**Carrera:** Licenciatura en Ciencias de la Computación

**Título:** *Christhin* - Analizador de Cromatografía de Capa Fina.

**Alumnos:** Cristian Tardivo y Sergio Baldani

**1. Introducción**

El proceso de Cromatografía de Capa Fina (TLC) [Sherma y Fried, 2003; Stahl y Ashworth, 1969] puede ser utilizado para analizar la composición química de los compuestos que conforman un aceite. Básicamente, el proceso consiste en sembrar una gota de aceite sobre una placa de vidrio para TLC con una película delgada de silica gel, que mediante el arrastre por capilaridad y a través de una mezcla de solventes específica, separa el aceite en los diferentes compuestos que lo conforman. Este proceso genera una película blanca con múltiples manchas coloreadas que se corresponden a los diferentes compuestos que forman el aceite, cuyo tamaño e intensidad es proporcional a la fracción másica de cada compuesto [Fuchs et al, 2011; Sherma y Fried, 2003] . Si se digitaliza esta placa de TLC, por ejemplo, tomando una fotografía, es posible mediante una manipulación gráfica de la imagen, cuantificar el porcentaje másico de cada compuesto del aceite.

*Christhin* (Chromatography-Riser-Thin) [1] es un prototipo que ha sido desarrollado por el grupo de Simulación Aplicada a Procesos Tecnológicos (SIMAP) de la Facultad de Ingeniería de la Universidad Nacional de Río Cuarto. Actualmente, *Christhin* está implementada como un paquete para Octave [2], y permite analizar cuantitativamente los resultados de una placa de TLC de forma rápida y eficiente. Sin embargo, *Christhin* tiene algunas limitaciones, que producen que la utilización de la herramienta requiera asistencia constante por parte del usuario: la imagen recibida por la aplicación debe estar orientada verticalmente, el usuario debe indicar los límites inferior y superior de la placa de TLC digitalizada, el usuario debe identificar manualmente la posición de cada mancha en la placa, etc. Estas actividades no sólo disminuyen la usabilidad de la herramienta, sino que además pueden afectar seriamente la precisión de los resultados obtenidos del análisis.

**2. Objetivos**

En este trabajo proponemos rediseñar y reimplementar *Christhin*, teniendo como principales objetivos contribuir a la usabilidad de la herramienta mediante la automatización de varias de las tareas que en el prototipo actual deben ser realizadas manualmente, y a la portabilidad, mantenibilidad y extensibilidad de la herramienta mediante un diseño adecuado. Para lograr esto, planeamos investigar librerías gráficas de software libre disponibles (como GIMP [3], ImageJ [4], etc.) que puedan ser integradas a *Christhin* para la manipulación de las imágenes, permitiéndonos entre otras cosas, detectar automáticamente la posición y contorno de cada una de las manchas formadas por cada compuesto del aceite sobre la placa de TLC, y estudiar alternativas de diseños arquitectónico y de clases para conseguir, para esta aplicación, un buen nivel de flexibilidad en cuanto a mantenibilidad y extensibilidad.

La herramienta a desarrollar contará, en primera instancia, con una versión standalone que podrá ser utilizada localmente por el usuario en su computadora. Por otro parte, el diseño de la aplicación debe admitir la provisión de una interfaz web, que permita al usuario utilizar la herramienta de manera remota desde un navegador, sobre una instalación de la misma como un servicio web. Para esto, el diseño y desarrollo de *Christhin* será pensado inicialmente en dos módulos separados: el Back-end será la parte proveedora del servicio, y la encargada de las tareas de mayor complejidad computacional, mientras que el Front-end será la interfaz gráfica que le permitirá al usuario acceder a dicho servicio. En particular, el Back-end, que se encargará de llevar a cabo el análisis a la placa de TLC, será desarrollado en el lenguaje de programación Java, respetando los criterios elementales de diseño orientado a objetos y aplicando patrones de diseño adecuados para el caso [Gamma et al. 1995], de manera de facilitar la extensibilidad, mantenimiento y parametrización en la solución. El Front-end de la herramienta contará con una interfaz web desarrollada en el framework AngularJS [5]. Para que la comunicación entre los módulos se lleve a cabo de manera correcta, necesitaremos definir una API-REST adecuada y equipar al Back-end con módulos de red apropiados [Tanenbaum 2003].

Como objetivo secundario, planeamos la posibilidad de desarrollar un Front-end para Android, que permita utilizar la herramienta directamente desde un celular o tablet y facilite al usuario la toma de fotografías de placas de TLC, envío de imágenes para análisis, y recuperación de resultados por parte del Back-end. El desarrollo de esta interfaz excede el alcance de este plan de trabajo; como parte del mismo se desarrollará solamente un prototipo, que sirva como prueba de concepto y permita evaluar una integración adecuada con el Back-end.

**3. Plan de trabajo**

**Fecha estimada de Inicio:** Abril 2015

**Fecha estimada de Finalización:** Diciembre 2015

**Duración Total Estimada de la Tesis (Desarrollo + Cursado de materias) : 9 meses**

**Duración Estimada del Trabajo de Tesis (sin contar el cursado): 7 meses**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actividad** |  |  | **Duración** |
| Comprender el problema. Estudiar conceptos teóricos necesarios. Buscar software libre para la manipulación de imágenes y estudiar cómo pueden usarse. Instalación, configuración y primeros experimentos. |  |  | 2 meses  **(1)** |
| Diseño de *Christhin*. | Comprender el funcionamiento actual de la herramienta. Detectar los algoritmos empleados para el cómputo de TLC que serán reutilizados en la nueva versión. |  | 15 días(en paralelo a la anterior) **(2)** |
|  | Análisis y Diseño gral de la herramienta, siguiendo algún patrón de diseño. |  | 1 mes(en paralelo a la actividad 1 y a continuación de la 2) **(3)** |
|  | Diseñar la **API REST** que usará la herramienta para que el Front-end se comunique con el Back-end. |  | 1 mes  **(4)** |
|  | Implementación del prototipo de la interfaz gráfica  (sin funcionalidad) para las versiones standalone y web. | - Análisis, Diseño e Implementación de la interfaz gráfica.  -Testing del prototipo. | 1 mes(en paralelo a la anterior)  **(5)** |
| Implementación de *Christhin*. | Desarrollo del Back-end | - Análisis, Diseño e Implementación de la lógica del juego.  - Testing de unidad. | 2 mes  **(6)** |
|  | Desarrollo del módulo de Red. | - Análisis, Diseño e Implementación del módulo de Red. | 1 mes (en paralelo a la anterior)  **(7)** |
|  | Integración entre el Back-end y el Front-end para las versiones standalone y web. |  | 1 mes y medio **(8)** |
|  | Implementación de prototipo de Front-end para Android y prueba de integración con el Back-end. |  | 1 mes y medio (esta actividad corresponde al objetivo secundario)  **(9)** |
| Informe, manual de usuario y presentación |  |  | 3 meses (en paralelo con la anterior)  **(10)** |
| Cursado de Materias de 5to Año de Licenciatura. | Cursado de materias de 1er. cuatrimestre |  | 3 meses (en paralelo a las demás)  **(11)** |
|  | Cursado de materias de 2do. cuatrimestre | - | 3 meses y medio (en paralelo a las demás)  **(12)** |

Dichas tareas están organizadas de acuerdo al siguiente cronograma:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Abril | May | Jun | Jul | Agos | Sept | Oct | Nov | Dic |
| 1 | x x | x x |  |  |  |  |  |  |  |
| 2 |  | x |  |  |  |  |  |  |  |
| 3 |  | x | x |  |  |  |  |  |  |
| 4 |  |  | x x |  |  |  |  |  |  |
| 5 |  |  | x | x |  |  |  |  |  |
| 6 |  |  |  | x x | x x |  |  |  |  |
| 7 |  |  |  |  | x x |  |  |  |  |
| 8 |  |  |  |  | x | x x |  |  |  |
| 9 |  |  |  |  |  |  | x x | x |  |
| 10 |  |  |  |  |  |  | x x | x x | x x |
| 11 | x x | x x | x x |  |  |  |  |  |  |
| 12 |  |  |  |  | x | x x | x x | x x |  |

Nota: cada x , representa 15 días.

**4. Herramientas y Librerías**

[1] http://www.ing.unrc.edu.ar/grupos/simap/developments/index.html

[2] http://www.gnu.org/software/octave/

[3] http://www.gimp.org

[4] http://imagej.net

[5] https://angularjs.org

**5. Referencias bibliográficas**

[Fuchs et al, 2011] Fuchs, B., Sub, R., Teuber, K., Eibisch, M., y Schiller, J. (2011). Lipid analysis by thin-layer chromatography - a review of the current state. Journal of Chromatography A, 1218(19):2754 2774. Planar Chromatography.

[Gamma et al. 1995] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-oriented Software, Addison-Wesley Longman Publishing Co., Inc., 1995.

[Sherma y Fried, 2003] Sherma, J. y Fried, B. (2003). Handbook of thin-layer chromatography, volume 89. CRC press.

[Stahl y Ashworth, 1969] Stahl, E. y Ashworth, S. (1969). Thin Layer Chromatography. McGrawHill.

[Tanenbaum 2003] A. S. Tanenbaum, Redes de computadoras, cuarta edición, Prentice Hall, 2003.