

Aseguramiento de la Calidad del Software

Avance #2

André Arroyo Piedra
2015073657

Bryan Vargas
2015011562

Juan Villacis
2016201681

8 de octubre del 2018

1. Introducción

En este documento se procede a detallar algunos aspectos relacionados a la calidad en el proyecto: Herramienta para la segmentación de células. Algunos de los aspectos que aquí se mencionan se relacionan como el Syrs, como la actualización de este, el mapeo entre elementos de diseño e implementación y requerimientos. Al igual, se cuantizan métricas relacionadas con los requerimientos establecidos para el sistema y se muestra como usar herramientas desarrolladas para tal fin. Por último también se detallan las pruebas unitarias realizadas para garantizar el funcionamiento de la herramienta.

2. Repositorio

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

3. Documento de requerimientos actualizado

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
- Modulo de segmentación
- Aplicación

La base de datos se encargará de almacenar la información de los usuarios, direcciones de imágenes y resultados. El modulo 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

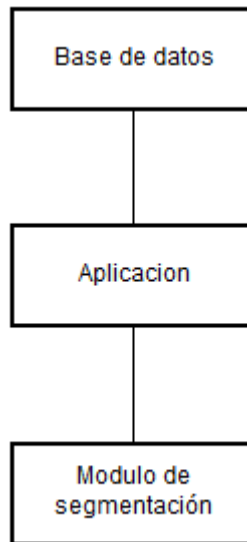


Figura 1: Mapa de ecosistema del proyecto

El diagrama de contexto en la figura 4 muestra el universo de la aplicación y las interacciones con el mundo exterior

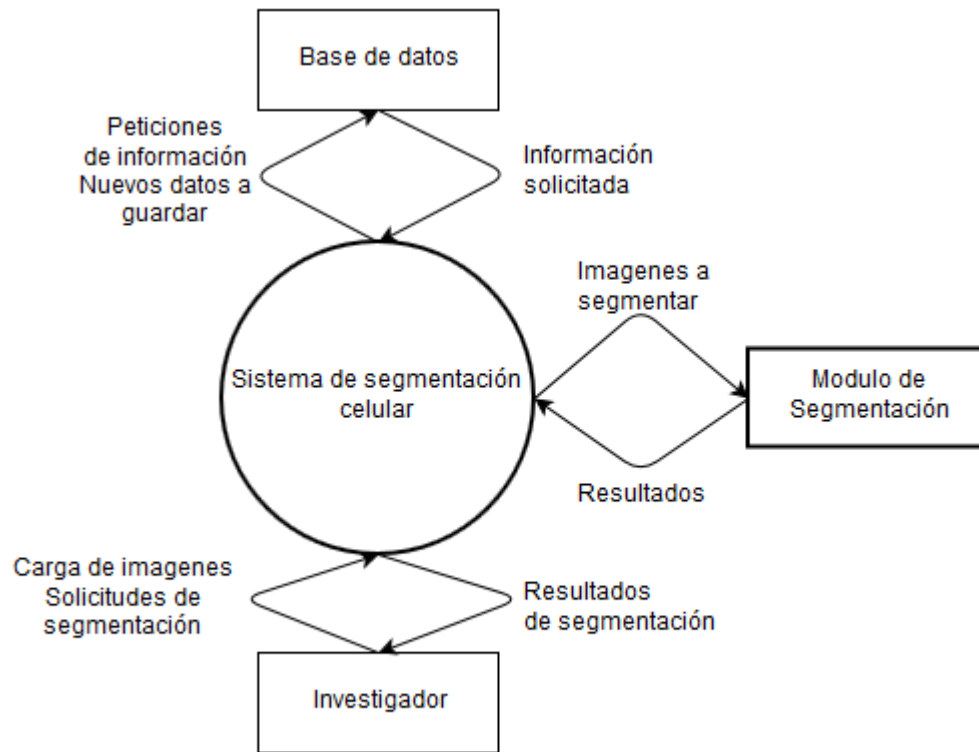


Figura 2: Diagrama de contexto del proyecto

3.3.2. Funciones del sistema

Las principales funcionalidades del sistemas son las siguientes:

- Carga de imágenes: Se debe permitir que un usuario seleccione un grupo de imágenes que desea cargar. Estas imágenes serán usadas para realizar la segmentación
- Envío de imágenes a la segmentación: Las imágenes cargadas deben poder ser enviadas al modulo de segmentación para que sean procesadas
- Segmentación de imágenes: 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 células presentes en la imagen
- Identificación de centros y áreas: En las células identificadas se deben identificar los centroides y el área correspondiente para cada célula

- 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:

- 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 3 se puede observar la manera en la que interactuarán con el sistema.

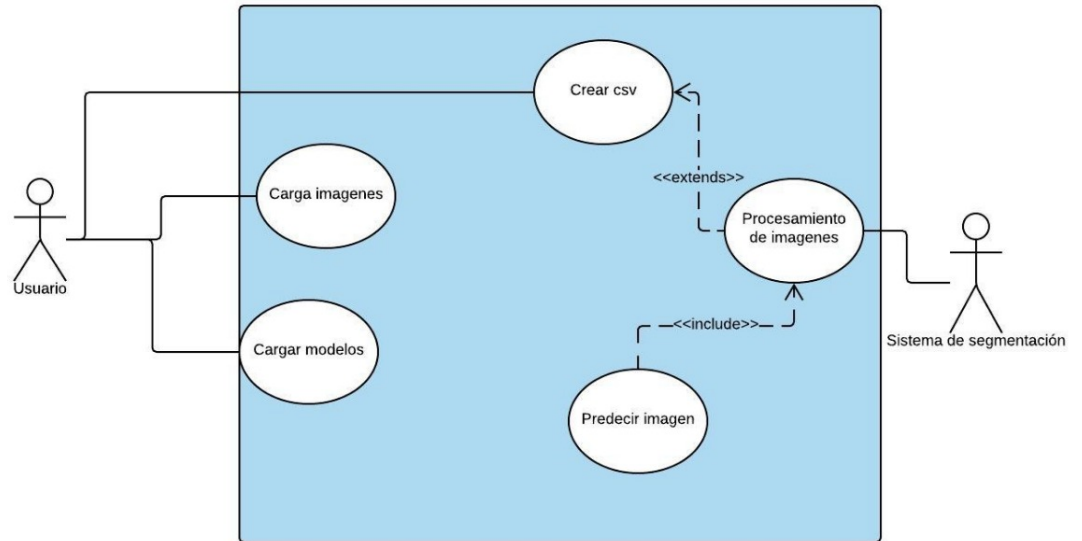


Figura 3: Diagrama de casos de uso del sistema

3.4. Requerimientos Funcionales

- REQ-1: Debe poder cargar las imágenes que se desean segmentar, más de una a la vez
- REQ-2: El sistema debe cargar los modelos con los cuales se van a procesar las imágenes
- REQ-3: Debe generar un archivo csv con los resultados de la segmentación
- REQ-4: Debe realizar la segmentación de las células en las imágenes cargadas, tanto en las regiones de fondo como en las de no fondo
- REQ-4: Debe permitir acceder a las funcionalidades desde una interfaz web.

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.
- REQ-4: El programa debe permitir realizar la segmentación sin tener que presionar más de 5 botones

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 imágenes con los que pueda trabajar y manejarlas de manera eficiente, debe tener una elevada eficiencia temporal.
- REQ-3: El sistema debe ser capaz de soportar al menos 10 usuarios de manera concurrente y responder con no mas de 3 segundos de demora a las peticiones web

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.
- REQ-3: Se debe garantizar una precisión constante entre los resultados obtenidos

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 imágenes.

El sistema generalmente trabaja con imágenes y ese es su principal manejo de información, aparte del resultado del análisis 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 ningún paciente.

3.14. Sostenibilidad del ciclo de vida del sistema

Se debe estar realizando un conteo y un análisis de los resultados que registra el software en cada prueba, y con esto saber si se deben realizar cambios o mejoras en algún 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. Descripción de la actividad de aseguramiento de la calidad

4.1. Pruebas de aceptación de la herramienta

Estas pruebas buscan validar el comportamiento de la aplicación para efectos del cliente. La idea es comprobar si la aplicación realiza correctamente las tareas para la cual fue creada.

Para efectos de estas pruebas se realiza basadas en los casos de uso del usuario.

1. Comprobar si la aplicación carga imágenes correctamente
 - Prueba: Se pone a correr la aplicación y se intenta cargar varias imágenes cerrando y abriendo la aplicación varias veces para comprobar la cantidad de fallos.
 - Resultado: En todos los intentos realizados, la imagen se carga correctamente al programa.
2. Cargar varios modelos al programa
 - Prueba: Se inicia la aplicación, se carga un modelo, se verifica si carga correctamente y se cierra la aplicación. Luego se repite este experimento varias veces para ver en cuantas ocasiones no carga un modelo correctamente.
 - Resultado: En todas las pruebas que se realizaron, se carga correctamente el modelo al sistema.
3. Verificar si luego de hacer pruebas. el programa crea y entrega el CSV correctamente

- Prueba: Se inicia el programa y se realiza todos los pasos necesarios para hacer la segmentación de imágenes, esto se realiza varias veces y en todas se pide el correspondiente CSV con los resultados para verificar si siempre lo crea correctamente.
- Resultado: Luego de realizar la prueba varias veces, se obtiene que el programa al terminar el proceso siempre crea el CSV para uso del usuario.

4.2. Evaluación del conocimiento de los desarrolladores

Esta prueba consiste en hacer varias preguntas sobre herramientas usadas en el sistema a los desarrolladores para medir que tanto conocimiento tienen sobre el funcionamiento interno de estas herramientas y así dar recomendaciones a estos, sobre temas a mejorar.

1. ¿Para crear el archivo CSV cual librería se usa?
2. ¿Para que sirve la herramienta web Flask?
3. ¿Cual herramienta es usada para detectar y descifrar patrones y correlaciones?
4. ¿Para que se usa la librería Numpy en el sistema?
5. ¿Cual es la librería usada para la generación de gráficos?
6. ¿Para que se usa AngularJS en este sistema?
7. ¿Cual librería es usada para mejorar la apariencia de la interfaz?
8. ¿Que es Keras?
9. ¿Para que se utiliza PyUnit?
10. ¿Para que sirve la red VGG_16?

5. Validación del diseño

Se puede observar en el cuadro 1 las relaciones existentes entre ítem de diseño y los requerimientos del sistema

Ítem de diseño	Requerimiento del Syrs asociado
Caso de uso: carga de imágenes	Funcionales-REQ-1, Empaquetado-REQ-1
Caso de uso: cargar modelos	Funcionales-REQ-4
Caso de uso: crear csv	Funcionales-REQ-3, Información-REQ-2, Empaquetado-REQ-2
Caso de uso: segmentar imagenes	Funcionales-REQ-4
Despliegue del sistema	Funcionales-REQ-[1,2,3,4]
Proceso: carga de imagenes	Funcionales-REQ-1, Empaquetado-REQ-1
Proceso: cargar modelo	Funcionales-REQ-4
Proceso: procesar imágenes	Funcionales-REQ-1, Empaquetado-REQ-1
Componente: frontend	Usabilidad-REQ-[1,2,3], Interfaces-REQ-[1,2,3], Empaquetado-REQ-1
Componente: backend	Funcionales-REQ-1, Interfaces-REQ-4, Adaptabilidad-REQ-2
Componente: IA	Funcionales-REQ-4
Clase: Server	Funcionales-REQ-1
Clase: Unet	Funcionales-REQ-4, Desempeño-REQ-1
Clase: Modelo	Seguridad-REQ-[1,2], Informacion-REQ-1, Desempeño-REQ-2
Clase: Prueba	Mantenibilidad-REQ-1, Confiabilidad-REQ-1

Cuadro 1: Relaciones entre ítem de diseño y los requerimientos definidos en el Syrs

6. Validación de la implementación

Se puede observar en el cuadro 1 las relaciones existentes entre ítem de diseño y los elementos de la implementación

Ítem de diseño	Ítem de implementación asociado
Caso de uso: carga de imágenes	server.py, cargar.html, visualizar.html
Caso de uso: cargar modelos	unet.py, segmentar.html
Caso de uso: crear csv	TBD
Caso de uso: segmentar imágenes	unet.py
Despliegue del sistema	server.py
Proceso: carga de imagenes	server.py, cargar.html
Proceso: cargar modelo	unet.py, segmentar.html
Proceso: procesar imagenes	cargar.html, visualizar.html, server.py
Componente: frontend	cargar.html, index.html, visualizar.html, segmentar.html, login.html, results.html, segmentadas.html
Componente: backend	server.py
Componente: IA	unet.py
Clase: Server	server.py
Clase: Unet	unet.py
Clase: Modelo	modelos.py
Clase: Prueba	pruebas.py

Cuadro 2: Relaciones entre ítem de diseño y la implementación

7. Verificación del estándar de codificación

Para este proyecto se utilizará la herramienta incorporada en PyDev llamada pep8, esta ayudará a verificar que el estándar de codificación del proyecto se este cumpliendo correctamente.

7.1. Modo de uso de la herramienta

Para utilizar esta herramienta se debe instalar la librería externa PyDev.

Ya teniendo la librería antes dicha instalada se debe abrir el proyecto en eclipse, acá en el botón Window → Preference. Acá se abrirá un menú en donde se muestran varias opciones del proyecto, se dirige a Editor → Code Analysis → PyLint, se presentarán distintas opciones de manejo de errores, warnings y demás.

En la opción pep8 se debe marcar la opción Warning en vez de Don't run. Para correr esto correctamente después de implementar la configuración se debe dar click derecho en el proyecto dentro de eclipse y dirigirse a la opción PyDeV → Code analysis.

Para finalizar se debe agregar al path el proyecto, se logra haciendo click derecho sobre el proyecto, luego elegir la opción PyDev → Set as source-folder. De esta manera indicará los errores cometidos según el estándar de codificación.

7.2. Pruebas de la herramienta

Como prueba de lo que se logra ver usando la herramienta se adjunta la siguiente imagen:

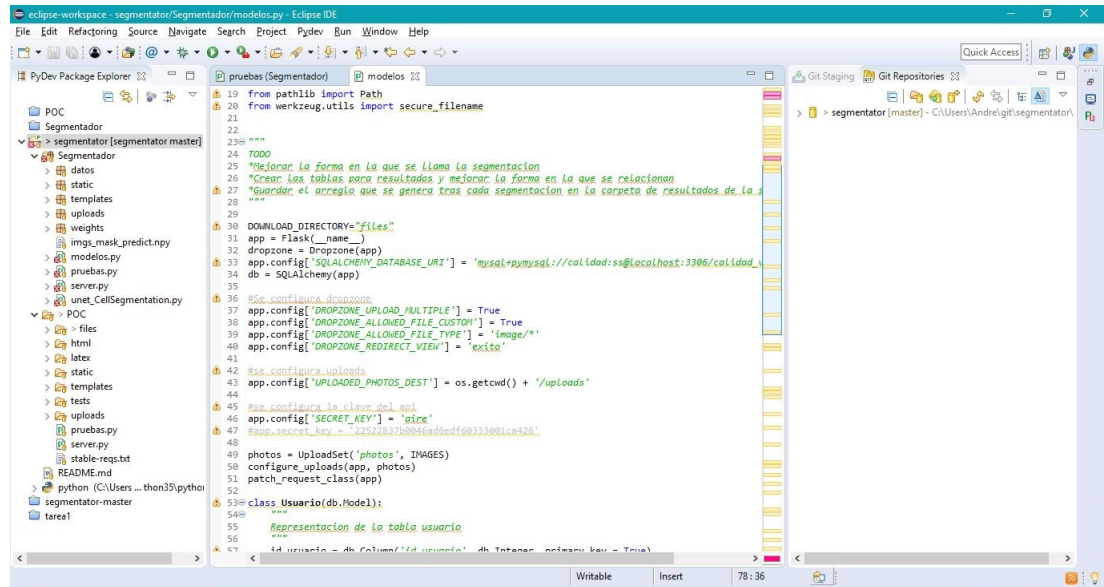


Figura 4: Prueba de herramienta

8. Versión actual del proyecto

Para determinar la versión actual del proyecto se usará el versionamiento semántico. El primer número se determina que es un 0, esto se debe a que aún se está en la etapa de desarrollo y no ha habido ningún cambio que provoque incompatibilidad en el sistema. Con respecto al segundo número se establece que este es un 3, debido a que esta es la tercera versión con cambios importantes pero no exclusivos que se realiza. La primera fue el POC, la segunda fue el avance #1 y la actual es el avance #3. Con respecto a el tercer número se establece que este es un 2, debido a que se han hecho dos cambios menores. Esto se puede observar en las figuras 6, 5 y 7. En estas figuras se puede observar claramente las secuencias de cambios que pertenecen a cada versión menor del sistema, así como los cambios que pertenecen a cada sprint. El proceso general para poder determinar esta versión consiste en evaluar primero los cambios excluyentes, después considerar los cambios menores y por último considerar los cambios menores que se han realizado desde el último cambio mayor. En resumen, se considera que el sistema se encuentra en la versión 0.3.2 ya que se ha venido trabajando en la misma base en todos los entregables, donde solo han habido cambios pequeños.

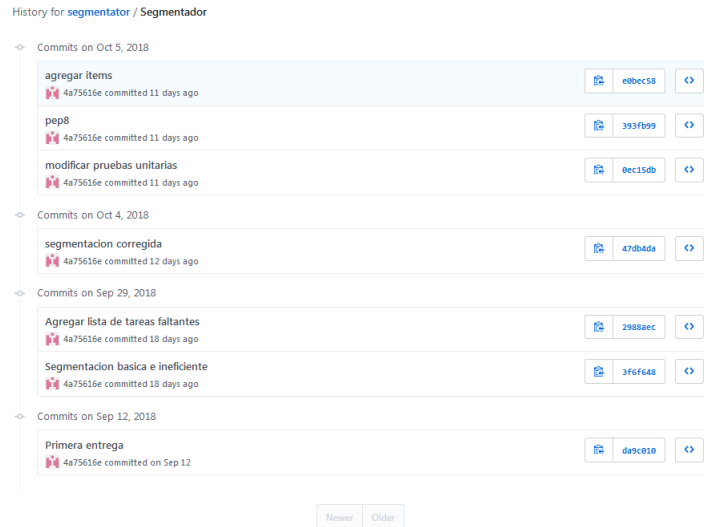


Figura 5: Diferencia entre versiones del proyecto

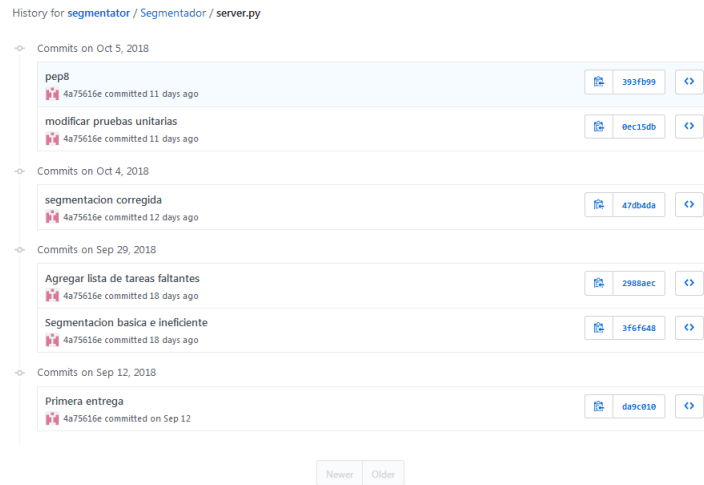


Figura 6: Diferencia entre versiones del código del servidor

Como se muestra en la imagen se puede apreciar que se le han agregado diferentes componentes, un ejemplo de esto es la base de datos. Sin embargo aún con todos estos cambios las versiones siguen siendo compatibles.

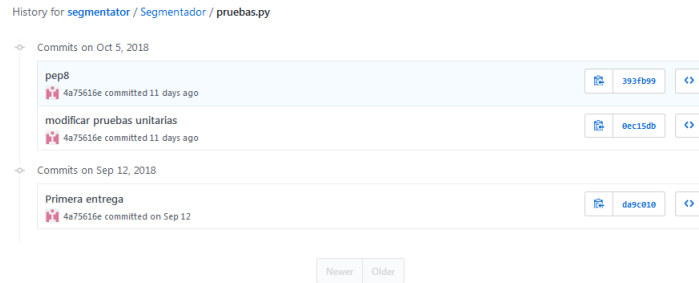


Figura 7: Diferencia entre versiones del código de las pruebas

9. Métricas de atributos del programa

9.1. Cuantificación de métricas del programa

9.1.1. Desempeño-REQ-3

Este requerimiento se relaciona con la capacidad del sistema de soportar eficientemente a varios usuarios al mismo tiempo. La métrica que se usará para realizar esto será el tiempo promedio para responder a una solicitud web y también el número de usuarios que puede atender de manera concurrente sin incurrir en errores. Para realizar esto se usará la herramienta Locust. Los resultados de cuantizar esta métrica se pueden observar en la tabla 3

Tipo de petición	Nombre	# de peticiones	# de fallos	Tiempo promedio	Tiempo mínimo	Tiempo máximo
GET	/cargar	100	0	1017	1003.05	1041.05
GET	/exito	240	0	1019	1002.05	1092.06
GET	/index	89	0	1017	1003.05	1117.06
POST	/login	0	10	0	0	0
GET	/segmentadas	212	0	1018	1004.05	1115.06
GET	/segmentar2	51	0	1018	1006.05	1034.05
GET	/visualizar	0	159	0	0	0

Cuadro 3: Resultados de las mediciones realizadas con Locust

9.1.2. Seguridad-REQ-3

Lo que se ahorrará con esta métrica es cuantizar la seguridad del sistema. Para esto se probará la tasa de falsos positivos, falsos negativos, verdaderos negativos y verdaderos positivos cuando se intenta ingresar al sistema. Con esto se podrá determinar la exhaustividad y precisión del sistema de ingreso al sistema. Esto se hará de manera manual sin usar herramientas. Los resultados de esto se pueden observar en el cuadro 4. Se puede determinar que la precisión del sistema de login es de 1, mismo valor que tiene la exhaustividad.

		Login interpretado	
		Correcto	Incorrecto
Login real	Correcto	2	0
	Incorrecto	0	2

Cuadro 4: Resultados de intentos de acceso al sistema

9.2. Actualización de métricas

Con respecto al requerimiento Desempeño-REQ-3 se tiene que se puede actualizar este para que establezca lo siguiente

REQ-3: El sistema debe ser capaz de soportar al menos 10 usuarios de manera concurrente y responder con no más de 1.5 segundos de demora a las peticiones web

Con respecto al requerimiento de seguridad de acceso se determinó que el sistema cumplía a cabalidad las necesidades y por lo tanto se determinó que la forma actual de este es la correcta

9.3. Modo de uso de Locust

Antes de poder realizar mediciones con Locust se debe instalar. Esto se hace de la siguiente manera

```
pip install locustio
```

Una vez que se ha instalado el programa se debe definir un archivo de Python con las secciones de la aplicación que se quieren probar. Un ejemplo de este archivo es el siguiente

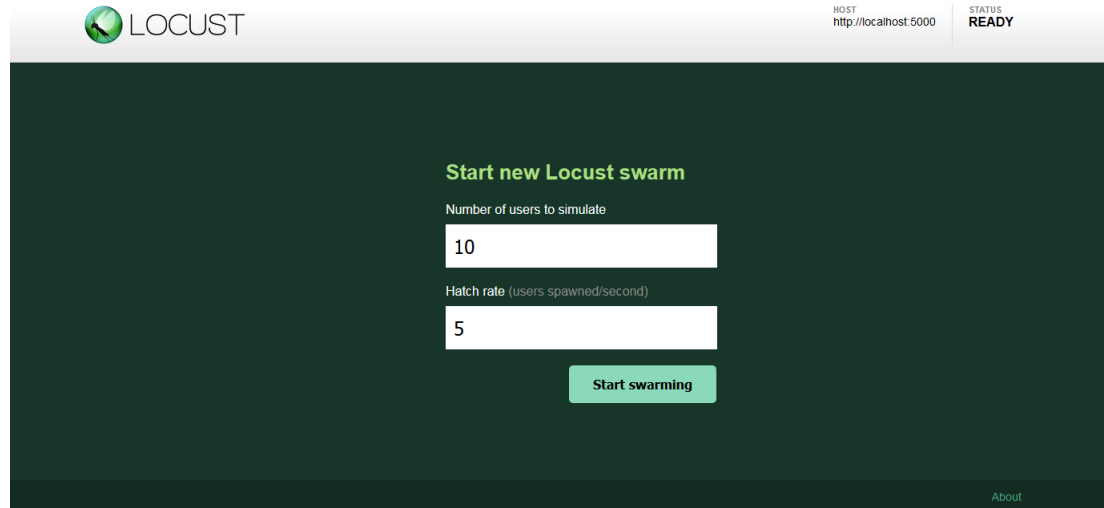
```
from locust import HttpLocust, TaskSet, task
class UserBehavior(TaskSet):
    def on_start(self):
        """ on_start is called when a Locust start
        before any task is scheduled """
        self.login()
    def login(self):
        self.client.post("/login", {"username":
        "rodrigo@ucr.ac.cr", "password":"pwd-de-rodrigo"})
    @task(2)
    def index(self):
        self.client.get("/index")
    @task(1)
    def segmentar(self):
        self.client.get("/segmentar2")
class WebsiteUser(HttpLocust):
    task_set = UserBehavior
    min_wait = 5000
```

```
max_wait = 9000
```

Lo que hace este archivo es definir como inicializar las pruebas, establecer que secciones de la página web se quieren probar y establecer algunos parámetros para las pruebas. Lo siguiente que se debe hacer es inicializar la herramienta, lo que se hace con la siguiente instrucción

```
locust -f locustfile.py --host=http://localhost:5000
```

Una vez que se ha hecho eso se inicializa una interfaz web, que puede observarse en la figura 8. En esta primera pantalla se debe definir con cuantos usuarios se quiere probar la aplicación. Después de esto se inician las pruebas, y el resultado puede observarse en la figura 9



The screenshot shows the Locust web interface. At the top, there is a header with the Locust logo on the left, the host address 'http://localhost:5000' in the center, and the status 'READY' on the right. The main content area has a dark green background. It features a section titled 'Start new Locust swarm' in yellow. Below this title, there are two input fields: 'Number of users to simulate' with the value '10' and 'Hatch rate (users spawned/second)' with the value '5'. A green button labeled 'Start swarming' is positioned below these fields. At the bottom right of the interface, there is a small link labeled 'About'.

Figura 8: Definición de los usuarios con los que se quiere probar la aplicación con Locust

10. Especificación de pruebas unitarias

Las pruebas unitarias que se han definido hasta el momento son 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.
 - Entrada: La entrada es una petición GET a la pagina */index*
 - Salida: La salida es el código 200 si todo funciona correctamente

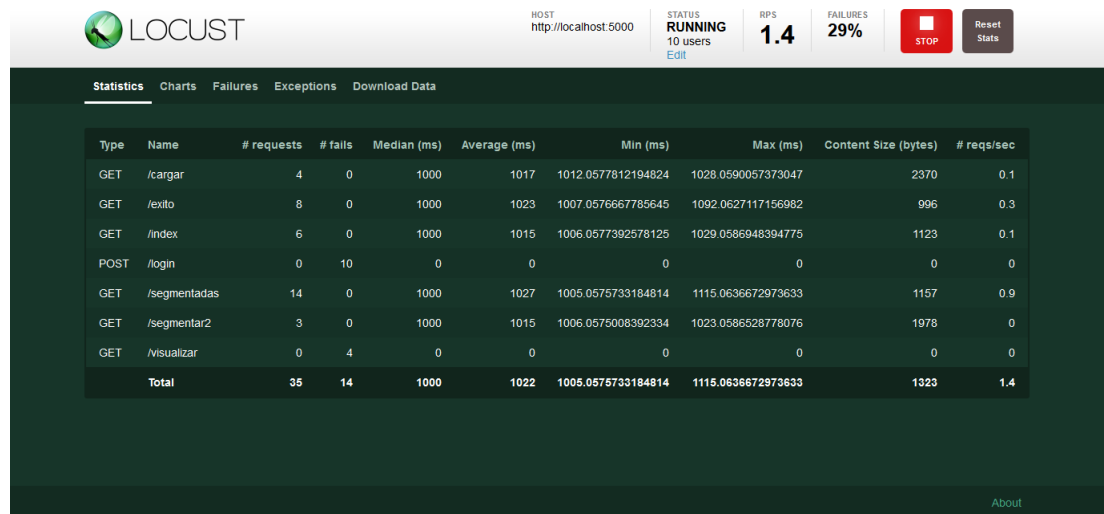


Figura 9: Ejemplo de pruebas con Locust

2. Prueba de envío de imagen al servidor: 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
 - Entrada: Recibe como entrada la petición POST que contiene la imagen
 - Salida: La salida es el código 200 si todo funciona correctamente
3. Prueba de envío de 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
 - Entrada: Recibe como entrada la petición POST que contiene las imágenes
 - Salida: La salida es el código 200 si todo funciona correctamente
4. Prueba de despliegue de imágenes: 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.
 - Entrada: Recibe como entrada la petición GET a la pagina que se encarga de mostrar las imágenes
 - Salida: La salida es el código 200 si todo funciona correctamente

5. Prueba de pagina de segmentación: Esta prueba tiene como objetivo determinar si la pagina donde se inicia el proceso de segmentación funciona correctamente
 - Entrada: Recibe como entrada la petición GET a la pagina que tiene el menú de segmentación
 - Salida: La salida es el código 200 si todo funciona correctamente
6. Prueba de pagina de imágenes segmentadas: Esta prueba tiene como objetivo determinar si la pagina donde se muestran las imágenes segmentadas funciona correctamente
 - Entrada: Recibe como entrada la petición GET a la pagina que muestra las imágenes segmentadas
 - Salida: La salida es el código 200 si todo funciona correctamente
7. Prueba de login: Esta prueba tiene como objetivo determinar si la funcionalidad de login funciona correctamente
 - Entrada: Recibe como entrada la petición POST a la pagina que tiene maneja el proceso de login
 - Salida: La salida es el código 200 si todo funciona correctamente
8. Prueba de segmentación: Esta prueba tiene como objetivo determinar si la funcionalidad de segmentación funciona correctamente
 - Entrada: Recibe como entrada la dirección donde se almacenan las imágenes y la dirección donde dejar el resultado
 - Salida: La salida es un *True* si todo funciona correctamente

11. Conclusión

En este trabajo se ha podido observar el planeamiento alrededor de varios puntos relacionados con el aseguramiento de la calidad del sistema de segmentación celular. Se mencionó la cuantización de métricas y la asociación de ítems de diseño con requerimientos y elementos de implementación entre otras cosas. Con la realización de este documento de