

Educational Web Tool for Digital Image Processing

Martin Poletti*, Horacio Legal*, Jacques Facon†, Claudio Barua* and Jose Luis Vazquez*

*Facultad Politécnica, Universidad Nacional de Asunción
San Lorenzo, Paraguay

Email: {mpoletti,hlegal,cbarua,jlvazquez}@pol.una.py

†Programa de Pós-Graduação em Informática aplicada
Pontifícia Universidade Católica do Paraná, Brazil

Email: facon@ppgia.pucpr.br

Abstract– Due to its versatility, the image processing area offers a very wide range of techniques to solve challenges in an effective way linked to fields such as medicine, agriculture, biology, industrial automation and document processing. Therefore a correct and advanced training of professionals in this area is an important task. In this sense, a new educational image processing tool is currently being developed at the Facultad Politécnica of the Universidad Nacional of Asunción. The development focused on improving the interaction between students and teachers and also showing new advances in digital image processing area. To achieve this goal, a tool is being developed to be expanded in the future to be adapted to new challenges and different audiences. The first stage of development was completed, which allowed developing an extendable basic tool in the near future.

I. INTRODUCCIÓN

En la Facultad Politécnica (FP-UNA), en el 2007 se crearon dos programas de postgrado, Maestría y Doctorado en Ciencias de la Computación. Junto con estos programas, se creó -entre otros- el Grupo de Procesamiento Digital de Imágenes. Este grupo es integrado por alumnos de Postgrado, tesis de Grado, pasantes de los últimos años de Ingeniería en Informática e investigadores a tiempo parcial. Las investigaciones vigentes y concluidas, realizadas en cooperación con otros Laboratorios de la Universidad y del exterior han generado publicaciones internacionales en revistas, artículos completos en congresos y resúmenes [1], [2], [3], [4].

En la actualidad, los cursos de postgrado en el área de Procesamiento Digital de Imágenes (PDI), se desarrollan con el uso de una herramienta llamada FePI (Ferramenta de Procesamiento de Imagens), diseñada en la Pontifícia Universidade Católica do Paraná (PUCPR) para complementar el uso de un libro didáctico [5]. La misma presenta varias limitaciones que fueron apareciendo con el correr del tiempo y el cambio en las tecnologías, que la vuelven poco práctica para su uso; además, el mantenimiento del referido software fue discontinuado. Durante este lapso, han surgido nuevos algoritmos y métodos importantes que deben ser incorporados en el desarrollo de los cursos de procesamiento de imágenes, y con los cuales no cuenta esta herramienta. De allí la necesidad del desarrollo de una nueva herramienta que utilice las nuevas tecnologías y

permita tener un control y soporte continuo sobre los métodos necesarios para el desarrollo de la materia de procesamiento de imágenes, en los cursos de postgrado como en los cursos de grado, y ayude a compartir con la comunidad los nuevos avances que se están logrando en el área.

La herramienta se desarrolla en conjunto con los alumnos de grado como una pasantía en el área de procesamiento digital de imágenes y en el área de desarrollo de aplicaciones web de la Facultad, guiados por investigadores del área.

Para lograr los objetivos mencionados, la herramienta debe cumplir con los siguientes requerimientos:

- Ejecución online de algoritmos de procesamiento y análisis de imágenes.
- Asignación y corrección de tareas.
- Identificación de los alumnos y documentación de su progreso en el curso.

Este trabajo está estructurado de la siguiente manera: en la sección II se presentan las tecnologías analizadas para el desarrollo de la herramienta, en la sección III se detalla el ciclo de vida del desarrollo del software, en la sección IV se muestra la propuesta del modelo de software a desarrollar, en la sección V se describe la metodología de evaluación de usabilidad de un software, en la sección VI se presentan algunos resultados preliminares, en la sección VII se mencionan algunas conclusiones y por último en la sección VIII se proponen trabajos futuros.

II. TECNOLOGÍAS

El procesamiento digital de imágenes es una agrupación de técnicas utilizadas con el fin de aumentar la calidad o encontrar cierta información relevante en las imágenes digitales [6]. Es una de las disciplinas que se ubica dentro de las Ciencias de la Computación, por lo que su estudio se considera importante en la carrera de Informática. Entre las áreas de aplicaciones del PDI se pueden citar la biología y genética [7], la agricultura [8], defensa e inteligencia militar [9], procesamiento de documentos, y automatización industrial e imágenes satelitales [10]. De todo esto surge la necesidad de contar con una herramienta para su uso en los cursos de PDI desarrollados en la FP-UNA.

En la actualidad, existen varias tecnologías que pueden ser utilizadas para el desarrollo de la herramienta de PDI. En esta

sección se mencionan, de manera breve, aquellas que fueron analizadas en la etapa de diseño de la nueva herramienta.

II-A. Jboss

Jboss es un servidor de aplicaciones que implementa Java EE¹, es de código abierto e implementado totalmente en lenguaje de programación Java, lo que permite ejecutar en cualquier equipo que posea una máquina virtual Java disponible [11].

II-B. Matlab

Matlab es un lenguaje de alto nivel para computación numérica y desarrollo de aplicaciones, posee un entorno interactivo para el desarrollo de nuevos algoritmos con un lenguaje de programación propio [12]. Además, dispone de la herramienta Image Processing Toolbox, que cuenta con un conjunto de algoritmos, funciones y aplicaciones para el procesamiento, análisis y visualización de varios tipos de imágenes [13].

II-C. Opencv

Es una biblioteca de visión por computadora y aprendizaje de máquinas, de distribución libre bajo la licencia BSD. La librería cuenta con algoritmos optimizados, que incluye un amplio conjunto de algoritmos clásicos y del estado del arte; tiene soporte multiplataforma para GNU/Linux, Mac OS X y Windows [14]. Está programado en C y C++ y cuenta con interfaces para Python, Java, Matlab y Octave [15].

II-D. ImageJ

ImageJ es un programa de procesamiento de imágenes digitales, tiene las ventajas de poseer un arquitectura abierta que proporciona extensibilidad por medio de plugins escritos en Java. Es posible utilizarlo como una biblioteca para el acceso a los algoritmos contenidos dentro de la misma. Fue desarrollado en el National Institutes of Health (NIH) de Estados Unidos [16].

II-E. FePI

FePI es una herramienta para el procesamiento digital de imágenes desarrollada por David Menotti y Jacques Facon, en la Pontificia Universidade Católica do Paraná. Proporciona una interfaz de usuario donde se despliegan las imágenes seleccionadas y se listan los posibles algoritmos a realizar sobre esas imágenes. Puede ser utilizado para ver el funcionamiento de los algoritmos de procesamiento de imágenes y compararlos con implementaciones propias, así como para realizar ejercicios en los que dada una imagen de entrada se quiera obtener una imagen con ciertas características luego de haber aplicado los algoritmos disponibles. Actualmente, la herramienta ya no es adecuada para su uso debido a la incompatibilidad con los nuevos sistemas operativos (Windows 7 en adelante).

¹Java EE, Java Enterprise Edition

Con el objetivo de establecer una idea de la herramienta que se viene utilizando en los cursos de postgrado de la FP-UNA, se muestran a continuación algunas capturas de pantalla del FePI. En la Figura 1 se puede observar cómo una imagen a color (imagen superior) es transformada a una imagen en escala de grises (imagen inferior).

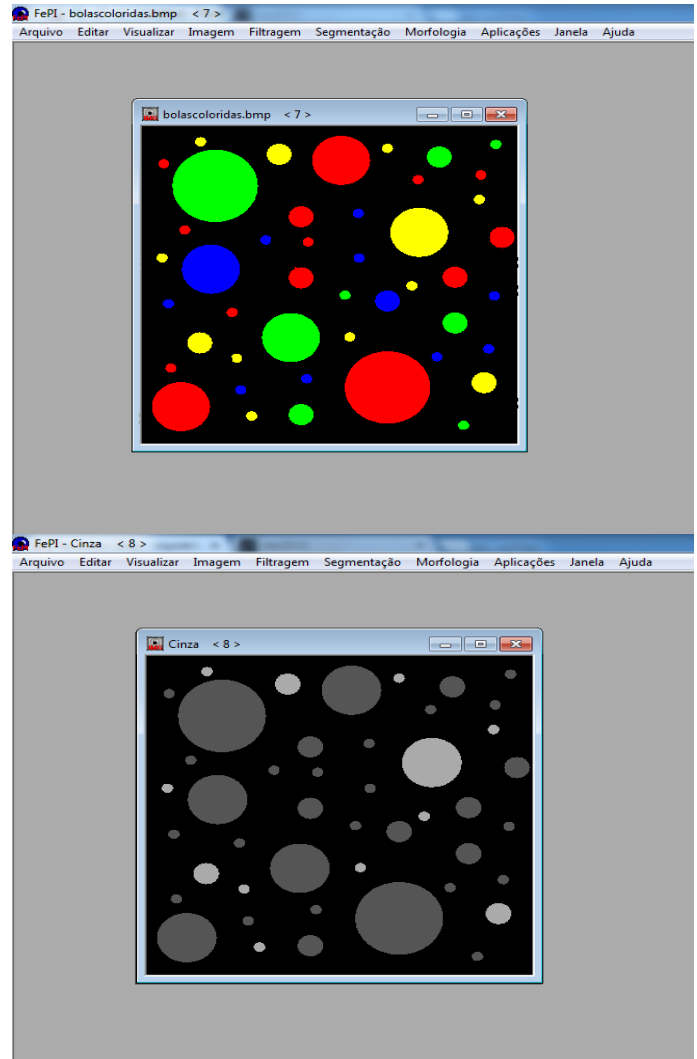


Figura 1. Transformación de una imagen a color a escala de gris

En la Figura 2 se muestra el resultado del proceso de erosión morfológica en escala de grises. En la imagen superior se puede ver la ventana donde se introducen los parámetros respecto a la forma y al tamaño del elemento estructurante y en la imagen inferior se ve el resultado de aplicar la erosión a dicha imagen.

A continuación se presentan los algoritmos disponibles en la herramienta utilizada actualmente y que se busca reemplazar en esta propuesta.

- Filtrado
 - Filtro lineal
 - Filtro no lineal
 - Ecuilización

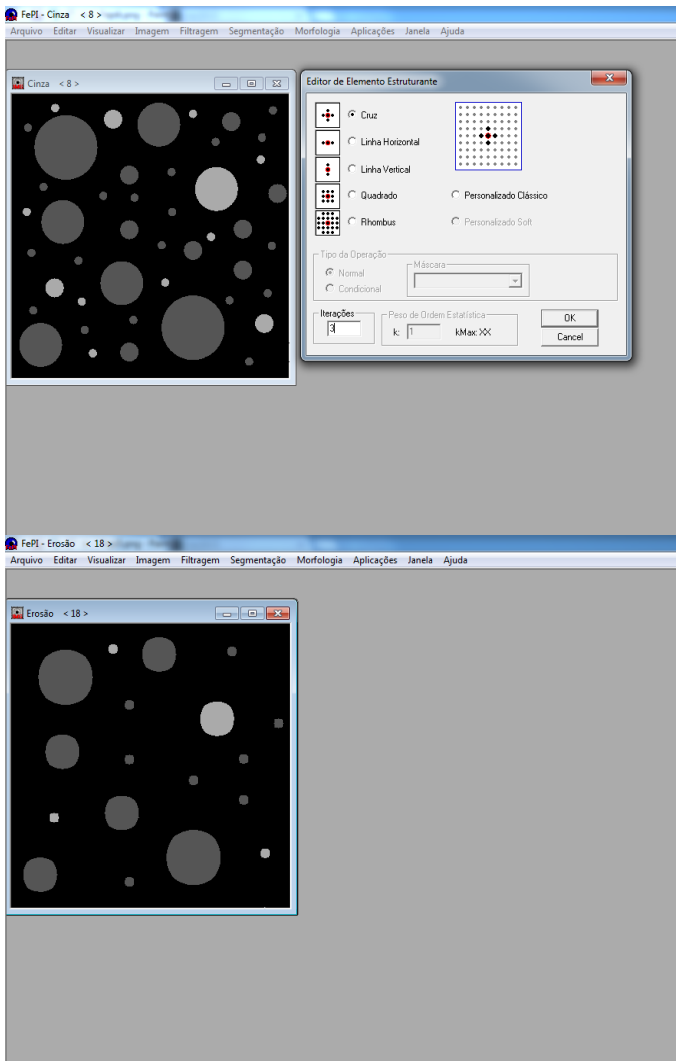


Figura 2. Operación de erosión morfológica

- Realce
- Ruído
- Segmentación
 - Afinamiento
 - Detección de bordes
 - Umbralización
 - Multi-Umbralización
- Morfología
 - Dilatación
 - Erosión
 - Apertura
 - Cierre

III. METODOLOGÍA DE TRABAJO

Para el desarrollo de un sistema software se debe seguir un modelo de proceso de software, que es la secuencia de actividades que permiten la elaboración de un producto software [17]. Existen varios modelos que podrían ser utilizados en el proceso de desarrollo de la herramienta propuesta, cada

uno con enfoques y características distintas. Para este trabajo se consideró más adecuado al modelo iterativo e incremental, debido a la necesidad de tener un producto operativo que pueda ser utilizado en los cursos, sin afectar el desarrollo e inserción de nuevos algoritmos en la herramienta.

A continuación se presentan, brevemente, el modelo en cascada y el modelo iterativo e incremental, analizados en esta etapa.

III-A. Modelo en cascada

Según Ian Sommerville [17], el modelo en cascada considera las actividades fundamentales del proceso como especificación, desarrollo, validación y evolución del software. Según este modelo, en las siguientes etapas se desarrollan las principales actividades del proceso:

- Análisis y definición de requerimientos
- Diseño del sistema y del software
- Implementación y prueba de unidad
- Integración y prueba de sistema
- Operación y mantenimiento

Las etapas se realizan una seguida de otra, con una evaluación al término de cada etapa para la retroalimentación y la corrección de errores.

III-B. Modelo iterativo e incremental

En Ingeniería de Software, el modelo iterativo es una estrategia de desarrollo de software en la que se reserva tiempo para revisar y mejorar las partes del sistema que ha sido desarrollado, basado en la retroalimentación del usuario [18].

En cada iteración se agregan funcionalidades y se entrega al cliente una versión mejorada del producto. Las iteraciones se repiten hasta tener un producto completo. Una de las ventajas de este modelo es que se genera un software operativo de forma rápida y en etapas tempranas del ciclo de vida del software, que el cliente puede utilizar mientras espera el producto final.

El modelo incremental, a su vez, es una estrategia que permite desarrollar partes del sistema en diferentes momentos para ser integrados a medida que se completan [18].

Como se mencionó anteriormente, este modelo fue considerado más apropiado como metodología a seguir durante el desarrollo de la herramienta debido a las ventajas mencionadas.

Una vez vistas las tecnologías analizadas y la metodología de trabajo seleccionada, se presenta la propuesta que reemplazará al FePI, utilizado actualmente en la FP-UNA.

IV. PROPUESTA

El modelo propuesto para la nueva herramienta se basa en una arquitectura cliente-servidor (ver Figura 3), en la cual el cliente accede a una interfaz por medio de un navegador conectado a internet, siendo el servidor el encargado de realizar el procesamiento de las imágenes. Esto presenta la ventaja que los usuarios se abstraen de los requisitos de hardware y software necesarios para ejecutar la herramienta, permitiendo así mayor accesibilidad desde cualquier tipo de dispositivo.



Figura 3. Arquitectura Cliente-Servidor

Las funcionalidades que se buscan en la herramienta para satisfacer las necesidades de los cursos son:

- Identificar a los usuarios como alumnos o profesores.
- Seleccionar y desplegar las imágenes para su procesamiento o análisis.
- Personalizar el menú mostrando los algoritmos disponibles para el tipo de imagen, sean estas binarias, en escala de gris o a color.
- Proveer a los alumnos la opción para registrarse en un curso y ver las tareas asignadas por el profesor.
- Registrar las acciones realizadas por el alumno para resolver el problema.
- Permitir a los profesores crear tareas para los alumnos de los cursos y corregir las ya entregadas.

IV-A. Usuarios

Se clasificó a los usuarios en tres tipos diferentes debido a que pueden cumplir roles distintos en su interacción con la herramienta:

- Anónimo: Son usuarios que acceden a la herramienta sin identificarse. Estos usuarios pueden utilizar las funcionalidades que no involucran interacción con los profesores o los cursos.
- Alumno: Son usuarios identificados en el sistema. Pueden registrarse en los cursos disponibles y completar las tareas asignadas por los profesores. El progreso en las tareas quedará documentado para la evaluación del profesor.
- Profesor: Son usuarios identificados que tienen acceso a la opción de crear cursos. Dentro de cada curso pueden asignar tareas y controlar el progreso de los alumnos.

IV-B. Interfaz

La herramienta puede ser accedida a través de internet por medio de un navegador. Esto permite a los usuarios abstraerse de los requerimientos necesarios para ejecutar la herramienta. El acceso a través de la interfaz web permite la fácil asignación de tareas y el desarrollo de las clases de laboratorio, al eliminar la necesidad de la instalación de la herramienta en diferentes máquinas.

IV-C. Servidor

El servidor permite recibir e identificar las imágenes, realizar el procesamiento y devolverla al cliente para su despliegue. También se encarga de almacenar información sobre los cursos y el progreso de los alumnos.

Además, en el área de procesamiento de imágenes es necesario el acceso a bases de datos de imágenes, lo cual no siempre es una tarea trivial. Esta herramienta permitirá centralizar los trabajos y almacenar las imágenes obtenidas en los diversos procesos para su uso futuro.

IV-D. Metodología de trabajo

Como se mencionó anteriormente, para el desarrollo del software se utiliza un modelo iterativo incremental. Esto permite desarrollar la herramienta por partes, consiguiendo así un rápido acceso a las funciones más importantes y una retroalimentación a medida que se progresa en el desarrollo de la herramienta.

En el siguiente capítulo se describe la metodología de evaluación consistente en pruebas de usabilidad.

V. EVALUACIÓN DE USABILIDAD

Durante el proceso de desarrollo de un software, para los desarrolladores es difícil evaluar la usabilidad de una herramienta, debido a que están muy familiarizados con la misma. Por eso, durante el desarrollo de la nueva herramienta se realizaron pruebas con alumnos de la carrera de Ingeniería en Informática (FP-UNA), en la asignatura Electiva III (Procesamiento Digital de Imágenes).

V-A. Metodología

Las pruebas se llevaron a cabo con la colaboración de 16 estudiantes de PDI y se realizaron en las instalaciones de la FP-UNA. La cantidad de alumnos supera el mínimo recomendado según el estudio presentado en [19].

V-B. Participantes

Las pruebas se realizaron individualmente y cada prueba cuenta con un encargado y un usuario:

- Encargado: Es el responsable de dirigir las pruebas. Debe facilitar los materiales como manuales, tutoriales y tareas a los usuarios que se encargarán de probar la herramienta. También se encarga de tomar los tiempos y de realizar anotaciones de los sucesos relevantes para un análisis posterior.
- Usuario: Son los que interactuarán con el sistema y no poseen experiencia previa con el mismo. Sus acciones son registradas por el encargado para su posterior análisis.

V-C. Etapas de prueba

Con cada usuario se llevaron a cabo las siguientes etapas del experimento:

- Al inicio, como entrenamiento previo, se muestran al usuario el manual y un tutorial sobre el uso de la herramienta.

- A continuación, se pide al usuario que realice tareas con diferentes niveles de complejidad. Primero realiza una tarea simple con ayuda del encargado. Las siguientes tareas presentan una mayor dificultad y deben realizarlas sin ayuda del encargado. Cada tarea se compone de varias operaciones, que deben ser realizadas correctamente para obtener el resultado esperado. Cada usuario realiza un total de 15 operaciones para resolver cuatro tareas.
- Como tercera etapa, se presenta al usuario una encuesta para evaluar su satisfacción con el producto.
- Por último, los datos de las diferentes pruebas son recopilados y guardados para su análisis.

V-D. Factores analizados

Durante las pruebas se recopilaron algunos datos acerca de los usuarios y de los procesos realizados. A partir de esos datos se analizaron los siguientes factores:

1. Errores de la herramienta (E_1): Mide la cantidad de operaciones en las cuales ocurrieron errores en la herramienta al ser utilizado por el usuario.
2. Errores de los usuarios (E_2): Mide los errores que cometen los usuarios al utilizar la herramienta.
3. Tiempo de uso: Mide el tiempo en segundos que transcurre en cada tarea, siendo la primera tarea de aprendizaje y familiarización con la herramienta. Los tiempos se comparan con los de un usuario que ya conoce la herramienta. Siendo T_{ij} el tiempo que toma el usuario i en la tarea j y T_{ci} el tiempo de control en la tarea j . Se toma como eficiencia promedio F el promedio de las eficiencias para cada usuario en cada tarea, siendo eficiencia por tarea $F_{ij} = \frac{T_{ij}}{T_{ci}}$.
4. Satisfacción del usuario: Terminadas las pruebas se presentó a cada usuario una encuesta para evaluar su satisfacción con la herramienta, y se calculó la satisfacción promedio S en base a los resultados. Además, se pidió a los usuarios agregar sugerencias a la encuesta, en caso necesario.

Para el análisis de la satisfacción con la herramienta se utilizó la escala de Likert [20] del 1 al 7 con respecto a qué tan adecuado es:

- a) El menú de opciones
- b) El mecanismo de cargar imágenes
- c) La lista de imágenes y
- d) El manual de usuario

Luego de recolectar los datos de las encuestas se utilizó un método de estandarización por rango [22] para eliminar el potencial efecto del estilo de respuesta de los usuarios [23]. Siendo:

- Min_i la menor respuesta del usuario i
- Max_i la máxima respuesta del usuario i

Para cada respuesta R del usuario i , el valor ajustado R_a se define como:

$$R_a = \frac{R - Min_i}{Max_i - Min_i} \quad (1)$$

El valor de cada respuesta se ajusta en un valor de 0 a 1.

VI. RESULTADOS

Durante la realización de este trabajo se desarrolló la base de una herramienta extensible, en una arquitectura cliente-servidor, que se usará en los cursos de procesamiento digital de imágenes, en las áreas de grado y postgrado de la FP-UNA. La herramienta puede ser utilizada ingresando al sitio www.cc.pol.una.py/wepi.

Las tecnologías utilizadas para el desarrollo fueron:

- Jboss: Como servidor de aplicaciones fue utilizado Jboss. Esto se debe a que es de código abierto y provee mecanismos de autenticación como “Seam” [21] que cumplen con las necesidades para desarrollar la herramienta. Además, los alumnos ya están familiarizados con el lenguaje Java y con herramientas para el desarrollo de software Java.
- ImageJ: Este programa fue utilizado como una biblioteca que provee estructuras y funciones básicas para la implementación de algoritmos propios. Adicionalmente, ImageJ provee una interfaz para utilizarla como herramienta, esto proporciona la ventaja de que los algoritmos implementados puedan ser incluidos en el mismo como plugins, con la posibilidad de acceder a ellos fuera de línea. En el desarrollo del proyecto fue utilizado como una biblioteca para facilitar la implementación de los algoritmos que están contenidos dentro de la herramienta.

En la primera fase del desarrollo se analizaron las herramientas que se utilizan actualmente para el desarrollo de software, así como el FePI, que es utilizado en los cursos locales de PDI. Los casos de uso que se llevaron a cabo en las primeras iteraciones del desarrollo de la nueva herramienta fueron obtenidos del FePI.

En la Figura 4 se muestran los Casos de Uso básicos que se tuvieron para la primera iteración. Como se puede observar, en esta iteración la cantidad de Casos de Uso es reducida y la funcionalidad de la herramienta es básica.

En la Figura 5 se muestran los Casos de Uso que se tuvieron para la última iteración. Como se puede observar, en esta iteración la cantidad de Casos de Uso abarca una mayor cantidad de algoritmos y funcionalidades necesarias para los cursos de la FP-UNA.

En la fase de diseño del software fueron realizados los diagramas de clase y los diagramas de secuencia. El diagrama de clase contiene una clase Imagen de la cual heredan las clases Imagen_binaria, Imagen_gris e Imagen_color. Cada clase utiliza estructuras implementadas en la librería ImageJ para representar las imágenes y utilizar los algoritmos disponibles en la misma.

En los diagramas de secuencia se representó el funcionamiento de la herramienta. Como ejemplo, en la Figura 6 se muestra el diagrama de secuencia para el caso “cargar imagen”.

En la Figura 7 se muestra la pantalla inicial del sistema propuesto. La misma no tiene una imagen cargada para

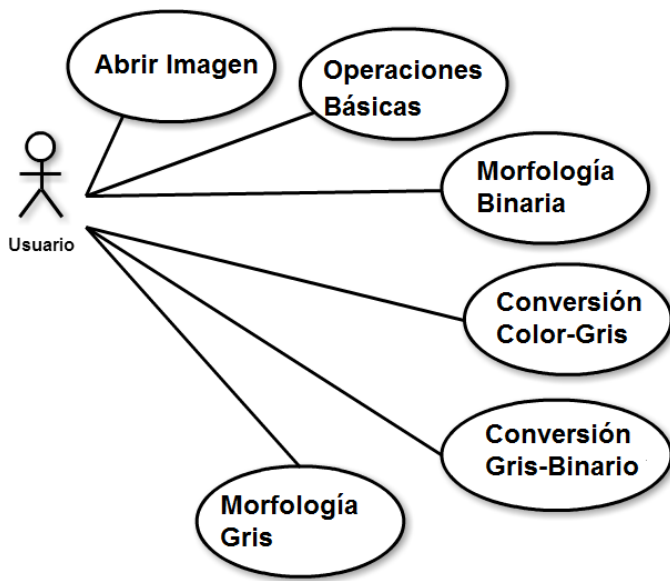


Figura 4. Diagrama Caso de Uso de la primera iteración

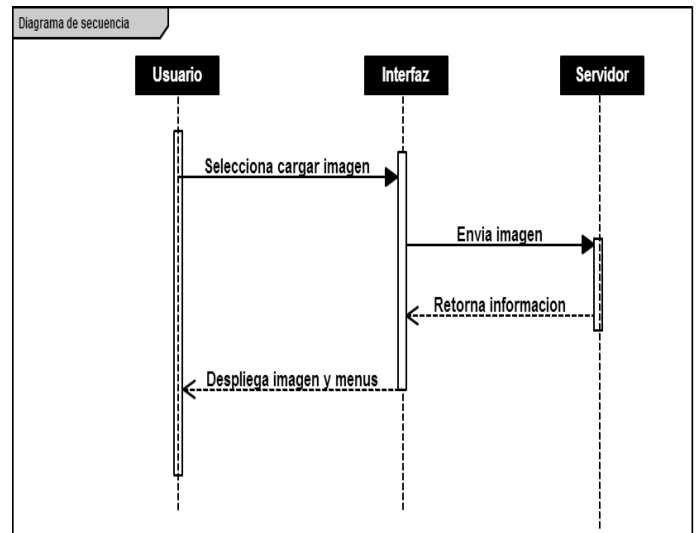


Figura 6. Cargar imagen

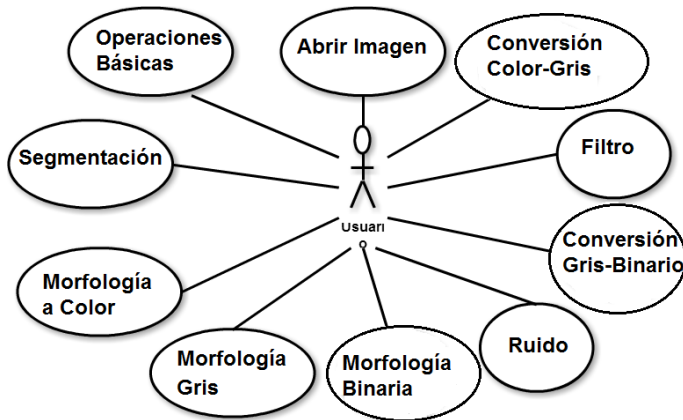


Figura 5. Diagrama Caso de Uso de la última iteración

procesar. Los elementos dinámicos de la interfaz como la barra que permite la selección de algoritmos no son desplegados hasta tener una imagen disponible. La sección que permite navegar entre las imágenes se encuentra centrada en la parte inferior y vacía debido a la ausencia de imagen.

En la Figura 8 se muestra la pantalla para subir una imagen nueva. Se debe seleccionar la imagen utilizando el botón “Add” y confirmar con el botón “Upload”.

En la Figura 9 se puede ver una imagen a color luego de ser seleccionada y subida para su procesamiento. En la parte superior se despliega la barra de opciones, en donde puede ser seleccionado el algoritmo para su ejecución, y en la parte inferior se puede observar la lista de imágenes.

En la Figura 10 se puede ver la imagen de la Figura 9 convertida a una imagen en escala de grises. El tipo de imagen es detectado por la herramienta cambiando automáticamente

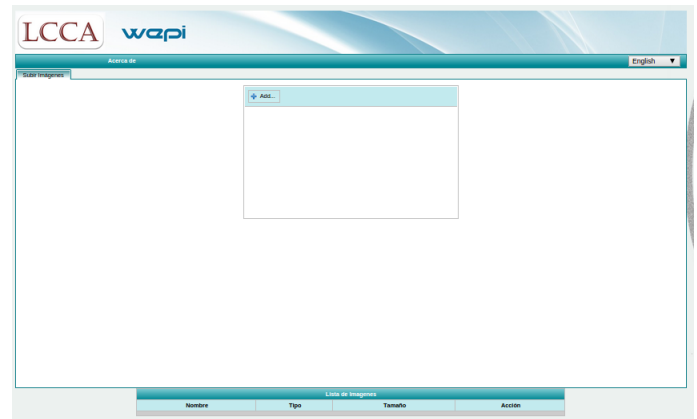


Figura 7. Pantalla inicial de la herramienta

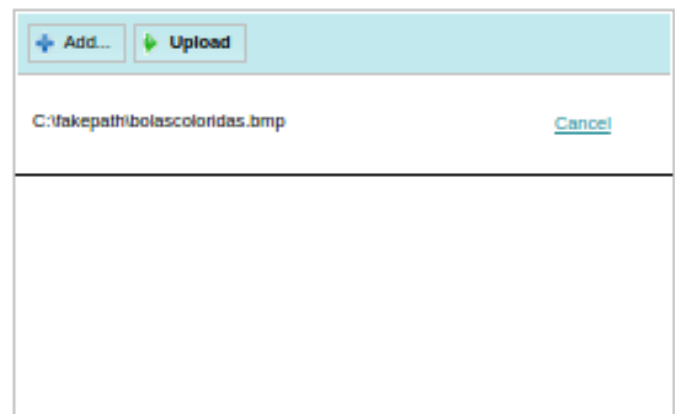


Figura 8. Subir imagen

los procesos disponibles en la barra de opciones de la parte superior.

La imagen de la Figura 11 es el resultado de umbralizar la

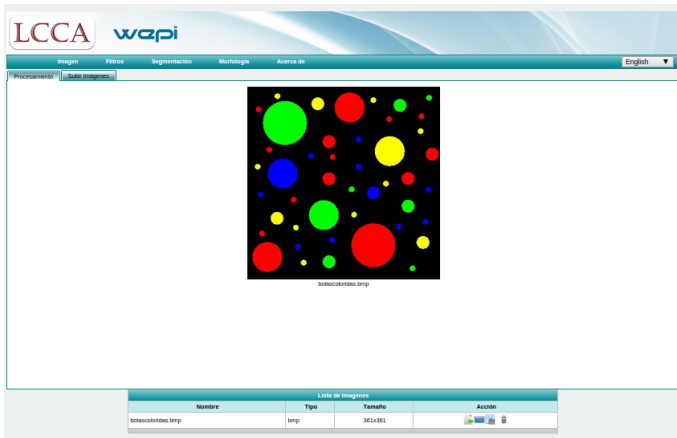


Figura 9. Imagen a color

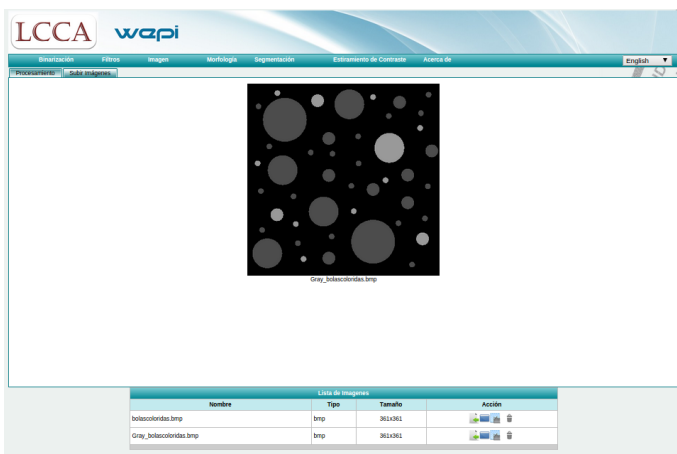


Figura 10. Imagen en escala de gris

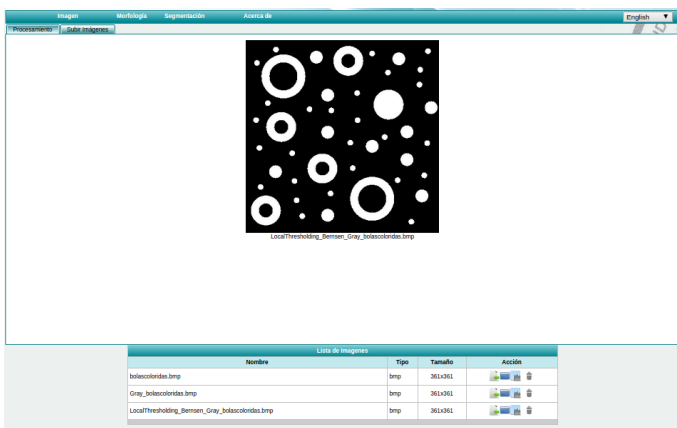


Figura 11. Imagen Binaria

imagen en escala de gris de la Figura 10. En las Figuras 9, 10 y 11 se muestran los 3 tipos de imágenes y los cambios en la barra de opciones para cada uno. Además, se puede notar la barra correspondiente a la lista de imágenes, lo que permite una navegación sencilla entre imágenes.

Po último, en la Figura 12 se muestra la ventana que es

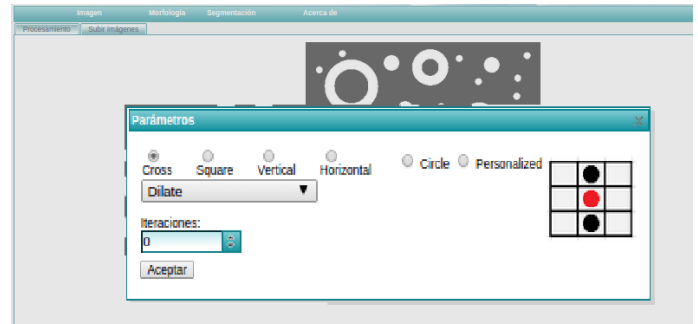


Figura 12. Parámetros para algoritmos de morfología binaria

Cuadro I
RESULTADOS DE USABILIDAD

Usuarios	E_1	E_2	F	S
1	1	1	1.1	5.25
2	0	2	1.25	6.25
3	0	0	1.3	6.5
4	1	1	1.32	6.5
5	0	1	1.16	6.5
6	0	1	1.22	5.75
7	0	0	1.31	5.5
8	1	1	1.27	6.5
9	1	0	1.31	6.25
10	0	1	1.21	6.5
11	0	1	1.18	6
12	0	1	1.34	6.25
13	0	0	1.42	6
14	0	1	1.15	6.25
15	0	1	1.22	6
16	0	0	1.24	6.25

utilizada para introducir el elemento estructural. El mismo es un parámetro necesario para un algoritmo de morfología en imagen binaria. Permite elegir entre elementos estructurales pre-definidos con formas comúnmente utilizadas o personalizar el propio elemento.

En la fase de Pruebas se realizaron pruebas de usabilidad. Estas se realizaron con el objetivo de medir el funcionamiento adecuado y la facilidad de uso de la herramienta por parte de los usuarios. En el Cuadro I se presentan los valores obtenidos por usuarios para cada uno de los factores explicados en la sección de Evaluación.

En el Cuadro II se pueden observar los valores reescalados para cada pregunta y para cada usuario. Se puede notar, por ejemplo, que para el usuario 15 no se pudo calcular el valor reescalado. Esto debido a que el resultado es una división indefinida.

Finalmente, en el Cuadro III se presenta un resumen de los promedios obtenidos por cada factor de usabilidad evaluados.

Como puede verse, durante las pruebas se detectaron cuatro errores para el valor de E_1 . Estos se produjeron al introducir los parámetros para ejecutar los algoritmos, los usuarios intentaban finalizar la operación presionando la tecla "enter" en lugar de presionar el botón de la interfaz que finaliza la operación, causando que la operación no se ejecute. En las siguientes iteraciones se introducirá la corrección a este error.

Cuadro II
RESULTADOS DE USABILIDAD

Usuarios	a)	b)	c)	d)
1	0.75	0.5	1	0
2	0	0	1	0
3	0	1	1	1
4	0	1	1	0
5	1	1	0	0
6	0	0.65	0.65	1
7	1	0	0	1
8	0	0	1	1
9	0	0	1	1
10	0	1	1	0
11	0.5	1	0.5	0
12	0	0.5	1	1
13	1	0	0.5	0.5
14	0	0	1	0
15	na	na	na	na
16	0	0	1	0

Cuadro III
RESULTADOS DE FACTORES DE USABILIDAD ANALIZADOS

Factor	Promedio
E_1	4
E_2	12
F	1.25
S	6.13

El valor de E_2 indica la cantidad de ocurrencias en las cuales los usuarios cometieron errores al interactuar con la interfaz de usuario de la aplicación. Cada uno de los 16 usuarios debía realizar 15 operaciones, en promedio, totalizando 240 operaciones durante la prueba de usabilidad. El porcentaje total de errores cometidos por los usuarios fue de 5%. Estos errores ocurrieron al momento de seleccionar el algoritmo correcto para realizar la operación. Se notó, por ejemplo, que los menús dinámicos cambiaban su estructura haciendo que necesiten una ayuda por parte del encargado para continuar. Los errores, una vez cometidos, no volvieron a ocurrir con el mismo usuario, excepto en un caso.

Los resultados de la encuesta, en relación a la satisfacción de los usuarios al utilizar la herramienta, mostraron un nivel muy bueno, obteniéndose un promedio de 6.13, superior al mínimo esperado de 5 en la escala de Likert del 1 al 7. Además, se pudo notar que la eficiencia promedio, es decir, el tiempo para realizar cada operación fue 25% más lento comparado con los usuarios de control. A medida que avanzaban las pruebas, los usuarios obtuvieron un tiempo con una diferencia del 5 al 10% con respecto al tiempo de control.

VII. CONCLUSIONES

En este trabajo se propuso un modelo de una herramienta que ayuda al aprendizaje en el área de procesamiento digital de imágenes dirigido a los alumnos de grado y postgrado de la Facultad Politécnica. Además, se llevó a cabo la implementación de la primera fase de la herramienta.

El proceso de desarrollo involucró a los propios alumnos de los cursos de grados, permitiendo así una participación activa en el proceso. Como resultado de la primera fase de la

implementación de la herramienta se cuenta con una aplicación web que permite la ejecución de algoritmos de procesamiento y análisis de imágenes necesarios para que los alumnos puedan comprender sus aplicaciones.

Los resultados de la encuesta mostraron muy buen nivel de satisfacción con la herramienta, teniendo el valor más bajo de 5.25 en el tópico sobre el menú de opciones que fue el área donde se produjeron los errores por parte del usuario. Las pruebas que se llevaron a cabo mostraron una buena respuesta por parte de los usuarios, quienes mostraron interés en la herramienta.

VIII. TRABAJOS FUTUROS

Como trabajos futuros quedan las correcciones de errores encontrados a partir de las pruebas de usabilidad. Además, se tiene pensado reestructurar los menús dinámicos para que mantengan una forma constante entre todos los cambios. Por otra parte, se continuará con el proceso de desarrollo de la herramienta para implementar las funcionalidades de la administración de usuarios y la asignación de tareas.

AGRADECIMIENTOS

Especial agradecimiento a los alumnos de la carrera de Ingeniería en Informática que fueron parte del desarrollo de la herramienta en el marco de su pasantía laboral en la FP-UNA.

REFERENCIAS

- [1] H. Legal-Ayala, J. Facon and B. Baran. "Postal envelope segmentation using learning-based approach", *CLEI Electron. J.*, vol. 11, 2008
- [2] J.L. Vazquez Noguera, H Legal Ayala, C.E. Schaerer and M. Rolon. "Mathematical morphology for counting Trypanosoma cruzi amastigotes", in *Computing Conference (CLEI), 2013 XXXIX Latin American*, 2013.
- [3] J.L. Vazquez Noguera, H Legal Ayala, C.E. Schaerer and J. Facon. "A color morphological ordering method based on additive and subtractive spaces", in *Image Processing (ICIP), 2014 IEEE International Conference on*, 2014, pp.674–678.
- [4] J.L. Vazquez Noguera, H Legal Ayala, C.E. Schaerer and M. Rolon. "RGB color space segmentation for Trypanosoma cruzi amastigotes: detection and counting", in *CNMAC 2012 XXXIV Congreso Nacional de Matemática Aplicada e Computacional*, 2012, pp.854-855.
- [5] J. Facon. "Mathematical Morphology: theory and practice" (In Portuguese), Editor Facon Jacques, 1996.
- [6] B. Chanda and D. D. Majumder. "Digital image processing and analysis". PHI Learning Pvt. Ltd., 2004.
- [7] H. Iwata and Y. Ukai. "SHAPE: a computer program package for quantitative evaluation of biological shapes based on elliptic Fourier descriptors" *Journal of Heredity* 93.5 (2002), pp. 384–385.
- [8] R. Bernstein. "Digital image processing of earth observation sensor data". *IBM Journal of Research and Development* 20.1 (1976), pp. 40–57.
- [9] L. H. Quam. "Road tracking and anomaly detection in aerial imagery". SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER, 1978.
- [10] R. J. Schalkoff. "Digital image processing and computer vision". Vol. 286. New York: Wiley, 1989.
- [11] M. Fleury and F. Reverbel. "The JBoss extensible server" in *Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware*, 2003, pp.344–373.
- [12] "MATLAB The Language of Technical Computing" Internet: <http://www.mathworks.com/products/matlab/features.html>, [Mar. 3, 2015].
- [13] "Image Processing Toolbox" Internet: <http://www.mathworks.com/products/image/index-b.html>, [Mar. 5, 2015].
- [14] "Open Source Computer Vision" Internet: <http://opencv.org/>, [Mar. 10, 2015].

- [15] "The OpenCV Library" Internet: <http://www.drdobbs.com/open-source/the-opencv-library/184404319>, [Mar. 10, 2015].
- [16] C.A. Schneider, W.S. Rasband and K.W. Eliceiri. "NIH Image to ImageJ: 25 years of image analysis", *Nature methods*, vol. 9(7), 2012, pp.671–675.
- [17] I. Sommerville. "Ingeniería del software". Pearson Educación, 2005, pp. 61–66.
- [18] A. Cockburn. "Using Both Incremental and Iterative Development". STSC CrossTalk (USAF Software Technology Support Center), 2008, 21(5), pp. 27–30.
- [19] R. Macefield. "How to specify the participant group size for usability studies: a practitioner's guide" *Journal of Usability Studies*, 2009, pp. 34–45.
- [20] E. Allen and A. Seaman. "Likert scales and data analyses", *Quality Progress*, vol. 40(7), 2007, pp. 64–65.
- [21] "Seam Framework - JBoss Seam" Internet: <http://seamframework.org/>, [Mar. 13, 2015].
- [22] M. Pagolu and G. Chakraborty. "Eliminating response style segments in survey data via double standardization before clustering", 2011, *SAS Global Forum 2011 Proceedings*.
- [23] R. Fischer and T. Milfont. "Standardization in psychological research", *International Journal of Psychological Research*, vol. 3(1), 2010, pp. 88–96.