

LibreScan: Software Libre para la digitalización de documentos utilizando escáneres de bajo costo

Jaime Gutiérrez Alfaro¹, Aurelio Sanabria Rodríguez¹, Diego Ugalde Ávila¹,
Daniel Solís Méndez¹, Melvin Elizondo Pérez¹, Tony Kong Hsu¹

¹ Unidad de Ingeniería en Computación – Centro Académico de Alajuela
Instituto Tecnológico de Costa Rica (ITCR)
Alajuela, Costa Rica

{jgutierrez, ausanabria}@itcr.ac.cr

{dua.0695, daniel.soliscr, melvin.aep, tonyzeru}@gmail.com

Abstract. *Computer Engineering program of the Costa Rica Institute of Technology at the Academic Center in Alajuela was open in 2012 with a teaching only approach. Mainly because of the accreditation process, by mid 2013 the Laboratorio Experimental (LabExp) was created as an initiative where students could get involved in Free Software and Hardware research projects. The DIY Book Scanner community has been sharing the design of low cost scanners. Free Information and Technology project at LabExp has developed LibreScan, a Free Software solution for digitizing documents with a DIY Book Scanner. This article presents the software as well as how it started being used in our University.*

1. Metadata

Version described in the paper: 1.0.0

License: GPLv3

Link to source code repository: <https://github.com/LabExperimental-SIUA/LibreScan>

Link to project website: <http://labexperimental-siua.github.io/LibreScan/>

Link to communication channel: <https://telegram.me/joinchat/AOFjZwZtkrOPQvvppFvWDw>

Programming languages utilized: Python, Javascript

Dependencies: python 3.4, tesseract-ocr 3.03.03, scantailor 0.9.11, ruby 2.1.5, ruby-dev 2.1.5, ruby-rmagick 2.13.2, pdfbeads 1.1.1, iconv 1.0.4, chdkptp alpha, libjpeg8 8d1-2, libjpeg8-dev 8d-1, libssl-dev 1.0.1, libffi-dev 3.1, libturbojpeg1 1.3.1, cffi 1.5.0, i18n 0.2, jpegtran-cffi 0.5.2, pyjade 3.1.0, pexpect 4.0.1, jinja2 2.8

Operating systems compatible: Debian 8 Jessie.

List of contributors: Melvin Elizondo Pérez, Jaime Gutiérrez Alfaro, Tony Kong Hsu, Aurelio Sanabria, Daniel Solís Méndez, Diego Ugalde Ávila.

2. Introducción

El Instituto Tecnológico de Costa Rica es la universidad pública que centra su quehacer en la Ciencia y Tecnología. En el año 2012 comenzó a ofrecer la carrera de Ing. en

Computación en el Centro Académico de Alajuela, ubicado en la Sede Interuniversitaria (SIUA). Si bien en el inicio la actividad principal era impartir lecciones, se hizo necesario ofrecer espacios para la investigación y extensión, principalmente debido a que la carrera decidió iniciar un proceso de acreditación con el Sistema Nacional de Acreditación para la Educación Superior¹ (SINAES).

El Laboratorio Experimental de Tecnologías Computacionales (LabExp) fue creado a finales del año 2013, permitiendo atender esa necesidad académica complementaria a la docencia requerida para el proceso de acreditación, y para ofrecer a los estudiantes de la carrera un espacio para involucrarse en proyectos de Software y Hardware Libre. La participación en estos proyectos le ofrece a estudiantes oportunidades valiosas para aplicar sus conocimientos técnicos, aprender sobre herramientas de desarrollo que no están incluidas en sus estudios formales y desarrollar destrezas blandas ("*soft skills*") [Nathan 2016].

Información Libre y Tecnología (ILT) es el nombre del primer proyecto que se planteó desde el LabExp, luego de establecer un vínculo con la comunidad *DIY Book Scanner*², quienes crean escáneres de bajo costo para la digitalización de documentos. Daniel Reetz, su fundador, donó un escáner al proyecto ILT para trabajar en el desarrollo de un Software Libre que permitiera facilitar el proceso de digitalización. Se reclutaron los primeros estudiantes del Centro Académico de Alajuela y se comenzó el proceso de desarrollo desde una perspectiva académica de investigación a nivel de pregrado.

Partiendo desde el reto técnico de proveer un Software Libre para la digitalización de documentos, uno de los principios del proyecto es contribuir a fomentar el acceso libre a la información y la cultura. En muchos casos proveer versiones digitales facilita el acceso a textos que físicamente son inaccesibles, por ejemplo cuando están ubicados en bibliotecas distantes o son documentos difíciles de conseguir. El acceso al texto en digital mejora las condiciones educativas, y en términos generales, contribuye a preservar el conocimiento como patrimonio colectivo de la humanidad.

Adicionalmente, aplicando algunas técnicas de reconocimiento de texto (Optical Character Recognition, OCR por sus siglas en inglés) es posible contribuir con el proceso educativo de personas con diferentes discapacidades visuales, cognitivas o físicas. Por ejemplo quienes tienen dificultades con la motora fina para manipular un libro y pasar las páginas [Ishihara et al. 2012], o personas con discapacidades visuales que se ven beneficiadas por la reproducción de audio de los textos, permitiéndoles acceder al conocimiento contenido en ellos.

A partir de la necesidad de la carrera de Ingeniería en Computación de generar espacios para trabajar en procesos de investigación y extensión con estudiantes, el proyecto toma la decisión de desarrollar *LibreScan*. Este software tiene como objetivo convertirse en una opción libre y gratuita para la digitalización de textos con escáneres de bajo costo. A su vez, permite a los estudiantes involucrarse con comunidades de Software, Hardware y Cultura Libre.

La siguiente sección de este artículo justifica la selección de un modelo de desarrollo de Software Libre con estudiantes de pregrado y ofrece generalidades sobre los

¹www.sinaes.ac.cr

²<http://www.diybookscanner.org/>

escáneres de la comunidad DIY *Book Scanner*. La cuarta sección describe el proceso de desarrollo y las características más relevantes de *LibreScan* como alternativa de Software Libre para escaneo. Posteriormente se comenta la experiencia de poner a disposición el escáner y *LibreScan* a la comunidad universitaria, comparándola como una alternativa a otras soluciones privativas utilizadas en la actualidad en el Instituto Tecnológico de Costa Rica. La quinta sección ofrece una discusión de los resultados obtenidos utilizando el software en la Institución. Finalmente se ofrece una sección que explica la instalación del software, una de conclusiones y trabajo futuro.



Figura 1. Profesores y estudiantes trabajando en el proyecto ILT del LabExp

3. Contexto

3.1. FLOSS en educación

En el proceso de formación en Ingeniería en Computación del Tecnológico de Costa Rica, existen varios tipos de proyectos prácticos que pueden ejecutar los estudiantes de pregrado para complementar los conocimientos teóricos del área de Ingeniería de Software, los cuales contribuyen al mejoramiento de las habilidades técnicas y de gestión de proyectos. Los más usuales son asignados por un profesor como parte de un curso, pero también es posible desarrollar experiencias prácticas en organizaciones o empresas. Estos dos tipos traen consigo restricciones, de tiempo y alcance de los proyectos en el primer caso, mientras que en el segundo caso el impacto del trabajo desarrollado es limitado a la empresa.

Como una alternativa, desde el Laboratorio Experimental y específicamente por medio del proyecto Información Libre y Tecnología (ILT), se ha ofrecido un espacio a estudiantes para desarrollar proyectos basados en modelos FLOSS (*Free/Libre Open Source Software*), los cuales tienen ventajas adicionales a las ya mencionadas. Según [Ellis et al. 2012], el hecho de que el código sea abierto permite que el trabajo pueda ser controlado por el docente, evitar problemas de licenciamiento y propiedad intelectual. Además brinda a los estudiantes la oportunidad de interactuar con colaboradores de distintas regiones y perfiles, lo que fomenta el trabajo en equipo y mejora las habilidades interpersonales.

Desde la perspectiva de las Universidades públicas es importante el desarrollo de soluciones de naturaleza altruista, siendo los proyectos FLOSS una buena opción, puesto que el trabajo realizado queda a disposición de la sociedad. Esta característica llama la atención de estudiantes, ya que por lo general los trabajos asignados en los cursos no trascienden a usos reales.

Se han realizado encuestas para recolectar información y opiniones con respecto a las experiencias de estudiantes en proyectos FLOSS. En [Hislop and Ellis 2015] la mayoría de encuestados aseguraron su crecimiento profesional en el desarrollo de proyectos de ingeniería de software y percibieron la experiencia como positiva y enriquecedora. Además se menciona que la mayoría tuvo interacción con la comunidad involucrada en el proyecto.

A pesar de los beneficios mencionados, existen distintas barreras al involucrarse en este tipo de proyectos. Según Steinmacher y Gerosa [Steinmacher and Gerosa 2015] una de las principales barreras es el código mal estructurado y poco comprensible, ya que este incrementa la curva de aprendizaje y dificulta la adaptación al proyecto, lo que desmotiva a las personas a contribuir con algo difícil de entender. Otras de las barreras que se mencionan son: la dificultad para encontrar una tarea inicial con baja complejidad para una persona que recién se incorpora en el proyecto, conocimiento o dificultad de las herramientas usadas, falta de información y documentación del proyecto, ineficacia para responder preguntas de los nuevos integrantes, entre otros.

En el trabajo de Steinmacher y Gerosa [Steinmacher and Gerosa 2015] también se describe que en el involucramiento de personas en proyectos FLOSS se presentan dos etapas principales que son el abordaje y la contribución. El abordaje es la fase donde se deben presentar las mayores facilidades, para llamar la atención de los interesados y de esta forma acercarlos a contribuir activamente en el proyecto. Una vez que inicien su participación en el proyecto se debe hacer sentir a las personas que son parte del equipo y mantener activo el factor de motivación para que los colaboradores sigan brindando su ayuda, como puede ser otorgar más permisos sobre el repositorio o privilegios de administración.

En cuanto a recomendaciones para tomar en cuenta en proyectos FLOSS están: mantener canales de comunicación locales y activos, guiar a las personas nuevas hacia tareas sencillas, crear una sección de preguntas frecuentes y mantener el código tan simple como sea posible, lo que implica una documentación completa [Steinmacher and Gerosa 2015].

3.2. Escáneres de bajo costo

Los escáneres poseen diferentes características que definen su costo y facilidad de obtención. Una categorización realizada por Daniel Reetz [Reetz 2010], establece que la mayoría pertenece a alguno de los siguientes tipos³:

- Cama plana: este tipo de escáner es de los más comunes y se caracteriza por su bajo costo. Los resultados al utilizarlos proveen una excelente calidad en las imágenes. Sin embargo, son lentos y son menos apropiados para el proceso de escaneo. Un escáner comercial de este tipo ronda entre los \$40 y \$250.
- Alimentado por hojas: este dispositivo toma las hojas una por una automáticamente, no obstante, requiere la destrucción del libro. El precio de venta ronda entre \$300 y \$500.
- Cara arriba completamente automatizado: esta clase de escáner trata más cuidadosamente los libros. El documento es colocado con el contenido cara arriba,

³Los precios son en dólares de EE.UU. y corresponden al 2010 cuando Daniel Reetz elaboró la categorización.

usualmente utilizando una plataforma con forma de V. Se utilizan dos cámaras que capturan las páginas del libro y la velocidad del escaneo esta prácticamente limitada al mecanismo que voltee las páginas. Un escáner de este tipo se considera profesional o especializado y su precio ronda entre los \$7.000 y \$150.000.

Si bien existen escáneres comerciales que se pueden adquirir con relativa facilidad, no suelen ser la mejor solución para una institución que no cuente con los recursos necesarios para digitalizar documentos con características específicas, por ejemplo un libro cuyas dimensiones no puedan ser abarcadas por un escáner económico de cama plana. Por otra parte, una persona que quiera digitalizar documentos personales se vería en la necesidad de hacer una gran inversión para obtener un escáner profesional, o bien, recurrir a un escáner para el hogar, como los que ahora se encuentran en muchas impresoras multifuncionales; con la desventaja de que estos suelen ser muy lentos.

La comunidad *DIY Book Scanner* ha trabajado en buscar una solución más asequible para digitalizar documentos. En la página principal de la comunidad⁴ se pueden observar las distintas estructuras de tipo cara arriba propuestas hasta la fecha. Cabe mencionar que *DIY Book Scanner* mantiene una ideología de Hardware libre, lo que significa que sus diseños son abiertos y cualquier persona puede acceder a ellos y construir una replica si así lo desea. Por este motivo, muchas personas alrededor del mundo se han animado a crear sus propios diseños. Los costos en este tipo de estructuras, como se menciona en [da Cunha Penteado and Amiel 2015], se pueden reducir según la facilidad de acceso a distintos materiales de bajo costo.

Aparte de procurar mantener diseño de bajo costo, la iniciativa *DIY Book Scanner* busca crear un escáner que proporcione un ambiente ideal para las fotografías de los documentos. Aquí entran en juego diferentes factores que afectan significativamente el resultado final y la calidad de las capturas. Entre ellos se encuentran condiciones lumínicas, mecanismos de captura, volteo de páginas y la adaptabilidad de la estructura ante diversas dimensiones de documentos. A continuación se presentan tres de los seis diseños trabajados por la comunidad⁵:

- ***Instructables Scanner (2009)***: este modelo dio inicio a la comunidad cuando Daniel Reetz creó una guía de cómo construir un escáner por aproximadamente \$300.
- ***Hackerspace Scanner (2012)***: este presenta un diseño más elaborado con respecto a versiones anteriores. Su estructura se puede apreciar en la Figura 2 y tiene un precio aproximado a los \$600.
- ***Archivist Quill Book Scanner (2016)***: este escáner es el modelo más reciente propuesto por la comunidad. En su página presentan dos kits para venta, el kit básico tienen un valor de \$1.100 y el kit completo tiene un valor de \$1.700. La información completa del contenido de los kits se puede consultar en la página principal de *DIY Book Scanner*.

El LabExp trabaja con el modelo *Hackerspace Scanner* y con dos cámaras marca Canon A2300. El manual de ensamblaje de la estructura está disponible en la página de

⁴<http://www.diybookscanner.org/designs.html>

⁵El valor de cada escáner es indicado como referencia y responde al precio por el cual se ofrecían en el sitio web.

la comunidad⁶. Es importante recalcar que el modelo de las cámaras puede repercutir en los resultados. En este caso se decidió utilizar cámaras Canon porque muchos de sus modelos permiten hacer uso del *firmware Canon Hack Development Kit* (CHDK), el cual da la posibilidad de controlar el dispositivo de manera remota. Esto es necesario para poder capturar fotos desde la computadora, almacenarlas directamente en disco y modificar valores de configuración como el enfoque y acercamiento.

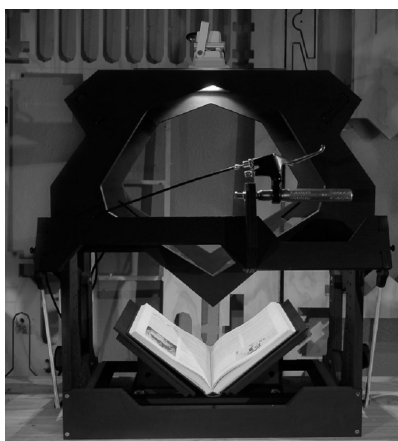


Figura 2. Modelo de DIY Book Scanner utilizado en el proyecto ILT

4. Acerca de LibreScan

4.1. Flujo de digitalización

Antes de iniciar con el trabajo en el software, fue fundamental entender a fondo todo el proceso de digitalización, incluyendo los pasos y herramientas involucradas. Para esto se realizó una reunión con Evelin Heidel (comunidad DIY Book Scanner) donde nos explicó la secuencia de tareas necesarias para obtener una versión digital de un texto realizando el procedimiento de forma manual y combinando varias herramientas de software. Como primer paso, el usuario debe proceder a tomar las fotos de las páginas del documento a digitalizar. Seguidamente se procesan las imágenes, para esto se necesita primero editarlas (recorte, limpieza, rotación) con *Scantailor* y producir archivos *tif*.

Una vez procesadas las imágenes se continúa con el reconocimiento de texto por medio del software *Tesseract-OCR*. Este último se encarga de generar archivos *hocr* con el texto reconocido, que servirán de entrada junto con los *tif* para ensamblar el producto final, en este caso un PDF generado con *PDFBeds*. En la Figura 3 se puede visualizar esta secuencia de pasos.

Durante las pruebas realizadas para familiarizarse con el flujo de trabajo se tuvo que lidiar con factores que afectan los resultados (como los mencionados en la sección anterior). Se obtuvieron documentos digitales de baja calidad debido a reflejos ocasionados por luz externa. Ante esto se intentó modificar el escáner añadiéndole una manta que disminuyera los reflejos en el área de captura de fotos y se probó mover la estructura a un lugar más oscuro con el objetivo de que sólo se contara con la luz artificial de la lámpara. En la Figura 4 se puede apreciar un texto con manchas blancas y algunas secciones más claras que otras.

⁶<http://diybookscanner.org/assemblymanual.pdf>

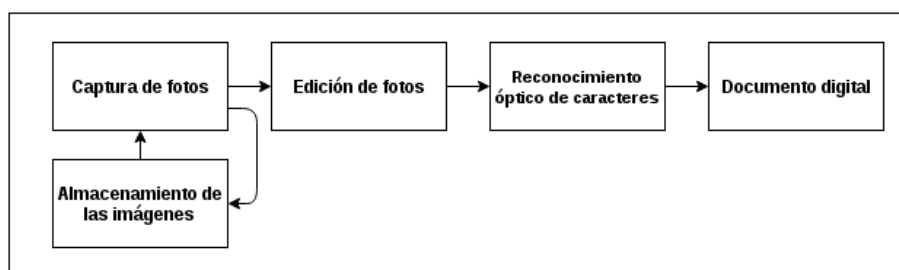


Figura 3. Diagrama de proceso de escaneo.

Máquinas que aprenden

El lector habrá creído que no tengo ningún argumento muy convincente de carácter positivo para apoyar mis puntos de vista. Si lo tuviera, no me habría esforzado tanto por señalar las falacias en las opiniones opuestas. Ahora daré la prueba de que dispongo.

Volvamos por un momento a la objeción de Lady Lovelace, quien declaró que una máquina solamente puede hacer lo que le decimos que haga. Se podría afirmar que un

Figura 4. Mancha en el texto digitalizado por reflejos de luz

4.2. Desarrollo del software

Parte de la experiencia de involucrar estudiantes a proyectos FLOSS fue el trabajar con un proyecto existente, Spreads. Este es un software que inició su desarrollo a mediados del 2013 y tiene el objetivo de facilitar el flujo de escaneo, al funcionar como un contenedor de herramientas que llevan a cabo las distintas fases de la digitalización. Sin embargo, se experimentaron algunas de las barreras mencionadas en el contexto, como la alta complejidad de un proyecto grande y avanzado, lo que incrementó la curva de aprendizaje y dificultad de adaptación de estudiantes de pregrado con poca experiencia y que iniciaban la carrera.

Es por esto que se consideró más apropiado separarse e iniciar un software desde cero, ya que se visualizó como un buen ejercicio académico complementario a distintos cursos de la carrera como Diseño de Software, Programación Orientada a Objetos, Desarrollo Web y Sistemas Operativos. No obstante, muchas de las herramientas utilizadas fueron escogidas tomando Spreads como base, lo que afirma la importancia de haber pasado por esta fase y experimentar un abordaje a un proyecto con esas características.

Por la situación descrita, se decidió iniciar el desarrollo de LibreScan, una herramienta de Software Libre que permite automatizar el proceso necesario para digitalizar un documento utilizando un escáner de bajo costo. Está disponible en un repositorio de GitHub para fomentar el trabajo colaborativo. El proyecto está abierto a personas que deseen involucrarse, por lo que se ha creado una página en donde se puede encontrar más información sobre el repositorio y el canal de comunicación en Telegram.

El código se desarrolló siguiendo una arquitectura de capas y principios de diseño de software aprendidos durante la carrera, lo que facilita la modificación, mantenimiento y adición de nuevos componentes. La versión que está funcionando actualmente cuenta con una interfaz web desarrollada con HTML5, CSS y Javascript. Los principales com-

ponentes de LibreScan se presentan en la Figura 5.

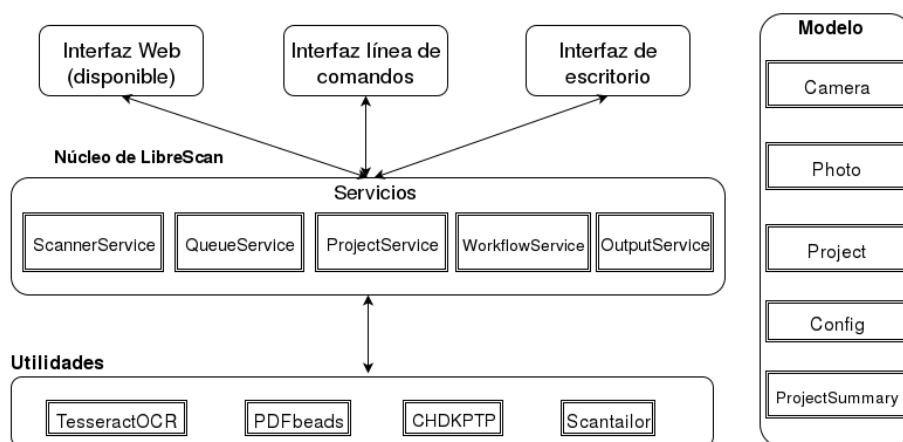


Figura 5. Arquitectura de software de LibreScan

Dentro de las características de LibreScan se encuentra la buena experiencia de usuario que ofrece, ya que posee una interfaz minimalista que se centra en pocos elementos y figuras entendibles, para de esta manera ser intuitiva al usuario. Además, oculta detalles utilizados con poca frecuencia, como configuración avanzada de la aplicación, cámaras y mensajes de error entendibles solo por desarrolladores. Tanto la interfaz de usuario como el reconocimiento de texto soportan múltiples idiomas; actualmente español, inglés y portugués.

A pesar de que el proceso de escaneo es muy extenso, LibreScan reduce la espera del usuario. El software se encarga de procesar imágenes mientras se capturan nuevas páginas, de esta forma, cuando se termina la fase de captura ya se habrá llevado a cabo la edición y reconocimiento de texto de buena parte de las imágenes.

Ya que el proceso de digitalizar documentos está propenso a errores, ya sea que se apague una cámara, se descargue la máquina o se presente un fallo eléctrico; LibreScan tiene la posibilidad de realizar el trabajo en varias etapas, es decir, se permite suspender el trabajo y reanudarlo después. Es por esto que LibreScan brinda al usuario una aplicación confiable y robusta, al tener la capacidad de recuperarse ante fallos y mantener la consistencia de los proyectos.

En seguida se muestran imágenes de la interfaz gráfica de LibreScan y un resultado de escaneo. En la Figura 6 se puede apreciar la pantalla que permite crear un nuevo proyecto, donde se despliega un pequeño formulario al usuario para introducir la información básica del documento por escanear. Por otro lado, la Figura 7 exhibe la pantalla donde se realizan las capturas de las imágenes. Por último, en la Figura 8 se visualiza un ejemplo de una página escaneada utilizando LibreScan.

4.3. Detalles de implementación

Una de las primeras decisiones tomadas fue la de utilizar el paradigma programación orientada a objetos, ya que el modelamiento de los elementos involucrados en el proyecto resultó natural y se facilitó la aplicación de patrones de diseño como Singleton y Command.



Figura 6. Pantalla para crear un nuevo proyecto

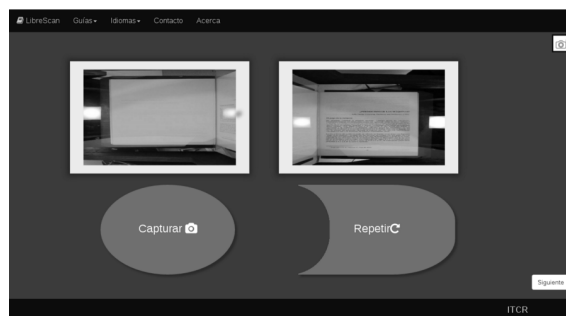


Figura 7. Pantalla de escaneo

La escogencia de una interfaz web se debe a que puede ser utilizada en múltiples plataformas y prevee el uso del software en distintos tipos de dispositivos. Para la generación de HTML5 se decidió emplear el lenguaje de plantillas Jade, el cual disminuye notablemente la cantidad de líneas de código, al abreviar las etiquetas y atributos que posee HTML5.

En cuanto a los lenguajes de programación, para el lado del servidor se utilizó Python pues fue uno de los primeros lenguajes aprendidos por los estudiantes dentro de la carrera, y por ser de muy alto nivel, se logra mantener el código simple y desarrollar aplicaciones de manera rápida. Además, previo al desarrollo de LibreScan se trabajó con Spreads (implementado en Python), lo que propició un desarrollo ágil al conocer aspectos de integración con las herramientas incluidas en el flujo de digitalización. Por otra parte el lenguaje para el lado del cliente es Javascript, dado que es el más conocido para el desarrollo de aplicaciones web y posibilita la carga de contenido dinámico con Ajax.

Otro aspecto a destacar es que algunas tareas realizadas por el software son independientes, lo que permitió paralelizarlas con hilos, y así, sacarle el máximo provecho al procesador. De esta manera se obtuvieron mejores resultados en rendimiento y el tiempo de procesamiento de las imágenes se redujo de manera considerable.

REPRESENTACIÓN VIRTUAL DEL CONOCIMIENTO

Verónica Dahl, *Computer* (1983)

INTRODUCCIÓN

La idea de que la lógica pueda servir como un lenguaje de programación fue puesta en práctica en 1972 en la forma de Prolog. Ha sido muy valiosa en muchas áreas de computación, incluyendo la comprensión del lenguaje ordinario, las bases de datos deductivas, la robótica, la integración simbólica y los sistemas expertos.

La lógica tradicionalmente ha proveído un marco conceptual firme para la representación del conocimiento.... La introducción de Prolog ha hecho posible representar conocimiento en términos lógicos en tal forma que las inferencias apropiadas puedan obtenerse de manera automática.

En Prolog el conocimiento se expresa de manera sencilla en términos de *hechos* y de *reglas* para computar otros hechos; y cualesquiera consecuencias lógicas de ese conocimiento están a la disposición del usuario, no teniendo este más que pedirlos. Estas consecuencias pueden entenderse como conocimiento virtual, no explícitamente presente pero deducible. Las deducciones necesarias para esto son realizadas por Prolog en forma transparente para el usuario....

133

Figura 8. Resultado de escaneo del libro Epistemología e Informática para el curso de Inteligencia Artificial

5. Uso de LibreScan

El proyecto ILT, además de enfocarse en la formación y crecimiento profesional de los estudiantes involucrados, busca mejorar el acceso a recursos educativos y disminuir las desigualdades generadas por la restricción o dificultad para obtener materiales didácticos. Por esto, desde los primeros meses de trabajo en el proyecto, se analizó cuáles serían potenciales lugares para poner a disposición el escáner y LibreScan. Se consideró conveniente proporcionar las herramientas a bibliotecas de centros educativos, puesto que son espacios donde se trabaja constantemente con documentos y se tiene la necesidad de digitalizarlos. En este sentido se escogieron la biblioteca de la SIUA y la biblioteca de la sede de Cartago del Instituto Tecnológico de Costa Rica como los primeros espacios para utilizar el escáner. Ambas ya cuentan con soluciones privativas al mismo problema, por lo que la experiencia serviría de insumo importante.

En el caso de la SIUA se contactó a la bibliotecaria Anabelly Tinoco, quien comentó que en el 2013 se matricularon en la Sede 3 estudiantes con discapacidades visuales. Como un servicio de la biblioteca se crearon y pusieron a disposición versiones audibles de los textos necesarios para los cursos. Para atender esta necesidad se procedió a adquirir un escáner HP ScanJet 5590 (de cama plana) y el software propietario Abbyy Fine Reader⁷, el cual ofrece la facilidad de escanear texto y generar un archivo de Microsoft Word editable. El fin de esto es posteriormente poder escuchar los textos a partir de otro software llamado JAWS⁸ encargado de realizar la lectura de la pantalla. Comúnmente son estudiantes asistentes los que se encargan de utilizar estas herramientas de digitalización y la bibliotecaria estimó que tardan aproximadamente 8 días en escanear

⁷<http://www.abbyy.com/finereader/>

⁸<http://www.freedomscientific.com/JAWSHQ/JAWSquarters01>

50 páginas y otros 8 días en depurar el texto manualmente. Este proceso de depuración consiste en tomar el texto que produce el software Abbyy Fine Reader y corregir todas las palabras que no fueron identificadas correctamente.

Por otro lado se realizó una reunión con el encargado de escanear y administrar el equipo de digitalización para la sede de Cartago, Eliécer Abarca, quien mostró el espacio donde se tiene todo el equipo adquirido hasta la fecha. En esta biblioteca se cuenta con hardware más sofisticado, pues se han adquirido dos escáneres Kodak complementarios, el Kodak i3450 (alimentado por hojas) y el Kodak A3 (cama plana). Estos dispositivos trabajan en conjunto con el software OmniPage⁹ para pasar de PDF a texto y poder editar el contenido reconocido. Les toma en promedio una semana y media escanear de 150 a 200 páginas.

Actualmente se encuentra matriculado en la carrera de Ingeniería en Computación un estudiante con discapacidad visual total (no vidente). Al igual que en Alajuela, se le ofrece un servicio donde se escanean los textos de los cursos para poder escucharlos a través de un software. Los textos audibles son guardados para poder ser usados en la universidad o ser compartidos con otras personas que requieran accederlos. Lo anterior refleja una necesidad común que puede ser solventada mediante un escáner de bajo costo y LibreScan.

En ambos casos, si bien se ha logrado atender con éxito la necesidad de equiparar las condiciones educativas de los estudiantes con discapacidades visuales, las soluciones han implicado utilizar tecnología privativa (hardware y software). Además desde el punto de vista económico se ha requerido una inversión considerable por parte de la universidad (ver Tabla 1).

Para los propósitos descritos, se ha puesto a disposición de la biblioteca de la SIUA el escáner donado. Se realizó la instalación del equipo y se dio una capacitación básica para su uso. Estudiantes y la bibliotecaria realizaron distintos escaneos durante un mes, con el objetivo de obtener versiones audibles de los libros. Posteriormente se contactó de nuevo a la bibliotecaria para obtener realimentación sobre el uso dado a LibreScan, centrando las preguntas en beneficios, limitaciones y recomendaciones para la solución ofrecida por el LabExp. La experiencia de poner a disposición de la biblioteca de la SIUA el escáner donado por la comunidad *DIY Book Scanner* y el software LibreScan ha dejado experiencias importantes que se discuten en la siguiente sección.

6. Discusión

En términos generales, la bibliotecaria de la SIUA se mostró satisfecha con las pruebas realizadas y cree que es una mejor solución que la opción privativa. Comentó que utilizar LibreScan y un escáner de bajo costo le permitió digitalizar más rápidamente, ya que en comparación al escáner comercial, se capturan dos páginas a la vez y los disparos de las cámaras son más veloces que el escaneo por luz. También se reduce considerablemente el proceso de depuración de texto, gracias a que el *Tesseract* utilizado en LibreScan se encarga del reconocimiento de la mayoría del texto y los errores a corregir son considerablemente menores.

Respecto a la interfaz de usuario, comentó que le resultó sumamente intuitiva

⁹<http://www.nuance.com/for-individuals/by-product/omnipage/index.htm>

Tabla 1. Comparación de costos por soluciones de digitalización

Modelo	Hardware	Software	Total
Hackerspace Scanner (2012) / LibreScan	\$468	\$0	\$468
Piezas de madera	\$200		
Otros materiales	\$75		
2 Cámaras Cannon A2200	\$193		
LibreScan		\$0	
Archivist Quill Book Scanner (Base Kit) / LibreScan	\$1.293	\$0	\$1.293
Estructura del Escáner			
2 Cámaras Cannon A2200	\$193		
LibreScan		\$0	
Archivist Quill Book Scanner (Complete Kit) / Pi Scan	\$1.700	\$0	\$1.700
Estructura del Escáner, Cámaras y otros materiales	\$1.700	\$0	
PiScan		\$0	
LibreScan		\$0	
Escáner HP ScanJet 5590 / Abby Fine Reader - SIUA	\$351	\$169,99	\$409,99
HP Scanjet 5590	\$240		
Abby FineReader 12 Professional		\$169,99	
Escáneres Kodak / Omnipage - TEC-Cartago	\$10.186	\$499,99	\$10.685,99
Kodak i3450 Scanner	\$8.709		
Kodak A3 (Cama Plana)	\$1.477		
OmniPage Ultimate		\$499,99	

debido a que distintas personas la usaron correctamente sin la necesidad de un manual. Otra característica llamativa para la entrevistada fue el autoguardado del software, lo que permite trabajar los proyectos por partes sin perder avance. Además de los beneficios relacionados a las funcionalidades del programa, dejó claro que el hecho de que sea Software Libre y gratuito es un punto más a favor, al permitir reducir costos y potencialmente adquirir mayor cantidad de escáneres.

En cuanto a las críticas constructivas, mencionó que cuando el escaneo es realizado por una sola persona puede resultar incómodo el proceso, al requerir pasar páginas y presionar botones en la computadora al mismo tiempo. Otro comentario que realiza es que al revisar los productos finales de escaneos se nota que la calidad final no siempre es la ideal, ya que la fuente de la letra tiende a quedar muy gruesa en algunas ocasiones. Además, cuando se escanean libros con imágenes y gráficos no se realizan bien los recortes, en comparación con los libros que contienen únicamente texto. Por último se recomendó cambiar colores en la interfaz de usuario, agregar un módulo de corrección ortográfica y extender el formulario de creación de documentos, debido a que hace falta información valiosa como el autor y la editorial.

Desde el punto de vista económico, la Tabla 1 sintetiza los costos de cada solución al problema de digitalizar y editar el texto obtenido. Si bien varían en cuanto a las facilidades para lograr este propósito, en general el proceso se consigue en dos pasos: el primero es el escaneo y reconocimiento de caracteres, y el segundo es la edición en una herramienta de ofimática. Los montos se indican en dólares de los Estados Unidos de

América. Se utilizó el sitio de compras por Internet Amazon.com¹⁰ para buscar el hardware (cámaras fotográficas y escáneres), por lo que posiblemente el precio sea mayor al incluir el envío hasta el sitio donde se quiera implementar la solución. El costo del software privativo es el que se indica en su respectivo sitio web. Los precios indicados corresponden al 30 de abril del 2016.

Se deben realizar algunas consideraciones importantes al analizar la tabla. En todas las soluciones comparadas se asume que se cuenta con una computadora de propósito general. En las soluciones privativas, para tener el precio de la solución completa, se debería incorporar el pago por utilizar un Sistema Operativo y paquetes de Ofimática privativos que tienen un costo importante que no se incluye en esta estimación. En el caso del kit completo del escáner de bajo costo *Archivist Quill Book Scanner* se incluye un Software Libre para tomar fotografías (PiScan), pero este no lleva a cabo el procesamiento de las imágenes.

7. Instalación del software

LibreScan depende de una serie de programas y bibliotecas, algunos de ellas son brindados por el Sistema Operativo (Debian 8 Jessie) y otros son módulos del lenguaje de programación Python3 y Ruby. Para la instalación de las dependencias se utilizan los administradores de paquetes: “apt” (paquetes Debian), “pip3” (paquetes Python3) y “gem” (Ruby). La lista de repositorios recomendados se puede ver en el archivo “apt.sources”¹¹ que se encuentra en el GitHub de LibreScan. A continuación se muestran los comandos para instalar las dependencias¹²:

```
# apt-get install python3-pip lua5.2 liblua5.2 git-svn libusb-dev
python3 python-dev libffi-dev libturbojpeg-dev libssl-dev libjpeg8
-dev libjpeg8
# pip3 install pyYAML bottle pyjade jinja2 polib cffi pexpect
# pip3 install jpegtran-cffi==0.5.2
```

Además se deben instalar las herramientas que ayudan a realizar el flujo de escaneo en LibreScan. Estas son Tesseract OCR, ScanTailor y PDFBeads. Para su instalación se ejecutan las líneas:

```
# apt-get install tesseract-ocr scantailor ruby ruby-dev ruby-rmagick
# gem install iconv pdfbeads
```

LibreScan utiliza una herramienta adicional llamada CHDKPTP, encargada de administrar las cámaras. Esta no cuenta con un paquete de instalación, por lo tanto se deben seguir una serie de pasos que no serán descritos en detalle ya que CHDKPTP es tomado como un software externo a LibreScan. En la guía que se encuentra en el repositorio de GitHub se detalla la instalación de esta herramienta.

Adicionalmente, CHDKPTP se comunica con las cámaras por medio del firmware CHDK que debe ser instalado en las cámaras; para lograr esto existen diferentes métodos disponibles que se pueden utilizar. El LabExp ha desarrollado una guía que podría ser de ayuda para realizar este proceso, dicha guía se encuentra en la Wiki de ILT del LabExp¹³.

¹⁰<http://www.amazon.com>

¹¹<https://github.com/LabExperimental-SIUA/LibreScan/blob/master/apt.sources>

¹²Es requisito tener privilegios de administrador del sistema

¹³<https://github.com/LabExperimental-SIUA/ilt/wiki/Instalaci%C3%B3n-de-CHDK>

Como un paso adicional para la configuración de las cámaras que se van a utilizar, es necesario identificar si la cámara está a la derecha o la izquierda. Para evitar que el usuario tenga que hacer esto cada vez que desea utilizar LibreScan, se debe crear en la raíz de la tarjeta SD de cada cámara un archivo llamado “orientation.txt”, este archivo debe contener la palabra “left” o “right” (sin comillas, espacios en blanco o saltos de línea) según la cámara que corresponda.

Finalmente para hacer uso de LibreScan se debe a clonar el repositorio (<https://github.com/LabExperimental-SIUA/LibreScan.git>) e ingresar en la carpeta del código fuente para ejecutar el archivo “setup.py” que se encarga de crear la estructura de directorios necesaria para el correcto funcionamiento de LibreScan. Una vez hecho esto se procede a ejecutar el archivo “main.py” con el argumento “web”. Este último paso ejecuta LibreScan y puede ser accedido desde un navegador web en la dirección <http://0.0.0.0:8180>. A continuación se presentan los comandos de las anteriores instrucciones:

```
$ python3 setup.py
$ python3 main.py web
```

8. Conclusiones y Trabajo Futuro

Los usos mencionados anteriormente son evidencia de una de las principales características de los proyectos FLOSS, el colaborar con comunidades. El hecho de que el trabajo realizado por estudiantes beneficie su institución y promueva la educación y la información libre resulta motivador e incrementa su compromiso con el proyecto.

En LibreScan se tomaron en cuenta las recomendaciones mencionadas en la sección de contexto, como establecer una comunicación activa con la comunidad de personas interesadas y procurar mantener el código limpio y entendible. Se creó un canal de comunicación en Telegram y una página de introducción para personas interesadas en utilizar o colaborar con el proyecto. Actualmente este medio de comunicación ha funcionado para mantener contacto con un usuario de Paraguay y uno de México.

El haber utilizado buenos principios de diseño y arquitectura de software, tiene como objetivo hacerle frente a la barrera de integrar colaboradores, al reducir la curva de aprendizaje y su adaptación al proyecto. Se considera fundamental tener una aplicación internacionalizada, la cual soporte distintos idiomas; es por esto que LibreScan fue diseñado de forma que se pueden añadir traducciones fácilmente.

Contar con buena documentación es importante cuando se trabaja en proyectos FLOSS, ya que facilita la incorporación de nuevos desarrolladores y también el proceso de instalación del software y sus dependencias, el cual se convierte muchas veces en un factor determinante en la utilización o descarte de un software.

Desde la perspectiva de los estudiantes que han participado del proyecto ILT y el proceso de desarrollo de LibreScan, se ofrecen las siguientes recomendaciones para facilitar la incorporación de otros estudiantes en proyecto FLOSS:

- Un conocimiento de software de control de versiones es deseable, dado que gran parte de los proyectos FLOSS evolucionan a través de estas herramientas, por lo que en algún punto se debe recurrir a su aprendizaje. En el caso del LabExp, se hace uso de Git para el manejo de versiones y GitHub para el hospedaje de los repositorios.

- Entre las cualidades a desarrollar están la paciencia y la persistencia, puesto que generalmente los procesos de aprendizaje, diseño e implementación suponen fallos, cambios y otros factores que repercuten en el progreso del proyecto. Por ejemplo el desconocimiento total de las tecnologías a utilizar, lo que implica una amplia investigación. Además, el surgimiento de errores puede llegar a obstaculizar el avance, es aquí donde es fundamental el trabajo en equipo entre estudiantes y profesores. Este contacto es necesario para que el proyecto evolucione de manera correcta, principalmente en la fase de abordaje donde los estudiantes aún están acoplándose.
- Se considera un punto a favor conocer la historia, visión y estado actual del proyecto; de esta forma se acelera el involucramiento y se da comienzo a la fase de contribución, de manera que los estudiantes puedan iniciar sus aportes en un corto tiempo.
- Una constante difusión del proyecto, por medio de exposiciones y presentaciones, colabora con el desarrollo de destrezas blandas, como lo es la interacción con personas externas al área de computación. Esto también permite dar a conocer el trabajo y aumentar la cantidad de personas interesadas en hacer uso o contribuir con el proyecto.

Respecto al LabExp, a pesar de ser la primer generación de estudiantes involucrados en proyectos de extensión, se considera exitoso pues logró el cometido de acercar e interesar estudiantes a este tipo de investigaciones vinculadas a proyectos FLOSS. Fue importante dar este primer paso para que el laboratorio siga creciendo y surjan más proyectos con miras a fortalecer el espacio y visibilizar los trabajos realizados por medio de publicaciones y actividades dentro y fuera de la universidad. Durante el periodo transcurrido se ganó experiencia y conocimiento relacionado a este tipo de proyectos, lo que será provechoso para los estudiantes próximos a integrarse, debido a que los siguientes procesos serán más organizados y podrán tomar en cuenta las recomendaciones y problemas experimentados en los inicios del LabExp.

En cuanto al trabajo futuro del proyecto, desde la perspectiva técnica, se plantea explorar los procesos relacionados con la identificación y corrección de errores introducidos por el reconocimiento de texto, cuya funcionalidad aún no ha sido integrada a LibreScan. Además, extender la utilidad del software con herramientas que faciliten el uso de los libros digitalizados a las personas con discapacidad, tales como la conversión de texto a voz (TTS por sus siglas en inglés).

También se planea mejorar la calidad de los escaneos de textos a color y documentos que posean gráficos y fórmulas matemáticas, ya que las pruebas evidencian que actualmente el software funciona de forma correcta solamente cuando el contenido es principalmente texto. Por último, migrar la aplicación a una Raspberry Pi reduciría los costos considerablemente, al remover la dependencia de una computadora para realizar los procesos; intentando convertir el escáner en un sistema embebido.

Otros aspectos, aún sin explorar, son la posibilidad de crear bibliotecas digitales o repositorios de acceso público similares al proyecto *Google Books* y eventualmente crear servicios de digitalización en bibliotecas públicas, lo cual implica hacer una revisión más detallada del marco legal existente en este tema, el cual varía según el país.

Durante el desarrollo del software y el proceso de escritura del presente artículo, se

escanearon textos que se requerían leer en cursos de la carrera. Muchas veces no se tiene disponibilidad de los libros en físico para cubrir todo el estudiantado. Esta necesidad motiva a la obtención de más escáneres de bajo costo que puedan ser instalados en la universidad para que sean usados libremente por personas que requieran generar versiones digitales y fáciles de distribuir.

Con la intención de aumentar el impacto en la comunidad, se está poniendo en marcha un proyecto en conjunto con el Liceo Otilio Ulate Blanco de Alajuela, donde el proyecto cobra un interés especial pues ellos también tienen estudiantes con discapacidades visuales. En el colegio se explicó ILT y en qué consiste un escáner de bajo costo para que los mismos alumnos construyan su propio escáner. Gracias a las características del hardware libre y los diseños abiertos, es posible trabajar de esta forma para que una vez terminada la estructura, utilicen LibreScan de forma gratuita y puedan contar con un dispositivo para digitalizar sus documentos a un bajo costo.

Esta labor colaborativa entre estudiantes universitarios y de otros centros educativos es una de las razones por las que son llamativos los proyectos FLOSS, pues deja en evidencia que brindando el apoyo y guía necesarios, los estudiantes son capaces de solventar sus propias necesidades y al mismo tiempo complementar su formación.

9. Agradecimientos

Los autores quieren agradecerle de forma muy especial a Daniel Reetz y Evelin Heidel por el invaluable apoyo y ayuda. También se dan las gracias a Anabelly Tinoco por la amabilidad mostrada durante el proceso de pruebas del software. A la coordinación del Centro Académico de Alajuela, la SIUA, la Carrera de Ingeniería en Computación en Alajuela y al Centro de Investigaciones en Computación por el apoyo económico.

References

- da Cunha Penteado, A. F. and Amiel, T. (2015). Construyendo um Scanner de Baixíssimo Custo. *XVI International Free Software Workshop*.
- Ellis, H. J. C., Hislop, G. W., Rodriguez, J. S., and Morelli, R. A. (2012). Student Software Engineering Learning via Participation in Humanitarian FOSS Projects. *ASEE Annual Conference and Exposition*.
- Hislop, G. W. and Ellis, H. J. C. (2015). Practical Experiences for IT Students in Humanitarian Free and Open Source Software Projects. *Proceedings of the 16th Annual Conference on Information Technology Education*, page 99.
- Ishihara, T., Itoko, T., Sato, D., Tzadok, A., and Takagi, H. (2012). Transforming Japanese Archives into Accessible Digital Books. *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 91–100.
- Nathan, W. (2016). The scarcity of college graduates with FOSS experience. *LWN.net*, pages 1–23.
- Reetz, D. (2010). Why in DIY Book Scanning, The. *NYL Sch. L. Rev.*, 55:251–269.
- Steinmacher, I. and Gerosa, M. A. (2015). Fostering Free/Libre Open Source Software community formation: guidelines for communities to support newcomers onboarding. *XVI International Free Software Workshop*.