

Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería en Computación  
Curso de Aseguramiento de la Calidad del Software  
Avance No. 1.2 de Proyecto:  
Documento de Extensión del SyRS

Francisco Monge Zúñiga (2013029434)  
Andrés Gutiérrez Salas (201223823)  
Joseph Salazar Acuña (2015100516)

September 19, 2018

## 1 Introducción

Los dispositivos de captura y procesamiento de imágenes biomédicas, como resonancias magnéticas, radiografías y tomografías computarizadas, cobran cada vez mayor protagonismo en los procesos de toma de decisiones para los profesionales de la salud. Más recientemente, la industria biomédica ha desarrollado interesantes sistemas de software que realizan diagnósticos automáticos a partir de imágenes y señales biomédicas.

En nuestro caso específico, se abordará un problema de gran interés: la segmentación de imágenes digitales tomadas de un tejido de glioblastoma, para la detección de células cancerígenas. Proceso que, de realizarse de manera manual, toma demasiado tiempo por parte de los microbiólogos encargados.

En el presente Documento de Especificación de Requerimientos (SyRS) encontrará una serie de funcionalidades, atributos y condiciones que el sistema de software a desarrollar debe cumplir, de cara a resolver satisfactoriamente las necesidades de quienes lo van a utilizar.

## 2 Estándares de Codificación

El estándar de codificación que se implementará en el lenguaje Python es conocido como Python Enhancement Proposals (PEP). Su última versión es la número ocho (PEP8), siendo esta es la que se utilizará como referencia. Se implementará dicho estándar de codificación dado que es el oficial y el que soporta Python y toda su comunidad.

Existen distintos métodos para realizar el chequeo del estándar. El principal se llama PyDEV, que es un complemento nativo para Eclipse. PyDEV incluye un chequeador del código de Python, pero es bastante permisivo y hasta cierto punto "light", ya que solo muestra en la barra lateral (con la numeración de la línea de código) una marca de color que indica que alguna regla definida en PEP8 no se está cumpliendo. No obstante no indica el tipo de error ni recomendaciones de cómo corregirlo.

Luego, existen otras herramientas que se pueden instalar como complementos que brindan funcionalidades similares como lo son PyLint y Pycodestyle. PyLint permite conocer en mayor detalle cuáles son los incumplimientos que se están cometiendo y permite formatear automáticamente el código a una forma más aceptada por PEP8. Por su parte Pycodestyle es una código externo, que permite generar un informe de errores por el incumplimiento de las normas PEP, con la indicación de las líneas específicas donde corregir.

Fuera del entorno de desarrollo Eclipse existen páginas web donde se puede realizar revisiones de dicho estándar, solo pegando el código y automáticamente se parsea a una forma estable.

Existen otros entornos de desarrollo (IDE) que son más exhaustivos y estrictos en dicha verificación del PEP8 y brindan otras facilidades, como lo es PyCharm.

### 3 Diagrama de Componentes

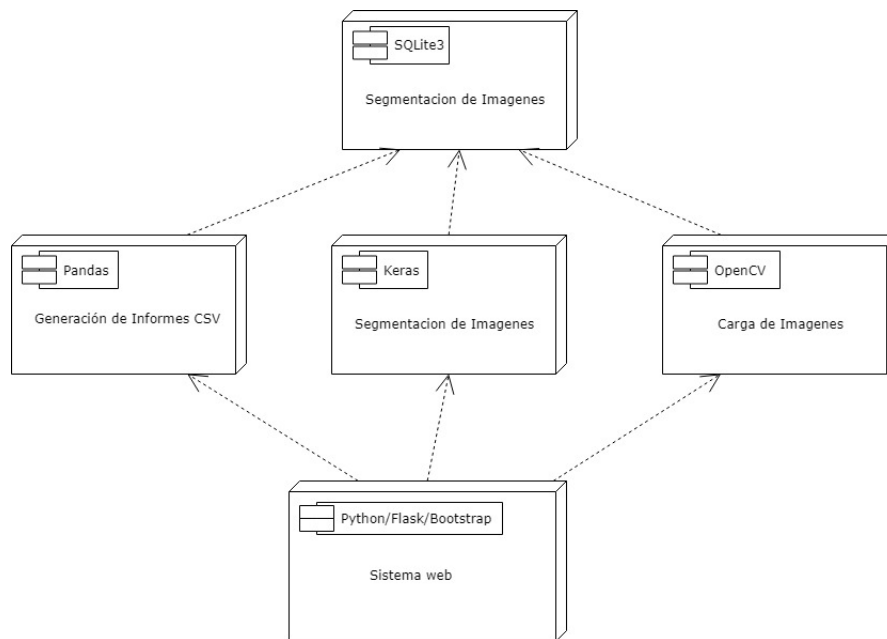


Figure 1: Diagrama de componentes de alto nivel del sistema implementado.

### 4 Diagrama de Clases

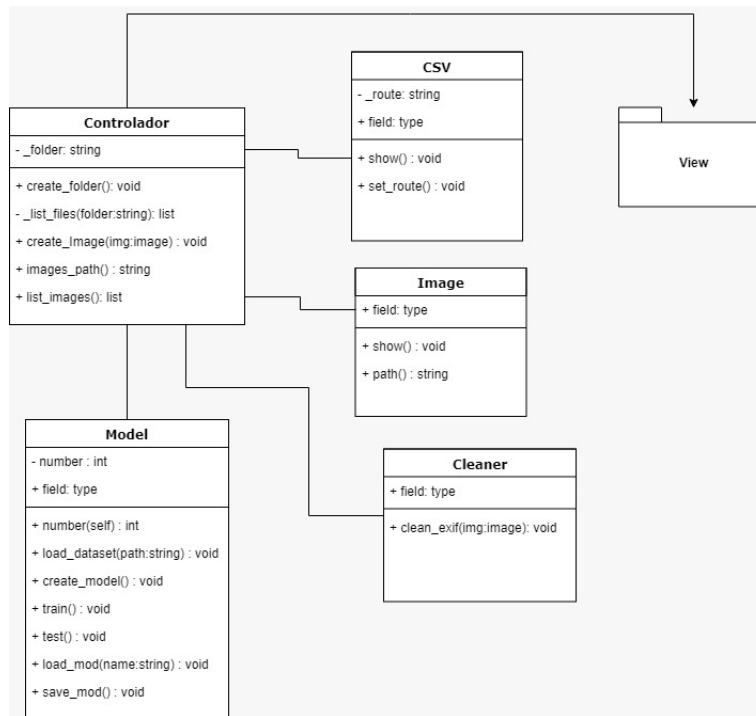


Figure 2: Diagrama de clases del sistema implementado.

Bibliografía consultada: Larman, C. (s.f.) "Appliyng UML and Patterns - An introduction to Object-Oriented Analysis and Design and the Unified Process".

## 5 Actividades de Aseguramiento de la Calidad

1) Evaluación de conformidad del diseño respecto a los requerimientos.

a) Actividad:

Realizar reuniones presenciales con la contraparte donde se verifique que el diseño preliminar propuesto concuerde con los requerimientos establecidos en el Documento de Especificación de Requerimientos (SyRS).

b) Objetivo:

Verificar si el diseño de la aplicación cumple con las expectativas y puntos establecidos, avances, del sistema.

c) Cronograma:

Una reunión durante cada sprint.

2) Evaluación de conformidad de la implementación respecto al diseño.

a) Actividad:

Realizar revisiones internas con el equipo de desarrollo donde se verifique que el diseño establecido concuerde con el código desarrollado para las funcionalidades específicas. El encargado de realizar dicha revisión no puede ser el miembro que desarrolló.

b) Objetivo:

Verificar que el código desarrollado por los miembros del equipo cumple con el diseño establecido y, por ende, con los requerimientos establecidos.

c) Cronograma:

Se debe realizar un pull request cada vez que se implementa una tarea.

3) Evaluación de conformidad de la implementación respecto a los requerimientos.

a) Actividad: Realizar reuniones presenciales con la contraparte donde se verifique que la implementación de las funcionalidades concuerde con los requerimientos establecidos en el Documento de Especificación de Requerimientos (SyRS).

b) Objetivo:

Verificar que la implementación de las funcionalidades cumple con las expectativas del cliente.

c) Cronograma:

Una reunión durante cada sprint.

4) Cumplimiento de estándares de codificación.

a) Actividad:

Realizar chequeos cada vez que se hacen pull requests al Github con las herramientas previamente establecidas en el apartado de estándares de codificación.

b) Objetivo:

Cumplir con los estándares para así asegurar la calidad del software.

c) Cronograma:

Cada vez que se realiza un pull request en Github.

5) Construcción y revisión de tests unitarios y de integración.

a) Actividad:

Realizar una serie de tests unitarios para verificar que lo implementado cumple con lo establecido, de una manera correcta y así disminuir la aparición de errores.

b) Objetivo:

Asegurar que las funcionalidades implementadas realizan su función específica de una forma correcta y así disminuir la aparición de errores.

c) Cronograma:

Se construyen previo a iniciar al desarrollo de una funcionalidad. Se prueban una vez finalizado el desarrollo de la funcionalidad.