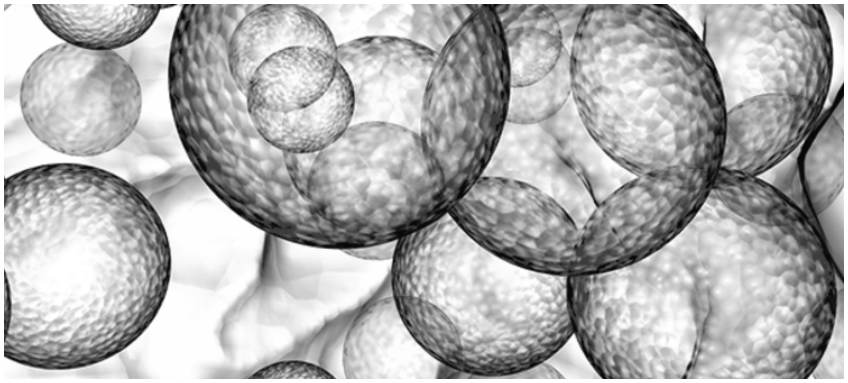
Instituto Tecnológico de Costa Rica, Sede Cartago
Escuela de Ingeniería en Computación
Aseguramiento de la Calidad del Software
Proyecto Semestral I. Segmentación de células
Profesor:
M. Sc. Saúl Calderón Ramírez
Estudiantes:
José Enrique Alvarado Chaves, 201129079
Reggie Stewart Barker Guillén, 2014050578
Joel Barrantes Garro, 2013120962
Joel Schuster Valverde, 2014096796
6 de marzo del 2017

Prueba de concepto



El presente documento es parte de la prueba de concepto del primer proyecto del curso de "Aseguramiento de software", en el cual se pretende realizar un conjunto de investigaciones que permitirán ver la factibilidad de las ideas, tecnologías o funciones investigadas por el grupo de trabajo.

Ahora bien, para la lectura del documento se presentará primero un conjunto de experiencias las cuales orientaron y conllevaron al equipo escoger las tecnologías-herramientas. Luego una estructura de alto nivel del problema y las principales subpartes de la primera parte del proyecto. Al final se presentan algunos de los problemas que se tuvieron en los algoritmos sobre las imágenes.

Experiencia:

El primer acercamiento del grupo fue para tener una definición acerca de las herramientas para el uso de:

- Repositorio de versiones
- Versión de OpenCV
- Versión de java.
- Versión de IDE.

Para la selección del repositorio de versiones se seleccionó github al inicio por convención del equipo, debido a su experiencia en este. Por otra lado en la selección de la versión de OpenCV se acordó utilizar la versión con más documentación para una mayor facilidad a la hora del desarrollo del proyecto, la documentación de OpenCV conllevo a seleccionar java 1.7 debido a que se presenta la recomendación utilizar dicha versión para una mayor estabilidad en el sistema. Para la selección del IDE se presentó la utilización de Eclipse Indigo, siendo este una de las versiones viejas de la plataforma, pero al mismo tiempo una de las que posee una interfaz de usuario más estable y veloz.

Para el segundo acercamiento se presentaron algunas propuestas y se cambiaron algunas de las herramientas presentadas.

Cambios realizados:

- Versión de java.
- Versión de OpenCV.
- Versión de IDE.

Propuestas realizadas:

- Utilizar Angular 2 para el desarrollo del cliente.

- Utilizar Tomcat Native para crear el lado de servidor.
- Utilizar Spring para crear el lado de servidor.
- Utilizar Jersey para crear el lado de servidor.

Los cambios realizados sobre la versión de java se debieron a que las actualizaciones sobre la versión de OpenCV facilita varias funciones sobre el proyecto a desempear, pero como consecuente se pide una actualización de la versión de java utilizada a la versión 1.8. Al mismo tiempo que tres integrantes se encargaban del área de openCV y veían esta dificultad, también se encontraba el compaero restante que se encargó del lado del servidor, en el cual encontraba más facilidad el usar java 1.8 debido a las características que se proveían en las versiones actualizadas de las herramientas de Spring y Jersey. Al cambiar a la versión de java 8, se cambio además el IDE de la versión Indigo por Neon, debido a que posee mas facilidades a nivel de interfaz de usuario, pero se sacrifica rendimiento.

Algunas otras herramientas probadas fueron Jersey y Tomcat, que permiten la realización de servicios web con java, pero el equipo ha optado por el uso de la herramienta Spring para la construcción de un servicio web RESTful. Las principales razones para la utilización de la misma, es la facilidad del manejo de dependencias que tienen Spring-maven, uso de patrones de diseño que proporcionan varias facilidades como por ejemplo la instanciación de objetos en java y por último la facilidad de comunicar la misma con diferentes tecnologías ya sean de bases de datos o otros servicios.

Por último, una de las áreas que se empezó a investigar por parte de dos de los miembros fue el uso de angular 2, para el desarrollo del lado del servidor, que proporciona facilidades como el manejo de dependencias del lado del cliente y además un conjunto de importante de características que lo hacen amigable al usuario. Por problemas con respecto al aumento en la curva de aprendizaje en esta otra herramienta se decide utilizar Spring MVC temporalmente para proporcionar al cliente una interfaz que permita interactuar de forma amigable.

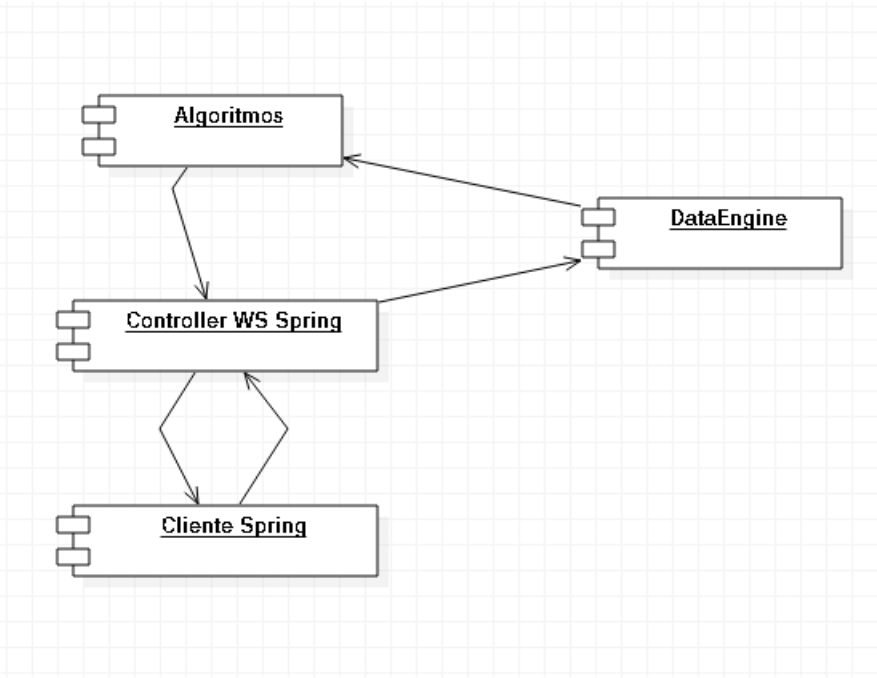
Como resultado a esta primera etapa del proyecto se estructura el sistema con las siguientes herramientas:

- Spring MVC 4.0.
- Spring GS Uploading files 0.1.0.
- Maven 4.0.0.
- OpenCV 3.2.0.

Para los próximos entregables se verá la utilización formal de Spring JPA y hibernate.
Enlace a proyecto en repositorio Git: https://github.com/schust3r/PSA

Estructura de alto nivel del sistema.

Diagrama de componentes



Análisis básico de los trozos

1) Cálculo de umbral óptimo utilizando el algoritmo de Kittler

Implementación del algoritmo de Kittler determinar un valor óptimo con el fin de umbralizar una imagen.

Entrada(s):
Tipo: Mat
Matriz de píxeles correspondientes a una imagen en escala de grises.
Salida(s):
Tipo: Int
Umbral óptimo
$\tau \in \{0,1,2,...,254,255\}$

2) Umbralización de la imagen usando un valor arbitrario.
Toma como entrada una matriz de píxeles de una imagen en escala de grises para producir una matriz binaria con entradas aij, con aij =0 o aij = 255.
Entrada(s):
Tipo: Mat
Matriz de píxeles correspondientes a una imagen en escala de grises.
Tipo: Int
Umbral óptimo
$\tau \in \{0,1,2,...,254,255\}$

Salida(s):
Tipo: Mat
Matriz de píxeles correspondiente a una imagen umbralizada con entradas aij, con aij =0 o aij = 255.

3) Etiquetado de una imagen umbralizada Genera una nueva matriz representando el etiquetado de una matriz umbralizada.

Entrada(s):

Matriz de píxeles correspondiente a una imagen umbralizada con entradas aij, con aij =0 o aij = 255

Salida(s):

Tipo: Mat
Matriz con píxeles en el rango de colores RGB .

4) Carga de imagen Permitir al usuario subir una imagen desde el cliente y guardarla en el servicio web.

Lado del cliente:

Entrada(s):
Imagen proveída por el usuario final.
Salida(s):
Imagen enviada a través del protocolo de transferencia de hipertexto, proveído por la herramienta de spring.

Lado del servidor:

Entrada(s):
Imagen recibida por el lado del cliente con http, en Spring MVC.
Salida(s):
Imagen se guarda en el lado del servidor.

Repaso de la experiencia y algunos problemas encontrados en algoritmos-funciones.

Uno de los principales problemas encontrados durante el proyecto fue el manejo de versiones de las herramientas que se tenía en proceso de investigación, para posteriormente implementar en el proyecto. Por otro lado, en la investigación se tuvieron algunos problemas sobre las funciones de las herramientas escogidas, puesto que la documentación variaba considerablemente con respecto a las versiones.

Para finalizar no se tuvieron problemas a la hora de la utilización de los algoritmos de la herramienta OpenCV, puesto que una vez que el equipo logró hacer la instalación formal de todas las herramientas para el desarrollo, se invirtió tiempo en la lectura de entradas-salidas de los algoritmos necesarios para completar la primera etapa del POC.