

Scanner automático económico para libros

Juan Miguel Lombardi Suzzi
Bárbara Menares Aguilar
Daniel Cantarín

Descripción :

El proyecto consiste en la creación de un dispositivo para ser gestionado por computadora que, con mínima interacción humana, permita digitalizar libros completos, y sin que esto implique costos restrictivos de hardware y/o software, restricciones de patentamiento/licenciamiento en su construcción, ni captividad tecnológica de cara a los componentes que requiera.

Resumen previo a la construcción

Este documento detalla el proyecto anual del grupo de trabajo para la materia Laboratorio 3 de la cursada 2016. Como condición de aprobación, este proyecto debe cumplir con la premisa de aplicar diversos conocimientos adquiridos durante la carrera en diferentes asignaturas. Con este criterio, dentro del amplio espectro de desarrollos posibles, nosotros elegimos crear **un dispositivo automático para digitalizar libros originalmente impresos en papel.**

La motivación detrás de esta opción es hacer posible la **digitalización de libros de forma fácil, eficiente, y automática, en virtud de la longevidad de los textos y el acceso a la información y el conocimiento.** Existe mucho material que actualmente sólo consta en forma impresa y sin distribuciones editoriales vigentes (“fuera de catálogo”) que bien podría perdurarse y distribuirse en forma digital, no sólo facilitando así el acceso sino incluso abaratando los costos de distribución. Pero el proceso de digitalización manual es lento, tendiente a errores, y es común que genere material digital de escaso valor de cara a tecnologías actuales (dispositivos de lectura de libros digitales, bases de datos con motores de búsqueda de textos, etc).

Al mismo tiempo, existe un mercado de soluciones automatizadas para este tipo de trabajos, pero es un mercado con precios prohibitivos para el común de las personas (e incluso para muchas organizaciones dedicadas al tema). Entendemos que un dispositivo demasiado costoso tampoco es compatible con nuestras motivaciones, dado que no apuntamos a un público de grandes empresas sino de pequeñas organizaciones o incluso individuos autónomos, usualmente sin fines de lucro, sin posibilidades de lidiar con precios de componentes de alta calidad (típicamente de único acceso en el mercado internacional y no local) o licencias de software. Teniendo esto en cuenta, es importante marcar que entonces **nuestro proyecto también tiene como parámetro el intento de realizar el dispositivo de la manera más económica posible.** Esto va a determinar nuestras decisiones de diseño, implementación, y construcción final. Y la primer decisión al respecto es el uso de protocolos de estándar abierto donde corresponda, y software libre y gratuito (“FLOSS”) para el desarrollo lógico. Por esta razón, el software de este dispositivo será desarrollado sobre GNU/Linux, y será publicado en su completitud para el acceso de manera gratuita por parte de cualquier interesado, con licencia de distribución libre.

Entonces, en este proyecto vamos a poder aplicar conocimientos de buena parte (sino todas) de las asignaturas de nuestra carrera. Está el ejemplo evidente de las comunicaciones de control mediante software, para lo cuál vamos a lidiar con problemas tanto de electrónica como de lógica y programación, pero además enfrentamos diversos problemas de orden físico y mecánico cuyas soluciones están atravesadas por múltiples conocimientos especializados: fricciones, calibraciones, fuerzas necesarias para lograr algunos trabajos, materiales posibles, neumática, problemas de orden lumínico, entre otros. Creemos, además, que es perfectamente realizable en los tiempos y con los recursos disponibles.

El dispositivo está inicial e hipotéticamente compuesto (ya se verá en el desarrollo de este texto diferentes decisiones tomadas al respecto) por cinco módulos de diseño y construcción independientes, que deberán trabajar coordinados para lograr digitalizar un libro completo de manera automática. Un módulo se encarga del **control de una o dos cámaras**, para digitalizar como imágenes las páginas del libro a trabajar. Otro módulo opera como **sostén del libro**, y debe tener en cuenta diferentes disposiciones de libros así como también diferentes momentos del proceso de escaneo. Otro módulo

trabaja **aplanando las páginas**, de modo tal que se elimine su curvatura natural. Un cuarto módulo consiste en un dispositivo ad-hoc para **pasar de páginas automáticamente**. Y el quinto módulo es el **software, gestor de los eventos del proceso de escaneo y del empaquetado de la información**.

Cada módulo tendrá sus propios hitos a cumplir, y operará como caja negra con respecto a los demás módulos, comunicándose por señales que sabrá interpretar el software. Al tratarse de un modelo de cajas negras, en rigor existe cierta autonomía relativa entre los componentes del dispositivo, con las únicas excepciones del **protocolo de comunicación** que deban respetar y de la **disposición física** de los mismos; esto lleva a que se puedan plantear diferentes diseños para cada módulo, adecuándonos a las posibilidades reales de construcción con las que nos enfrentemos eligiendo entre múltiples diseños al caso.

Especulamos en un principio dar por finalizado y satisfecho al proyecto cuando el dispositivo tenga alguna versión terminada de todos y cada uno de los módulos que lo constituyen, y sea capaz de escanear ALGÚN libro (y NO CUALQUIER libro), independientemente del formato final (que se entenderá como un trabajo agregado de *post-processing*, y se entiende como ajeno al proceso de construcción del dispositivo). Sin embargo, llegado el momento, vimos que lograr trabajar los diferentes módulos correctamente coordinados es punto de conclusión suficiente, dándonos así tiempo para tomar muchas decisiones con respecto a futuras implementaciones de lo aprendido durante el desarrollo.

Para este proyecto se requirieron algunos sensores de diferente naturaleza, algunos motores, y algunos elementos de distribución como ser poleas y engranajes, para todos los módulos de orden físico (es decir, todos menos el software). El diseño final de cada módulo fué improvisado ad-hoc de acuerdo a los diferentes problemas que se enfrentaron y resolvieron; pero teniendo en cuenta modelos típicos/preliminares de estos módulos, asumimos sin mucho temor a equivocaciones que íbamos a necesitar al menos un motor servo o stepper para cada módulo, al menos un (y probablemente dos) sensores de fin de carrera para cada módulo, y muy probablemente también algún integrado para gestión y conversión de señales desde y hacia el software. La parte de los motores se mostró notoriamente desviada de la realidad. Además, necesitábamos al menos una cámara web para el módulo de adquisición de imágenes, dos placas de vidrio o acrílico transparente para el módulo anticurvaturas, y una infraestructura de madera y/o metal para el resto de la construcción de los módulos. Nuestro objetivo es lograr utilizar componentes reciclados en donde sea posible; al caso, tomamos motores, ejes, y otros componentes (como las placas de vidrio, por ejemplo) directamente desde dispositivos en desuso (scanners de tipo *flatbed*, impresoras, etc). El software será desarrollado sobre plataforma GNU/Linux, utilizando Python como lenguaje de alto nivel y C o dialectos de C con compiladores ad-hoc cuando se llegara a requerir grabar lógica en integrados (en cuyo caso, la primera opción será siempre Arduino).

Teniendo todo esto en cuenta, estimamos un tiempo de mes y medio de trabajo por cada módulo, que distribuyendo tareas pudiera realizarse incluso en paralelo.

Detalle y planificación general del diseño original

En primer lugar detallamos un poco los diferentes módulos mencionados durante el resumen del proyecto, de modo que se entiendan los diferentes grados de autonomía y vinculación que tiene cada uno con el otro.



Imagen N° 1 : disposición típica de scanner para libros de construcción casera.

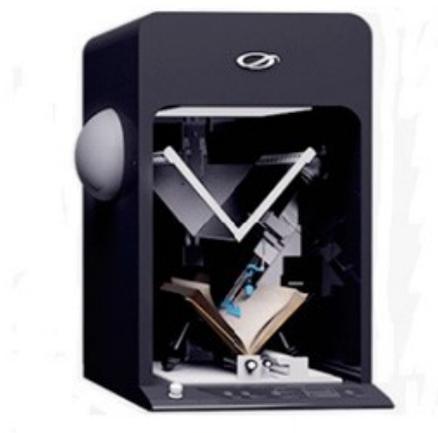


Imagen N° 2 : Misma disposición, pero en un scanner comercial.

Obsérvese en las imágenes N° 1 y 2 una disposición similar, si bien con una construcción final totalmente diferente. Esta es la disposición de la que hablamos durante el resumen. Básicamente, el corazón de esta estrategia de escaneo es la siguiente estructura:

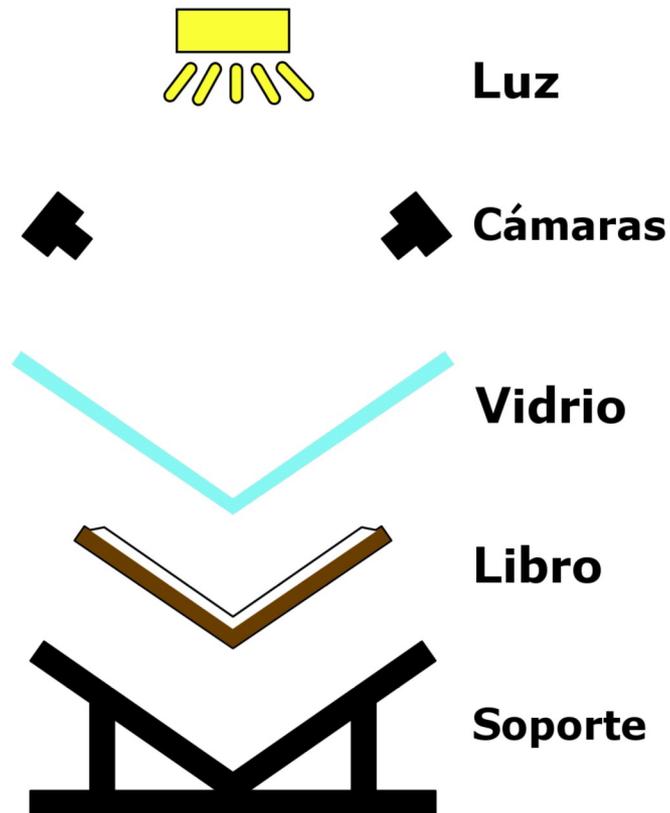


Imagen N° 3 : esquema de la disposición básica que vamos a tomar como base para el proyecto.

Nótese que no se trata de la única estrategia disponible. No vamos a explorar ninguna de las demás, porque no es el objeto de este documento.



Imagen N° 4 : ejemplo ilustrativo de otras estrategias de construcción de un scanner para libros.

Veamos cómo funciona la disposición que elegimos para nuestro dispositivo. En primer lugar tenemos el sostén/apoyo para el libro.



Imagen N° 5 : Imagen de un diseño de sostén para libro. (http://www.diybookscanner.org/archivist/?page_id=232)

Esta plataforma constituye nuestro primer módulo. Si bien a simple vista puede parecer una simple unión rígida, este soporte está pensado para ser compatible con diferentes tamaños de cuerpos de libros. Al caso, es necesario un mecanismo (enteramente mecánico) que permita abrir o cerrar las tablas del soporte.

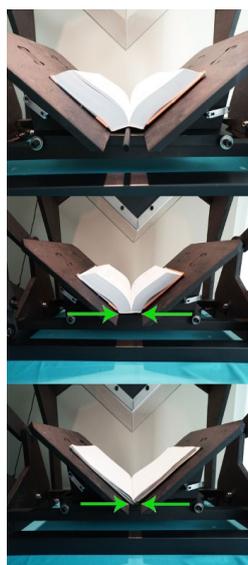


Imagen N° 6 : Movimiento del sostén para libro.

Además, es necesario que esta plataforma de sostén se pueda mover con facilidad sobre su eje horizontal. Esto se debe a que el centro de los libros no está en todo momento en el centro mismo del libro: hacia el principio y hacia el final del libro, el eje vertical cambia de posición sensiblemente con respecto al horizontal, y entonces este último debe adecuarse. Un simple movimiento sobre rodamientos respondiendo a presión será suficiente para lidiar con este pormenor.

Este primer módulo exige un trabajo de selección de materiales, diseño preciso (probablemente conste de varias piezas, algunas tal vez de precisión), y montaje tal que sea completamente compatible con otros dos módulos: el sistema anticurvatura de páginas, y el mecanismo de cambio de página automático. El usuario del dispositivo va a interactuar con este módulo simplemente posicionando al libro de manera adecuada; el scanner hará el resto del trabajo.

Siguiendo con el esquema de la imagen número 3, la estrategia para eliminar la curvatura de las páginas consiste en las placas de vidrio. Simplemente presionando dos placas de vidrio (o acrílico en su defecto, o algún otro material transparente al caso) sobre las páginas debería ser suficiente para que las mismas queden planas, sin curvaturas.



Imagen N° 7 : implementación de un sistema anticurvatura de páginas con vidrios en V.

En la imagen N° 7 podemos ver cómo el vidrio presiona las páginas, de modo tal que la curvatura se elimina. El ejemplo de la imagen da cuenta de la relación directa entre el módulo de apoyo del libro y las placas de vidrio. Pero en el ejemplo se aspira a un mecanismo manual, mientras que nosotros vamos a realizar un dispositivo enteramente automático.

Nuestro módulo anticurvatura entonces implica un trabajo menor de calibración con respecto al apoyo (dado que sencillamente las dimensiones no pueden ser arbitrarias), pero también implica idear un mecanismo que permita levantar las placas de vidrio automáticamente para más tarde pasar de página; el ciclo de funcionamiento de este módulo arranca cuando presiona un juego de páginas antes de una fotografía a las mismas, y termina cuando le deja lugar al mecanismo de cambio de página una vez ya tomada la fotografía. Por esto también es necesaria una coordinación entre estos módulos para un proceso ordenado. Al caso va a ser necesario que, cualquiera sea el diseño de este módulo, cuente con alguna forma de sensado que permita al software gestor del proceso tomar las decisiones para continuar con los pasos correspondientes; eso es, detectar cuando ya está presionando las páginas, y cuando ya está a una altura moderada para activar el sistema de cambio de página.

El siguiente módulo, de acuerdo al esquema en la imagen número 3, son las cámaras. Para esta disposición de módulos, lo típico es utilizar dos cámaras digitales montadas fijas perpendiculares a las alas del sostén del libro. Al activarse las cámaras, toman una fotografía a través del módulo de aplanado de páginas (y por ello es necesario que sea transparente). Y lo común es que se utilicen al caso cámaras fotográficas digitales de uso común, obteniendo imágenes digitales del orden de los 20MP, y obteniendo las dos páginas a un tiempo. Este montaje típico puede verse en la imagen N°1; a la izquierda se ve una de las cámaras montadas, y en el centro un módulo anticurvatura sobre un módulo sostén.

Pero nosotros tenemos como lineamiento en este proyecto tratar de relizarlo de la manera más económica posible, y las cámaras digitales que se utilizan para estos proyectos tienen valores del orden de las centenas de dólares; que se traducen en miles de pesos argentinos, y se convierten entonces en un cuello de botella económico para el proyecto. Al caso nosotros planteamos la estrategia de utilizar una sola cámara en primer lugar, y que no sea una cámara fotográfica digital sino una cámara para computadora de escritorio (“webcam”).

Esto nos trae algunas ventajas y algunas desventajas. En primer lugar, la webcam es lógicamente más económica (en el orden de las centenas o incluso hasta decenas de pesos argentinos), pero la calidad de la imagen es radicalmente inferior, y sólo algunas webcams puede llegar a ser aptas para este tipo de trabajo. Además, las webcams tienen como detalle agregado que poseen un zoom óptico (y no digital) fijo, a diferencia de las cámaras fotográficas que permiten configurarlo. Esto nos exige una calibración de la distancia de webcam mucho más reducida, y como consecuencia también nos trae problemas con la iluminación (porque si la cámara no está a una distancia prudencial, generará sombra sobre el papel y eso es indeseado). Y además, nosotros planteamos el uso de una sólo cámara, con lo cuál debemos adecuarla a un mecanismo que le permita girar hacia una página u otra, de manera correctamente calibrada. Todo esto complejizará bastante el diseño e implementación de nuestro módulo de adquisición de imágenes.

Nuestra estrategia será montar una cámara móvil sobre algún eje o riel adecuado al mecanismo mismo anticurvatura (las placas de vidrio), de modo que se encuentre suficientemente cercano al papel para poder lograr una calibración adecuada, y no se vuelva complicado el giro para obtener ambas páginas. También pensamos montar la iluminación sobre el mismo módulo, detrás o por encima de la cámara, frente a lo cuál tendremos que lidiar con estelas de brillo y otros problemas lumínicos. La idea es que, entonces, cuando opera el módulo anticurvatura, se activa la webcam para una página, y luego se mueve 90° para obtener la otra página. Otra opción puede ser agregarle un lente a la webcam, de modo que tengamos mejor rango de trabajo y mejor calidad de imagen. E incluso consideramos la opción de automatizar alguna cámara digital (no webcam) barata, usada, antigua, manipulando los controles de manera electrónica (por ejemplo, generando un contacto en el botón para obtener imágenes). En todos los casos, se pretende utilizar una sola cámara, y moverla para centrarla en cada página.

Hasta aquí exploramos los módulos indispensables de un montaje típico para scanner de libros. Nuestro scanner es 100% automático, con lo cuál tiene dos módulos agregados. El primero, es un módulo que permita cambiar de páginas de manera automática.



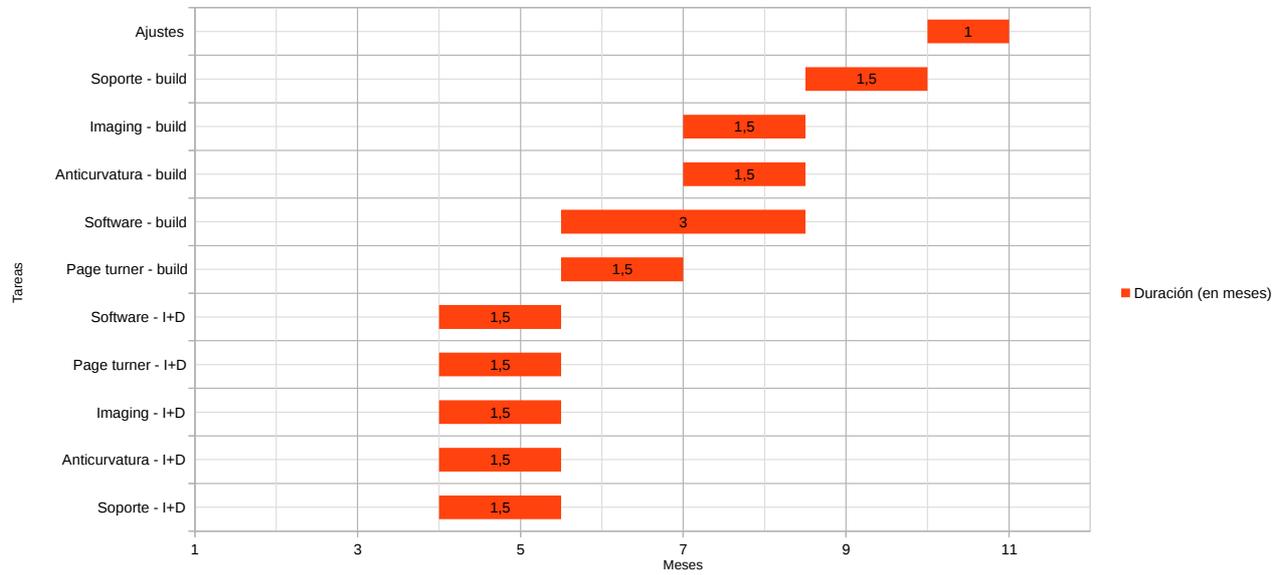
Imagen N° 8 : ejemplo de un dispositivo para cambio de página en un libro.

Este módulo, pensamos, sería definitivamente el que más ingenio requeriría. Aunque existan algunos diseños más o menos populares para un scanner de libros, el dispositivo que permite cambiar de páginas es ciertamente exótico y poco estandarizado. Es un dispositivo cuyo trabajo está muy determinado por la calidad del material y el tamaño del libro, dos problemas que en el caso de la adquisición de imagen se resuelven aumentando la calidad de la cámara, pero que aquí no resulta tan simple. Necesitamos poder separar una página (y sólo una) del libro, para poder moverla hacia su destino de manera sincronizada con el resto del proceso.

Se pueden usar diferentes estrategias para esto. Usualmente se apunta a la fricción, de modo que pueda curvarse la página (simulando la fricción que ejerce un dedo), para luego terminar de empujarla. Esto se podría lograr con componentes de impresoras, que trabajan con materiales especialmente adecuados para arrastrar papel mediante fricción. Pero nosotros vamos a apuntar para este módulo por una estrategia que involucre la succión de aire para sujetar levemente la página a desplazar, de modo que luego podamos moverla. Al caso vamos a armar algún dispositivo de succión de construcción casera, y utilizaremos algunos motores tanto para el dispositivo de succión como para mover la página hacia su posición final; probablemente un motor de CC para el succionador, y un servo para algún otro mecanismo agregado que mueva la página. La estrategia de la fricción la dejaremos como segunda alternativa, en caso de que nos topemos con problemas insalvables.

Y por último, al tratarse de un scanner enteramente automático, debe tener un software gestor del proceso completo. Este software sabrá interpretar señales del resto de los módulos, sabrá indicarles qué hacer y cuándo, y sabrá en última instancia recopilar los datos del proceso completo: el escaneo del libro listo para continuar la línea de trabajo en *post-processing*.

Cronograma original planificado



I + D: Módulo de obtención de datos (“imaging”)

El centro de nuestra investigación está en la posibilidad de utilizar o no webcams para nuestro proyecto. Esto por dos razones:

* Las webcams son notablemente más baratas que cualquier otro dispositivo que podamos utilizar para obtener datos.

* Las webcams pueden ser controladas fácilmente mediante protocolos estandarizados.

Una de las referencias en el tema de armado de dispositivos como el nuestro, Daniel Reetz de <http://diybookscanner.org>, recomienda enfáticamente no utilizar webcams (ver http://www.diybookscanner.org/archivist/?page_id=651). Nosotros necesitamos confirmar si son posibles de usar o no, en un lapso de un mes de acuerdo a nuestros tiempos.

Al caso, exploramos una serie de estrategias posibles. La primera es buscar webcams de calidades tolerables. En nuestra experiencia, a partir de 1.3 MP podrían llegar a funcionar.

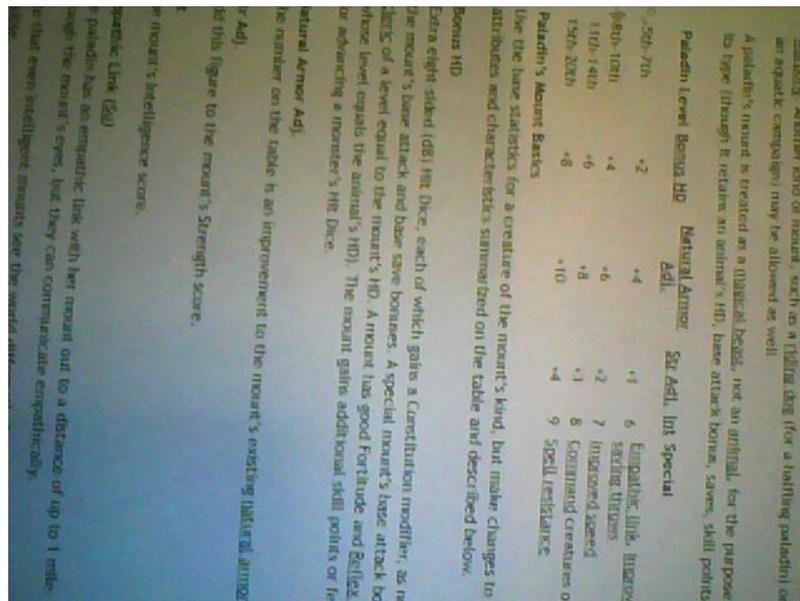


Imagen N° 9 : ejemplo de imagen obtenida con una webcam de 1.3MP sin calibrar.

La imagen N° 9 muestra un texto mayormente legible, teniendo en cuenta que la imagen se obtuvo con un tamaño máximo de 640x480 pixels (no es la mejor calidad que permite la cámara). Pero con la comunmente llamada “calidad de la cámara” no alcanza, como se verá a continuación:

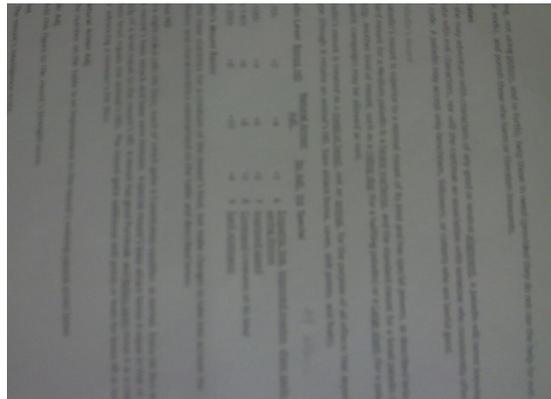


Imagen N° 10 : otra imagen obtenida con otra webcam de 1.3MP.

La imagen N° 10 se obtuvo con OTRA cámara web de 1.3 MP, y esta vez a su máxima resolución. El resultado: un texto borroso y absolutamente ilegible.

La nitidez de la imagen no depende enteramente de la cantidad de megapixels que tolere la webcam. Descubrimos investigando que también depende del lente que se utilice en la webcam. Algunos entusiastas han hecho la experiencia de modificar webcams para diferentes tareas de observación: desde cámaras de seguridad, hasta fotos panorámicas, hasta astronomía (ver <https://www.youtube.com/watch?v=tKS5eI-VQxM>).

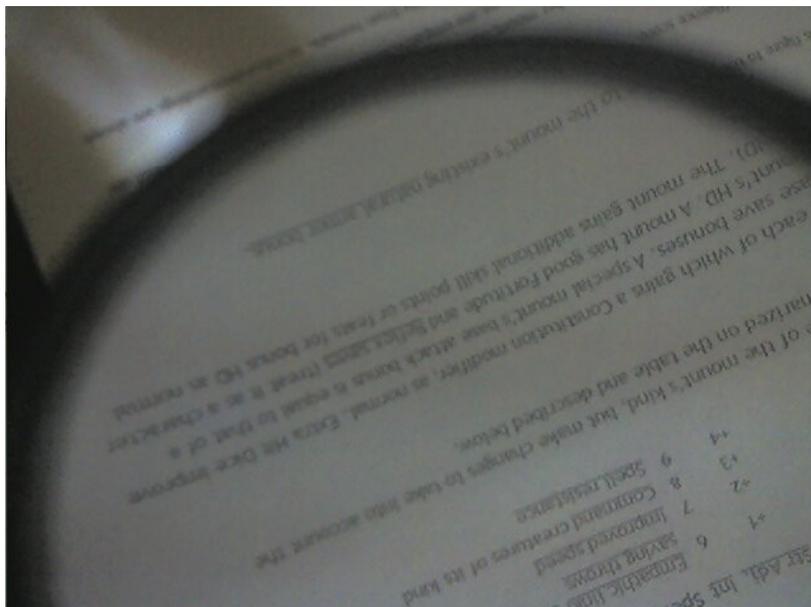


Imagen N° 11 : Misma cámara que la imagen N° 10, pero con una lupa.

Teniendo esto en cuenta, empezamos algunos experimentos vinculados al lente. La imagen N° 11 muestra que, simplemente agregando una lupa, podemos obtener una imagen más nítida (y ahora sí legible) del mismo texto.

Siguiendo esta línea de experimentación, conseguimos un lote de diferentes cámaras en desuso, de las cuales obtuvimos sus lentes. También tuvimos acceso a una webcam de 320KP (la resolución más baja del mercado) para poder hacer experimentos con ella. A continuación los resultados.



Imagen N° 12 : imagen obtenida con la cámara de 320K y lente original.



Imagen N° 13 : Cámara de 320K, con lente de otra webcam agregado.

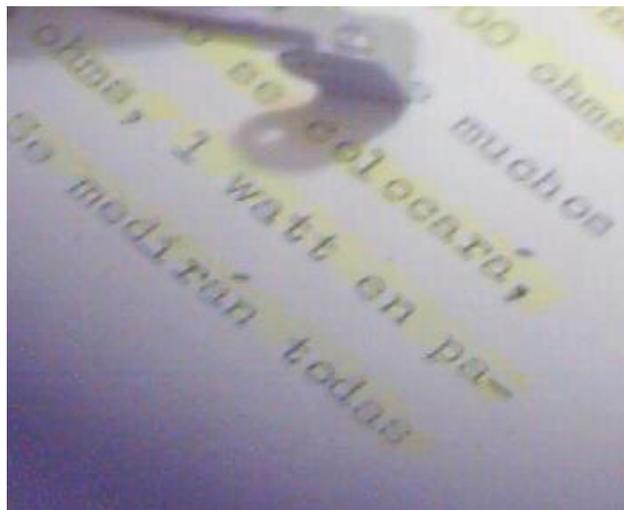


Imagen N° 14 : Cámara de 320K, con lente de zoom agregado.

Como se puede ver en las imágenes, el cambio de lente determina cambios radicales en la obtención de la imagen, aún cuando la cámara no permita gran calidad. El secreto es lograr una disposición de lentes tal que permita a la cámara un rango de una página entera y además un ajuste de foco tal que permita la nitidez suficiente para imágenes legibles. Esto será en buena medida independiente de la webcam, teniendo en cuenta una resolución mínima adecuada (que especulamos alrededor de los 1.3MP, probablemente 2MP).

Lamentablemente, las webcams suelen tener un foco estático, y no disponer de mecanismo para agregarles accesorios ópticos como sí los tienen las cámaras fotográficas. Y no sólo eso, sino que también es usual no encontrar especificaciones técnicas reales de las webcams; usualmente se promocionan con parámetros que no están determinados por su hardware sino por mecanismos de software (como es el caso con los megapixels que provee). En este problema se incluye la capacidad de zoom que suelen disponer: realizan un zoom por software, que agranda la imagen (y por ello se dice que la cámara tiene más megapixels) a cambio de perder nitidez; y para nuestro caso de uso la nitidez es crucial, porque sin nitidez el texto es ilegible. Necesitamos zoom óptico.

Los próximos problemas a enfrentar serán entonces los pormenores del montaje de estos lentes. El nivel de zoom estará determinado por los lentes, pero necesitamos algún mecanismo para poder calibrar el foco entre dos lentes dados. Además, como se nos advierte en diybookscanner.org, existen otros parámetros tales como la exposición del lente o la apertura. Lo cuál hace que esta estrategia parezca, en principio, muy difícil de lograr con los tiempos que disponemos.

En esta investigación también descubrimos otro factor que puede llegar a servirnos: las webcams suelen venir con un lente que tiene un tamaño aparentemente estandarizado. Es decir, que existen lentes fácilmente intercambiables para las webcams (aunque no existe un mercado específico para tal caso de uso), y si consiguiéramos lentes de estas características que se adecúen a nuestra necesidad de zoom y foco tendríamos una solución eficiente (precios mediante).

Si ninguna de esas opciones llegaran a funcionar, otra estrategia es simplemente buscar algunas de las cámaras recomendadas en diybookscanner.org (las cannon hackeables) en el mercado de segunda mano. Pero aún usadas estas cámaras suelen ser costosas, y cuando son baratas se encuentran defectuosas.

También existe la posibilidad de trabajar con algunos modelos de teléfonos celulares, que tienen cámaras de resoluciones considerables. Pero tal y como sucede con las cámaras fotográficas digitales, esto nos determina a utilizar mecanismos de control subestandar, usualmente privativos, que convierten al proyecto en una experiencia irreplicable y traen un cuello de botella en el módulo de software.

También estamos evaluando la posibilidad de utilizar módulos de captura de imagen para Arduino o Raspberry PI. Son básicamente una webcam. Pero no son populares en Argentina, salvo por una versión de muy baja calidad, y los precios de los componentes de mejor calidad en el exterior tampoco se mostraron muy tentadores para el proyecto (a partir de los 30 dólares).

Otra opción posible sería dismantelar algún teléfono celular e interfasear su cámara web directamente utilizando electrónica. Pero nuevamente vamos a tener dificultades con todos los componentes que no sean populares y no tengan información pública.

Y por último, la última opción que contemplamos es trabajar directamente con video analógico: utilizar alguna antigua cámara filmadora en desuso, conectando la salida de video de la cámara a la entrada de video de la computadora. Esto sería de control sencillo. Pero no todas las computadoras disponen de entrada de video analógico, y además las cámaras aún cuando en desuso suelen ser también costosas.

I + D: Módulo para cambio de página (“page turner”)

Como mencionáramos en el resumen inicial, nuestra estrategia principal para el pasaje automático de páginas iba a estar basada en succión. Esto debido a que, especulábamos, la fricción estaba mucho más sometida a condiciones físicas de los componentes (fundamentalmente del papel), y era mucho más complicado lograr un mecanismo de precisión (anti-error) basado en fricción que en succión. Pero así y todo, aparentemente la primer opción de muchos otros proyectos online siempre fue la fricción.

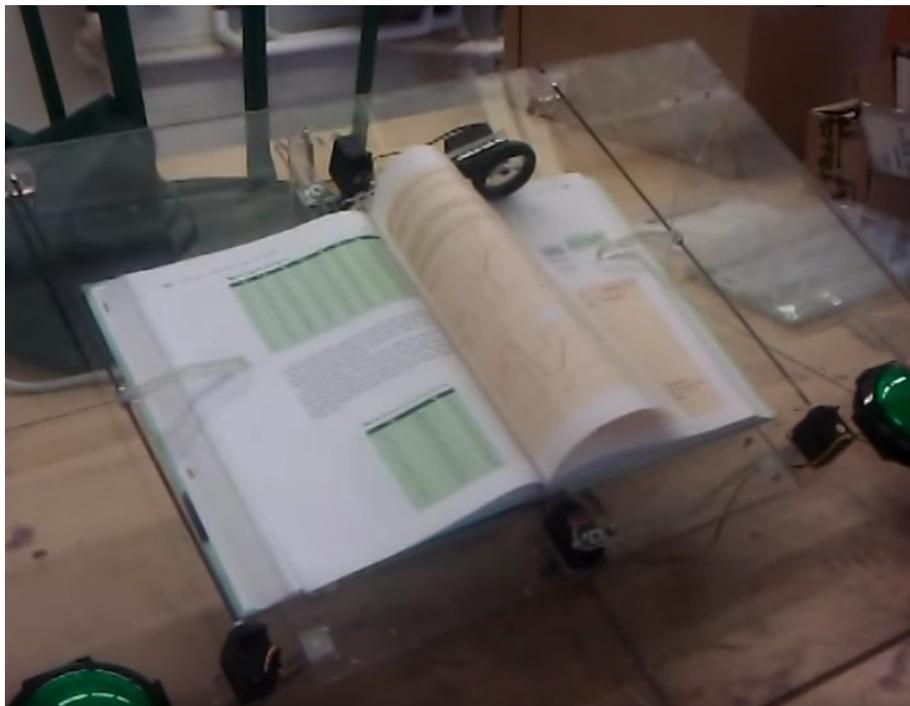


Imagen N° 15: dispositivo pasa páginas de fricción con rueda.

Para la fricción es muy común utilizar ruedas o rodillos, que encorvan la página con su mecanismo de giro, y usualmente se utilizan en combinación con algún otro mecanismo que eventualmente empuje la página encurvada, como puede ser una pequeña palanca. Esto requiere un par de motores que trabajen coordinados para lograr un sólo movimiento.

La succión, por otro lado, no requiere mucha precisión y es bastante independiente de las características del papel. Nuestro plan original fue montar alguna forma de tubo de succión, probablemente utilizando algún pequeño compresor barato y/o de fabricación casera, o incluso pensamos hacerlo controlando el émbolo de alguna jeringa con una ventosa de goma en la punta, y el movimiento a realizar serían básicamente dos pasos: primero presionar un borde del tubo de succión contra la página a mover, y luego mover el tubo durante el succionado. Esto nos permitiría movimientos en una sola dirección, y un grado de eficiencia del mecanismo que no dependiera del libro sino de los componentes del módulo.

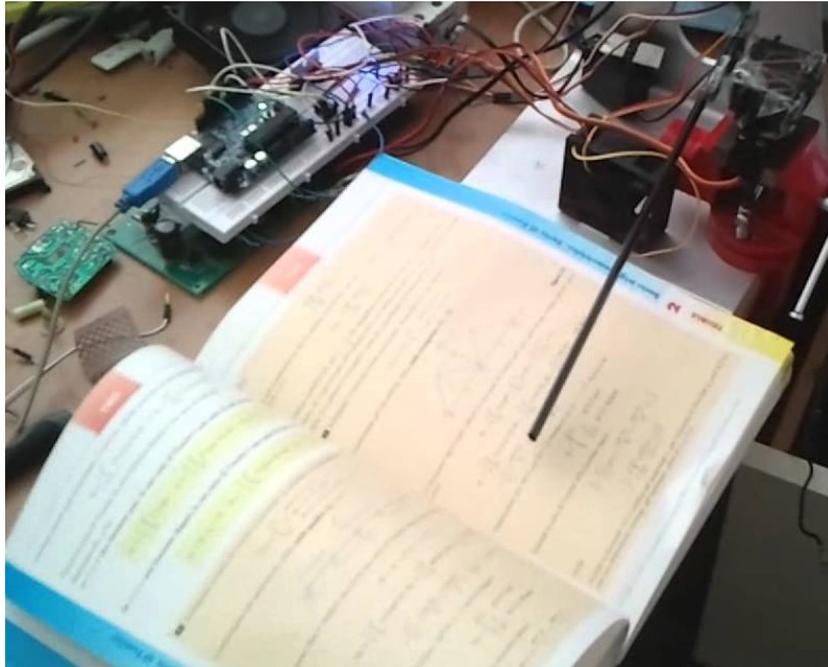


Imagen N°16: mecanismo pasa páginas de succión. Nótese el pequeño tubo apuntando hacia la página a mover.



Imagen N°17: mecanismo pasa página con punta adhesiva. El principio es similar al de succión: adherir la página para moverla.

Existen algunas variaciones de estos mecanismos generales. Por ejemplo, el de punta adhesiva, que consisten en una estrategia similar a la de succión con tubo, pero sin succión sino adheriendo la página mediante algún componente adherente en la punta del dispositivo; luego sólo se requiere, al igual que con el tubo de succión, mover la punta adhesiva para que la página acompañe el movimiento.

Pero encontramos algunas variaciones del mecanismo de succión también prometedoras, especialmente de cara a los intereses de este proyecto (simpleza, para que cualquiera pueda construirlo, y componentes baratos). Una de esas variaciones es el uso de coolers para succionar, reemplazando al tubo.

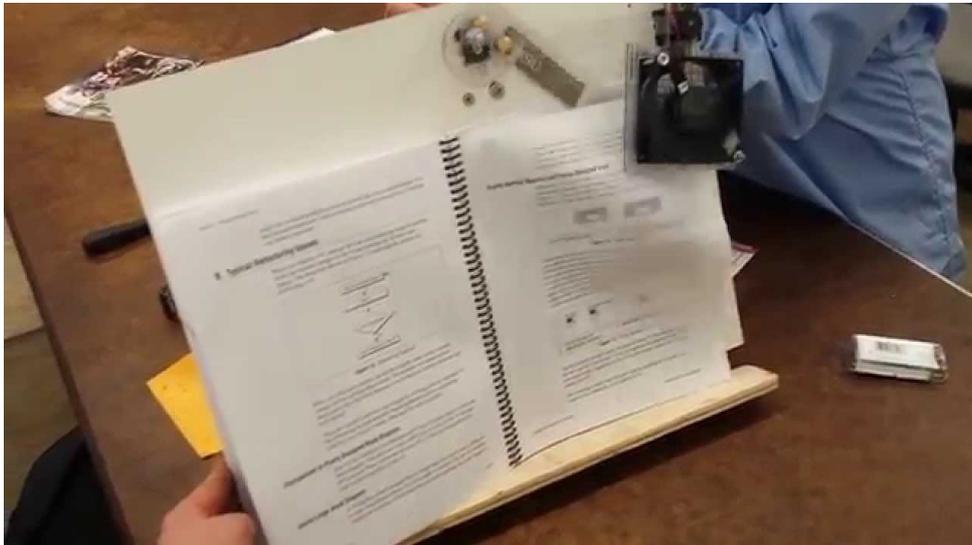


Imagen N° 18: Mecanismo pasa página con cooler.

Investigando diseños de otras personas dedicadas al tema durante la última década, pudimos ver que algunos llegaron a experiencias productivas con componentes sumamente económicos como los coolers. El secreto es manipular el área de succión, tal y como nosotros mismos estudiamos en Sistemas Hidráulicos y Neumáticos durante el año 2015. Como se puede ver en la figura N°16, alcanza un cooler tiene la fuerza de succión suficiente para levantar una página, que luego es manipulada por un segundo mecanismo de palanca (tal y como en los mecanismos de fricción que encorvan la página).

Pero siguiendo con la investigación, nos topamos con un diseño que finalmente fuera el que elegimos, y puede verse en acción en la siguiente dirección: <https://www.youtube.com/watch?v=pkZ8OfgRig0>



Imagen N°19: diseño de succión central.

Ese diseño dispone el área de succión en una abertura entre los dos vidrios del módulo anticurvatura de páginas, logrando así integrar dos de nuestros originalmente planeados módulos en uno. Realizamos algunas pruebas de esta estrategia y fueron exitosas, razón por la cuál luego de este descubrimiento readecuamos toda la estructura y planificación del diseño del dispositivo completo para cumplir con esta estrategia de escaneo.

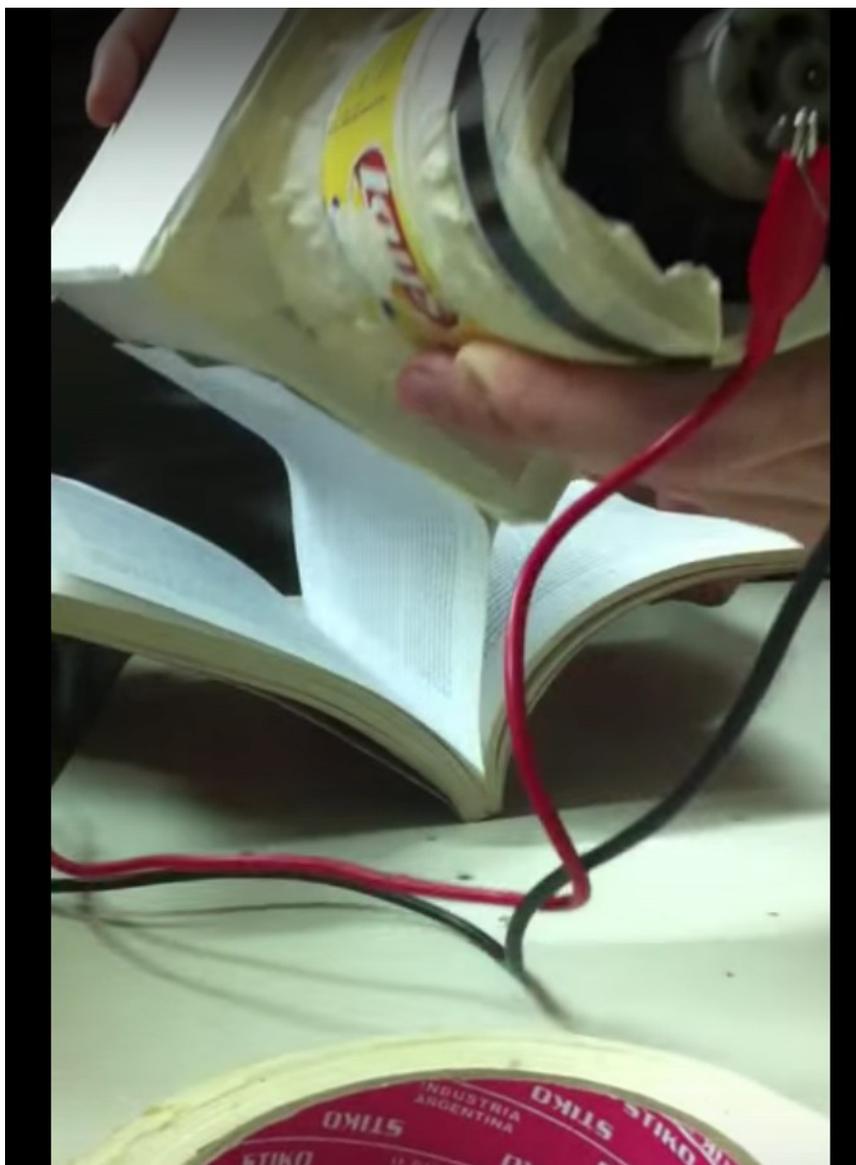


Imagen N°20: primer prueba con la estrategia de succión central.

Con esta estrategia, el diseño se volvía mucho más elegante, y había menos problemas para resolver: sólo necesitamos un motor que suba y baje el plato, el mismo movimiento opera como mecanismo pasa páginas, y luego solamente necesitaríamos sensores para detectar los diferentes momentos del ciclo de escaneo. En la imagen número 20 se puede ver que el primer experimento lo hicimos con un poco de cartón, cinta de papel, y un motor adecuado a una lata; exactamente el tipo de tecnología que se pretende para este proyecto. Luego de esto, ya no seguimos investigando esta cuestión.

I + D miscelaneo

La cuestión software siempre fue bastante clara: el software será públicamente accesible con licencia libre, y al caso será programado en plataformas también libres. Por ello se utilizarán GNU/Linux y lenguajes amigables para el software libre, como ser Python. Y existiendo Arduino como un standar de facto para electrónica comunitaria, es ciertamente la opción dada para toda la parte de hardware.

Pero el misterio a solucionar de cara al software estaba en el control de las cámaras. No podemos hacer un software que utilice componentes de control privativos, ni podemos hacer software hackeando dispositivos que no permitan dichos usos, porque si ese fuera el caso estaríamos incurriendo en delito y el proyecto no sería replicable de la manera que nosotros pretendemos. Por esta razón, la opción inicial fueron las webcams compatibles con sistemas tales como SANE y/o Video4Linux. De esa manera tenemos múltiples estrategias para acceder a los dispositivos. No es el caso con otros dispositivos como las cámaras de fotos digitales, las llamadas "cámaras ip", y algunos otros etcéteras.

Experimentando con cámaras nos encontramos con que algunas de ellas jamás llegaron a ser compatibles con GNU/Linux. Esos casos son los menos. Pero también durante nuestra investigación nos dimos cuenta de un detalle con respecto a las webcams que resultó preocupante para el proyecto. Sucede que las webcams son una especie en extinción; se usan cada vez menos, dado el auge de cámaras integradas tanto en computadoras portátiles como en teléfonos. Eso las convierta mayormente en chatarra, lo cuál ayuda a que sean más baratas: pero un modelo que se encuentra hoy en el mercado, en pocos meses es un modelo discontinuado, y eso es un problema para la longevidad del proyecto. Y si bien existen webcams de calidad en el mercado que difícilmente desaparezcan rápido, también tienen un costo más bien elevado, en el orden de los USD 100; esos precios no son adecuados para este proyecto.



Imagen N°21: calidad lograda con una cámara de ARS 140 y compatible con software libre.

Eventualmente encontramos cámaras de muy bajo costo para estándares actuales (ARS 140 en 2016), con 1.3MP reales de calidad máxima y una nitidez aceptable, y de moderada disponibilidad en el mercado. Con esas cámaras, una vez calibradas (manualmente), obtuvimos imágenes aceptables, y dimos por cerrada la cuestión cámaras. Pero cuando debamos replicar la experiencia, seguramente tendremos que rever este tema.

El resto de la etapa más fuerte de la investigación estuvo asignada al mecanismo para subir y bajar el plato anticurvatura de páginas. Pero al respecto vamos a hablar mas en el próximo apartado, donde comenzamos a detallar la experiencia de construcción general del dispositivo.

Proceso de constitución y construcción del dispositivo

Luego de algunas certezas pasado el proceso de investigación, la estrategia de construcción se basó en ubicar una disposición tal de componentes que nos permita realizar un mecanismo de succión central adecuado los principios de este proyecto: barato, simple, replicable, y libre.

Sabíamos desde un principio que para la autonomía del dispositivos necesitaríamos motores con los cuales implementar las funciones motrices, y nuestra primer estrategia fue apuntar a algunos motores populares, muy fáciles de conseguir y de manipular: los motores de corriente continua de 24V que suelen integrar las impresoras de chorro de tinta.



Imagen N°22: motor de corriente continua de 24V de impresora.

Estos motores tienen como ventaja de que se obtienen sumamente baratos (cuando no gratuitos) desde chatarra, y al ser de corriente continua no requieren controladores sofisticados: un control binario para el prendido y apagado, e inversión de polaridad para manipular el sentido de rotación. El desafío pasó a ser entonces de qué manera podríamos utilizar estos motores para automatizar el scanner; concretamente, para subir y bajar el plato de succión y anticurvatura. En paralelo fuimos construyendo nuestra primer versión del plato, para optimizar nuestros tiempos.



Imagen N°23: primera versión del plato.



Imagen N°24: primera versión del plato.



Imagen N°25: primera versión del plato.



Imagen N° 26: primera versión del plato. Montaje con columna.

Inmediatamente que construimos el plato, nos dimos cuenta de que sería pesado para los motores que manejábamos, y que con ellos no podríamos implementar un mecanismo como el que originalmente vimos en el diseño que inspiró nuestro modelo. Además, notamos que al montarlo sobre dos simples columnas centrales que pretendíamos solidarias al movimiento del plato, se nos agregaban muchos puntos de imprecisión al movimiento como inclinaciones o juego, lo cuál requería todavía más fuerza para poder ser superados por cualquiera fuera el motor que elijamos.

Teniendo esto en cuenta, nuestra primer idea de solución fue trabajar con alguna forma de sistema de contrapesos, como lo hacen los ascensores, de modo que no se le exija mucho torque al motor. Y fuimos nuevamente a las comunidades especializadas para revisar qué mecanismos tenían previamente diseñados otros equipos.



Imagen N°27: estrategia de palanca para movimiento del plato.

Especulamos con la posibilidad de montar un mecanismo de palanca con contrapeso, similar al de la imagen N° 23. Esto lo moveríamos con uno de los motores CC 24V comentados anteriormente, ya sea directamente en el eje del mecanismo o a través de alguna línea de transmisión como ser cinta dentada o cadena.

Pero comentando esta cuestión con algunos de los profesores de la institución, nos pareció muy tentadora la estrategia de la varilla roscada. Básicamente, uno de los motores con el eje fijado a una varilla metálica roscada, de un valor económico y fácil de conseguir (~ARS30, y en venta en cualquier ferretería), que al girar moviera una tuerca montada de alguna manera en el plato. Este es un mecanismo que permite una gran reducción, permitiendo mover cargas muy superiores a las que permite el torque original del motor, a cambio de velocidad en la operación (es un mecanismo lento). La simpleza (no requería alta precisión ni componentes complejos) y economía de este mecanismo nos llevó a implementarlo en nuestro primer experimento.



Imagen N°28: mecanismo con varilla roscada.

Para este primer experimento, aprovechamos también para agregarle un segundo juego de columnas. Las mismas, a diferencia de los diseños más populares implementados en scanners para libros, fueron montadas en el frente y contrafrente del plato, mientras que lo usual es tener los ejes solidarios a los lados, como en la imagen N° 26.

De acuerdo a lo que se nos había informado, el mecanismo de reducción de la varilla roscada no requería un motor de gran torque, sino de gran velocidad (para lograr que el movimiento no fuera ya demasiado lento). Y al caso medimos la velocidad de nuestros motores con un sensor improvisado, que puede apreciarse en la imagen N°22 una rueda dentada, y un optoacoplador (que se ve en el fondo).

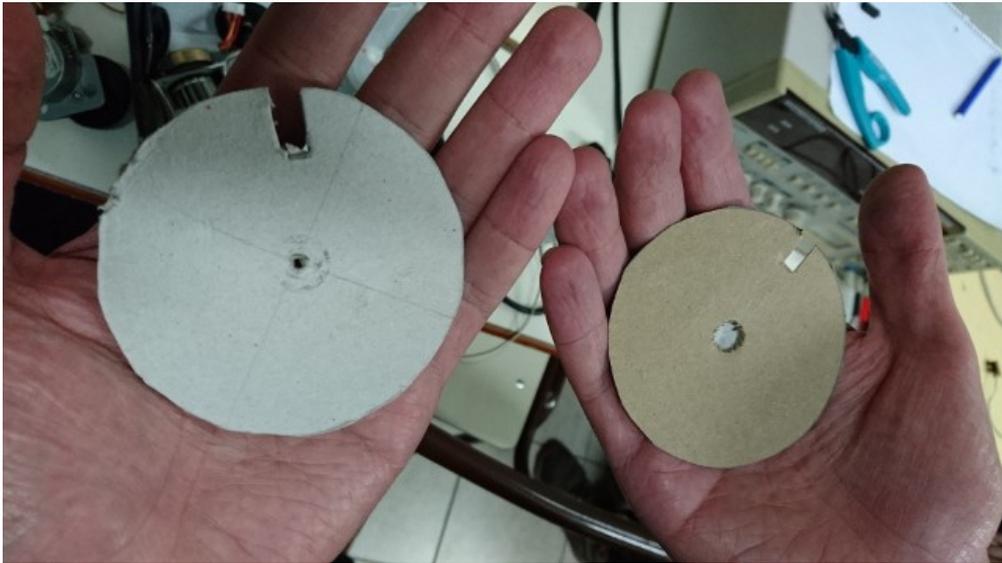


Imagen N°29: ruedas dentadas para medición de RPMs.

Se nos indicó que a partir de ~ 3000 RPM los motores podían empezar a considerarse útiles para este mecanismo, y nuestros motores mostraron un rendimiento cercano a los 6000. Con lo cuál era una estrategia sumamente prometedora.

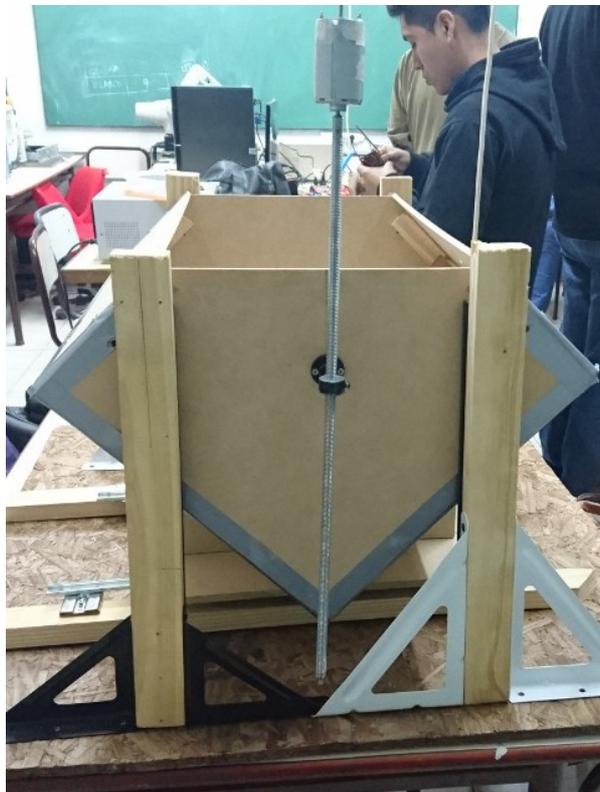


Imagen N°30: montaje final con varilla roscada.

Sin embargo, el experimento fue un absoluto fracaso. El mecanismo de reducción de la varilla roscada finalmente sí requería precisión y componentes de más alta calidad, dado que las varillas baratas tienen defectos de curvaturas, de fabricación de los dientes, se deforman fácil (de manera apenas perceptible por una persona, pero generando un requisito de trabajo mucho más alto), y como si fuera poco existe un juego tolerable para el mercado de estas piezas entre la varilla roscada y las tuercas. Todos y cada uno de esos detalles hacían que el motor no tuviera la suficiente fuerza para levantar el plato, aún cuando le diéramos más estabilidad y el movimiento fuera guiado.

Para este momento ya entrábamos en el mes de septiembre, y no teníamos una idea clara de cómo construir el mecanismo. Cada prueba que ideábamos fallaba inmediatamente en los experimentos preliminares, debido a que nuestros motores no contaban con el torque suficiente, y los parámetros del proyecto (buscar elementos fáciles de conseguir, simples de trabajar, y accesibles) hacían que nuestras opciones fueran cada vez menos.

Con esta preocupación, decidimos reconstruir el plato para lograr una versión mucho más liviana (la original pesaba alrededor de 5 kilos), buscar la manera de darle todavía más estabilidad al diseño final (para reducir requisitos de fuerza impuestos por fricciones y movimientos indeseados), y recién entonces ver qué podíamos implementar.

Y además todavía teníamos otra cuestión sin resolver, que era la iluminación. Nuevamente encontramos información al respecto en las comunidades especializadas, y se nos informaba que la iluminación era un problema, porque se requería cierto grado de precisión en el ángulo de la luz para lograr tanto una imagen nítida como también para no generar reflejos en la imagen, que en definitiva es obtenida vidrio mediante. Y nuestro plan original era usar luces recicladas de scanners hogareños para iluminar dentro del plato desde adentro.

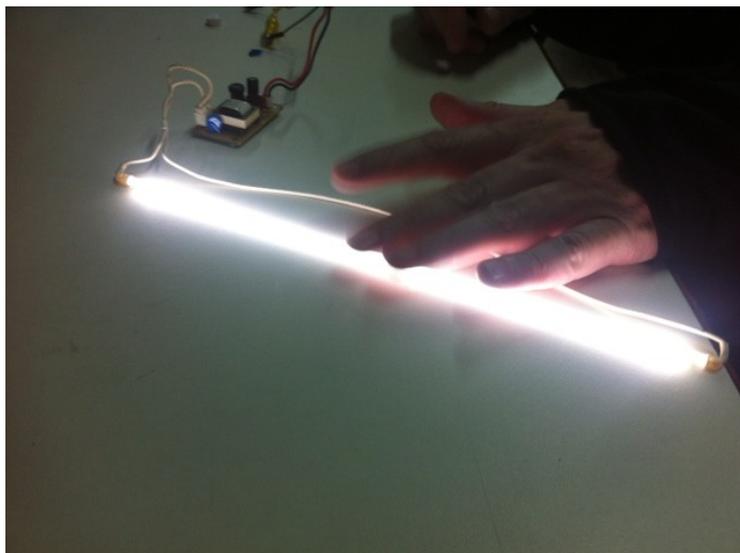


Imagen N°31: luz de scanner hogareño reciclada.

Finalmente, dado nuestro nuevo diseño de plato, decidimos utilizar tiras de leds montados dentro de los ejes del plato mismo para iluminar.



Imagen N°31: segundo diseño de plato.



Imagen N°32: segundo diseño de plato. Detalle del montaje de leds.

Este segundo diseño consta de solamente un esqueleto hecho en madera y con uniones de metal, y tapas de cartón. Con esta estrategia logramos reducir el peso del plato a poco menos de un kilogramo. En simultaneo, durante este momento de rediseño, decidimos probar una estrategia que no vimos implementada en ningún otro diseño: un polipasto, o aparejo de poleas. Eso debería ser un mecanismo de reducción también muy barato y fácil de implementar.



Imagen N°33: primer montaje con aparejo de poleas.

Para este nuevo diseño, también aplicamos un sostén ya no de columnas sino directamente de paredes, emulando al diseño original que inspirara al nuestro. Y con esta estrategia probamos nuevamente intentar levantar el plato con nuestros motores de CC.



Imagen N°34: montaje con paredes y el segundo diseño de plato.

Nuestro nuevo experimento fue otro fracaso; el motor no logró levantar el plato. Se logró apenas algún movimiento durante el régimen transitorio de encendido del motor, pero inmediatamente después fallaba. Algun experimento de combinar más de un motor en el mismo eje fue más prometedor, pero también se mostraba determinado por algunas imperfecciones que convertían al dispositivo en un mecanismo de precisión, algo indeseado para el proyecto.

Sucedía, además del muy bajo torque de nuestro motor, que nuestras poleas baratas tenían un alto grado de fricción y juego en sus ejes; eran ruidosas, y se sentía el trabajo que ejercían ya solamente al intentar mover el mecanismo con la mano. Y si bien era muchísimo más liviano que el diseño original, seguía siendo complicado de lograr por las mismas premisas del proyecto: los componentes baratos y accesibles en el mercado local no se prestaban a las tareas a las que los estábamos sometiendo.

Para este entonces ya estábamos bien entrados en Octubre y todavía no teníamos un mecanismo funcional para el scanner. Todas nuestras opciones atentaban contra alguno de los ejes del proyecto: algunos decían que debíamos implementar algún mecanismo motoreductor con engranajes impresos en impresora 3D; pero esto nos llevaba a diseñar piezas de precisión y depender de disponibilidad de una impresora, las cuales no son baratas, o enviar a construir las piezas, lo cuál es todavía más caro. Otras personas nos recomendaban simplemente cambiar los motores y utilizar alguno con motoreducción incluida; pero los motores suelen, nuevamente, no ser baratos, además de que empiezan a requerir otras tensiones y corrientes, y como si fuera poco es común que se utilicen motores paso a paso o de corriente alterna, cambiando así nuestra estrategia de control de los motores y muy a último momento. Y también se especulaba con la posibilidad de conseguir piezas de mejor calidad (otros materiales más caros, otras piezas más precisas), siempre con problemas similares.

La decisión final fue la de cambiar de motor. Utilizamos un motor de limpiaparabrisas de automóvil, que nos acercara un profesor del instituto: estos motores son de corriente continua, tienen buen torque porque cuentan con reducción integrada (tienen que superar la fricción de la goma sobre el parabrisas), son conseguibles en el mercado local, y existen en diferentes gamas de precios, algunos bastante accesibles (en el orden de los ARS 300). Con este motor improvisamos un carrete soldado de metal, y así finalmente tuvimos nuestro primer experimento exitoso.

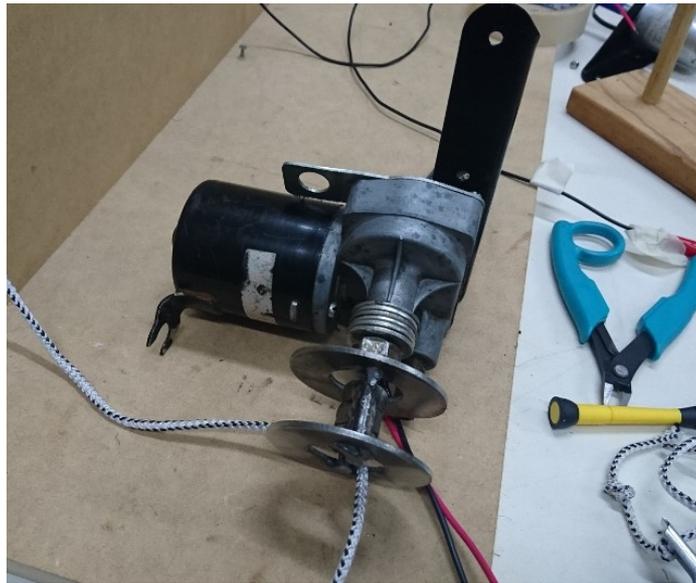


Imagen N°35: motor de limpiaparabrisas con carrete improvisado de metal.

Durante este experimento, descubrimos que el diámetro del carrete era fundamental para determinar la velocidad del mecanismo (un mayor diámetro nos da más velocidad), y que el modo de construcción también debía ser cuidado para evitar accidentes durante el proceso de enrollado y desenrollado de soga (recurrentemente tuvimos el problema de que la soga evitaba al carrete y se terminaba enrollando sobre el eje del motor). Además, durante esta etapa encontramos un repuesto para reemplazar nuestras poleas: unas ruedas utilizadas para mover puertas de muebles de pared (como placards), que tenían el mismo costo que las poleas originales (ARS 25) pero ninguno de sus problemas (fueron mucho más estables, casi no tienen fricción, y fueron incluso más cómodos para montar).



Imagen N°36: montaje con nuevas poleas, y el diseño definitivo.

De esta manera, una vez resuelto el problema mecánico de la gestión del ciclo del plato, sólo nos restó concentrarnos en la implementación de la lógica autónoma. Al caso, desde el principio del proyecto tuvimos disponibles algunas versiones preliminares de software de control, pero esperamos hasta cerrar algunas cuestiones constructivas para implementar la lógica de un ciclo de escaneo completo.

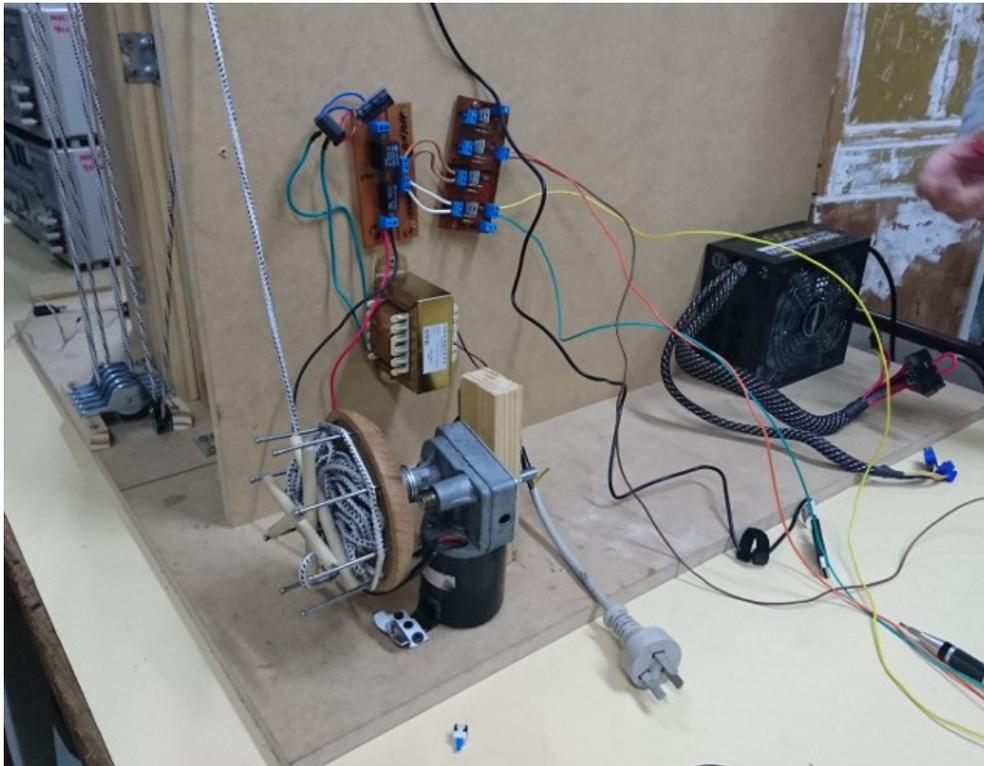


Imagen N°37: setup final, con circuitos lógicos, alimentaciones, y un nuevo carrete improvisado con madera.

La configuración final de hardware y software resultó en un transformador para alimentar el motor de limpiaparabrisas, controlado en su encendido y apagado por un circuito de relés, que también incluye control del sentido de circulación de la corriente mediante relés mismos. Además, una fuente ATX de computadora, que provee líneas de tensión de 5V para los leds que iluminan dentro del plato y de 12V para el motor que ejerce la succión (un motor de secador de pelo hogareño), un pequeño módulo Arduino Uno para implementar la lógica de hardware y el control por señales (para relés, leds, etc) así como también el sensado (fines de carrera en los diferentes momentos del ciclo de escaneo), y el Arduino Uno conectado vía USB a una computadora con GNU/Linux y nuestro software ad-hoc recibiendo y enviando comandos hacia el Arduino. Así, logramos una configuración de menos de ARS 2000, que para los costos actuales de materiales o de cara a otros dispositivos que se encuentran en el mercado (en el orden de los miles de USD), es un gran triunfo para las premisas del proyecto; y todavía puede mejorarse mucho más, dado que este primer prototipo lo logramos apurados tratando de cumplir con los tiempos de nuestras cursadas. Y no solamente es barato, es moderadamente fácil de construir, todas las piezas son fáciles de conseguir, el software está disponible para descargar/modificar por cualquiera, y no hay problemas de

permisos o patentes involucrados que determinen que no se puedan hacerle modificaciones en caso de necesitarlas.

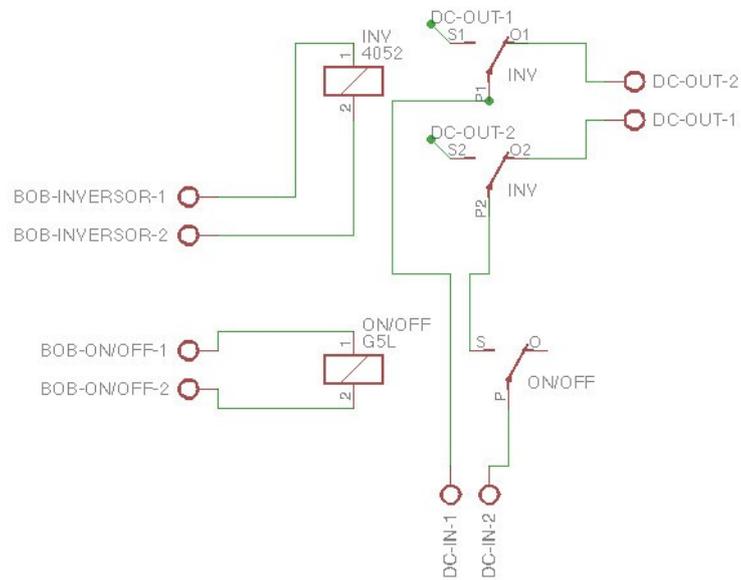


Imagen N°38: esquema de circuito de relés utilizado en el scanner.

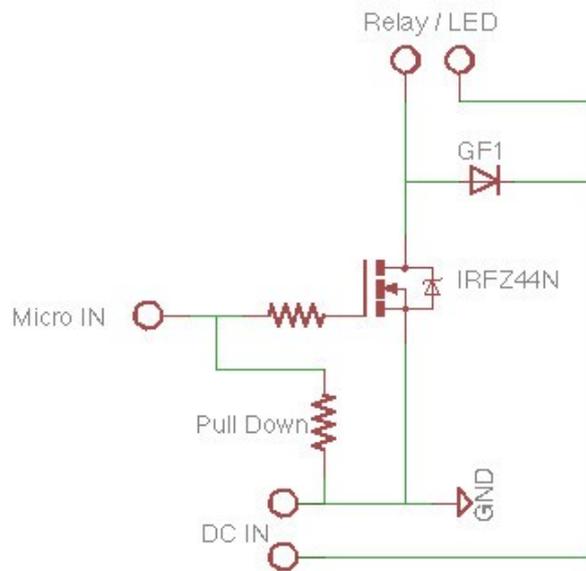


Imagen N°39: esquema de circuito de transistores como puente para control desde el arduino hacia los relés.

Un video grabado de nuestro primer experimento exitoso se puede ver en la siguiente dirección:

<https://www.youtube.com/watch?v=T0yTWxATKE8>

Notas finales y conclusiones

Durante este proyecto realizamos muchas experiencias que no se cuentan en el cuerpo principal de este texto porque consumiría mucho tiempo entrar en el detalle. Pero cabe dejar anotado que hicimos experimentos con decenas de cámaras webs, que hicimos nuestros propios sistemas de succión experimentales antes de optar por el que finalmente elegimos, que en laboratorio hicimos muchos experimentos no mencionados vinculados a la disposición de los componentes del scanner y el diseño general, que tuvimos que recorrer diferentes mercados para poder acceder a respuestas de electrodomésticos o computación que nos permitan pensar estrategias para resolver problemas que nos fuimos cruzando (siempre respetando las premisas del proyecto), y que finalmente fue la complementación entre los diferentes perfiles de los tres integrantes del grupo lo que determinó que lográramos una experiencia exitosa; no hubiera sido posible sin mucho esfuerzo puesto tanto en prueba y error como en diseño, pero tampoco hubiera sido posible si los perfiles de nuestro equipo de trabajo hubieran sido menos interdisciplinarios y/o mixtos.

El primer prototipo que logramos, del cual publicamos el video, carece de varias funcionalidades e implementaciones que debería tener una versión final, a saber: un pequeño ventilador que empuje las páginas luego de la succión, un mecanismo de empaquetado de las imágenes mediante software (en la versión actual las imágenes se escanean pero no se recopilan en un sólo paquete sino que se van sobrescribiendo), un mecanismo de gestión de comandos hacia el scanner más nutrido (actualmente sólo se pueden enviar comandos mediante websockets, y sólo mediante consola javascript), algún mecanismo de test manual (actualmente sólo se pueden detonar ciclos de escaneo con cantidad de páginas, o enviar el plato a una posición estática denominada "home"), es necesario pintarlo de negro (para evitar reflejos), es necesario calibrar las cámaras (que luego de comprobar en laboratorio que tenían una calidad mínima aceptable, no se volvieron a calibrar para la reconstrucción del plato), podría tener un sensor de presión para la detección de errores en el paso de páginas, entre otros detalles que van surgiendo a medida que imaginamos mejoras. Todos estos detalles son importantes para una versión 100% operativa. Pero para nuestro proyecto, de cara a la asignatura Laboratorio 3 y los tiempos de cursada, haber logrado demostrar que el scanner funciona con las premisas del proyecto y teniendo en cuenta que todas las modificaciones que mencionamos no son más que simples horas-hombre de dedicación que nos faltan por cuestiones estrictamente de calendario (y no por imposibilidades tecnológicas o faltas de pericia), damos al proyecto por satisfecho en el mes de Noviembre tal y como planteamos en nuestro cronograma original y cumpliendo el calendario académico.

Nuestras fuentes de información, coherentemente con los lineamientos del proyecto, fueron todas de origen comunitario y/o experimental, con contenidos que provienen de conocimiento libremente distribuido. El corpus más importante de nuestra investigación fueron las experiencias de otras personas encarando nuestros problemas y publicando sus conclusiones, tanto en textos detallados como en formatos multimediales (imágenes y videos). Del mismo modo, nosotros estamos realimentando ese circuito de distribución de información, publicando los detalles de este proyecto, que a su vez se

realimenta con comentarios de las comunidades involucradas en iniciativas similares; desde la publicación de nuestros resultados (el prototipo), diversas personas de la comunidad de hacedores de scanners para libros se comunicaron con nosotros (inclusive quien originalmente implementara el diseño de succión central en el que basamos nuestra versión) tanto para hacer consultas como para compartir sus comentarios al respecto; los mismos tienen acceso a los pormenores de la construcción del dispositivo, como también al código fuente de todas las capas de software del scanner (disponible públicamente en <https://github.com/Canta/book-scanning>). Y este círculo de comunicación ya comenzó recién a un día de publicados los resultados.

Fuentes:

<https://www.diybookscanner.org/> - Comunidad de entusiastas del armado de dispositivos para escanear libros.

<https://www.diybookscanner.org/designs.html> – Colección de diseños.

<https://diybookscanner.org/forum/> - Foro de discusión.

<https://www.youtube.com/watch?v=pkZ8OfgRiq0> – Diseño de scanner automático para libros de Hiroshi Yanasigawa, implementando la estrategia de succión central que nosotros también utilizamos.

<http://tges.blog117.fc2.com/blog-date-20141220.html> – Algunos textos y planos del señor Yanasigawa (en japonés).

https://www.youtube.com/results?search_query=auto+book+scanner – Incontables demostraciones en video de diferentes estrategias para el armado de dispositivos y solución de problemas puntuales.

<http://www.scantips.com/> - Sitio dedicado a detalles técnicos puntuales sobre el proceso de escaneo y los pormenores de la digitalización de imágenes.

<http://www.dofmaster.com/equations.html> – Ecuaciones de cálculo visual, utilizadas durante el proceso de experimentación y calibración de cámaras.

<https://www.arduino.cc/en/Reference/HomePage> – Documentación de referencia de Arduino, para la gestión de señales de hardware y comunicación.

<https://pypi.python.org/pypi> – Repositorio de paquetes de Python, para la búsqueda de librerías que ya implementen funcionalidades que necesitamos (websockets, webserver, serial, etc).

<https://developer.mozilla.org/en-US/> - Mozilla Developer Network (MDN) para la gestión de código web.