizan las actividades de aseguramiento del plan de calidad para determinar el grado de cumplimiento que se tuvo

	Realización de mediciones relacionadas a la precisión y exhaustividad de la herramienta	Bryan Vargas	Se prueba el sistema con varios casos de prueba para determinar la precisión y exhaustividad de esta, usando las formulas para tal fin
	Realización de mediciones relacionadas a el desempeño de la herramienta	André Arroyo	Se mide el tiempo de carga de imagenes, de respuesta, de procesamiento entre otros para determinar estas características del sistema y compararlas con las de los requerimientos
	Evaluar aceptación del producto	Bryan Vargas	Se evalua el grado de aceptación de acuerdo al plan desarrollado
	Evaluar el cumplimiento del plan de calidad	André Arroyo	Se analizan las actividades de aseguramiento del plan de calidad para determinar el grado de cumplimiento que se tuvo

7. Carga de imagenes al servidor

Las pruebas unitarias implementadas para esto fueron las siguientes

1. Obtener un código 200 al buscar la página inicial: Esta prueba tiene como objetivo determinar si la página inicial del sistema está en línea, esto se hace para determinar si el sistema logró levantarse correctamente y el servidor está respondiendo solicitudes
2. Obtener un código 200 al enviar una imagen al manejador de peticiones POST de carga de imágenes: Esta prueba tiene como objetivo determinar si la funcionalidad básica de carga de una sola imagen está funcional y responde correctamente cuando se le envía una imagen
3. Obtener un código 200 al enviar varias imágenes: Esta prueba tiene como objetivo determinar si la primera funcionalidad requerida para esta etapa (la de cargar varias imágenes) se encuentra funcional y responde correctamente cuando se le envía más de una imagen
4. Obtener un código de respuesta 200 al hacer una petición a la página que muestra las imágenes cargadas: Esta prueba tiene como objetivo determinar si la segunda funcionalidad requerida para esta etapa (mostrar las imágenes cargadas se encuentra funcional y despliega la información necesaria).

8. Conclusión

En este trabajo se ha podido observar el planeamiento alrededor de varios puntos relacionados con el aseguramiento de la calidad del sistema. Se definió un estilo de codificación y un calendario de actividades de aseguramiento de la calidad, que harán que el proceso de garantizar la calidad del sistema sea más ordenado y completo. Adicionalmente la realización del diseño del sistema y la definición de los requerimientos le proveerán a los desarrolladores una guía a la que apegarse durante el desarrollo del producto

Referencias

- [1] Google. Google python style guide. <https://github.com/google/styleguide/blob/gh-pages/pyguide.md>, 2018.
- [2] MediaWiki. Coding conventions/python. https://www.mediawiki.org/wiki/Manual:Coding_conventions/Python, 2018.
- [3] Guido von Rossum and Barry Warsaw. Style guide for python code. <https://www.python.org/dev/peps/pep-0008/#code-lay-out>, 2013.
- [4] sourcemaking. Singleton design pattern. https://sourcemaking.com/design_patterns/singleton.