



**UNIVERSIDAD NACIONAL
DEL LITORAL**

**Facultad de Ingeniería
y Ciencias Hídricas**

Trabajo Final de Carrera

Ge.Li.B.A.

Diciembre 2023

UNL

FICH

**Facultad de Ingeniería
y Ciencias Hídricas**

UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas
Tecnicatura Universitaria en Software Libre

Título del proyecto

Proyecto Ge.Li.B.A.

Generador Libre de Bibliotecas Automatizadas

Autor/a

Hugo Ignacio Boulocq

Trayecto Programación

Prefacio

En las próximas páginas voy a explicar procesos y situaciones que me llevaron por este derrotero maravilloso que fue desechar una aplicación con bases de datos y marco de trabajo preparado en otras materias y decantarme por algo menos pretencioso y simplista. Si tuviera, como dicen algunos prefacios que pude haber leído sobre guías, instar que se salten algunos puntos, podría arriesgar a saltarse todos y resumir todo en estos dos puntos definiendo qué es Ge.Li.B.A.:

1) Tus libros para compartir en una web. Libre. Sencillo. Práctico.

2) Un *script* de Python te copia tus libros a una carpeta y te genera una página web.

Fin :)

El desarrollo es un poco más extenso que la descripción, claro está. Voy a tratar de describir de qué va todo esto.

Como en el Aikido y estimo que sucede en varias disciplinas donde prima la técnica, en mi caso el trabajo fue interno y de evolución, para (de alguna manera) aprender la forma de realizar el objetivo (la técnica, es decir, el código), ir adaptándolo para luego salirse un poco de lo estrictamente establecido, pero sin perder de vista que primero deben internalizarse conceptos que fui adquiriendo a lo largo de la Tecnicatura, la cual comencé en el último tramo del 2015 y que por diversos motivos personales hube de dejar "madurar"(me) por algún tiempo. Con ello el aprendizaje del código en sí.

Como cualquier obra, partiendo de la breve experiencia editorial que tuve durante varios años, considero interesante y productivo para el trabajo dejarlo evolucionar durante un tiempo y volver a retomar. ¿Cómo? Así: trabajar con ahínco durante un primer período (durante la cursada) y dejarlo guardado en un cajón virtual cierto tiempo, para volver a retomarlo tiempo más tarde. Varias veces lo hice y puedo asegurar que tiene resultados provechosos, espero esta ocasión haya resultado así. En cuanto a la técnica se refiere, podría decir que en este caso sucedió lo mismo. Dejé reposar el mismo desde noviembre de 2022 y lo he retomado a principios de 2023. Cabe aclarar que el cronograma realizado previamente en trabajos prácticos presentados puede llegar a diferir.

Cambiando de tema y aproximándome a el trabajo realizado, un ejemplo de lo que intento describir puede apreciarse en este texto:

https://gitlab.com/bicivoladora/tusl-trabajo-final/-/blob/master/Vivas_libres_federadas.pdf

Página 15: *La cualidad libre aparece vinculada a algo precario o casero.*

Este caso pretende acercarse a lo casero. Prejuicioso, durante muchos años asocié los trabajos (los trabajos grandes quiero decir, aplicaciones de envergadura) que no tuvieran una empresa detrás o con muchos ojos de corrección atento a los errores con algo que no fuera a funcionar, o que podía llegar a esconder segundas intenciones. Por ende tendía a globalizar, también en forma errónea, que las pequeñas producciones sin demasiadas pretensiones tuvieran escasa utilidad. Este trabajo final y la tecnicatura toda me acercó a mirar desde otra perspectiva esto, a acercarme a los repositorios de otros y ver qué hacían, clonarlos, aprender, apropiarme en intentar remezclar y por sobre todas las cosas, abrazar lo casero. De hecho este proyecto fue hecho con esta intención, que sea lo más casero posible, lejos de proyectar "*profesionalismo*" en el resultado y más cerca de esas obras de arte de antaño, permítaseme controversial ejemplo, usando software privativo y que vulneraban ciertas restricciones como contrapartida a esos limitantes, como eran lo *cracks* que tenían música de fondo al estilo *chiptune*, con un esfuerzo pocas veces valorado en lo creativo.

Vivimos épocas muy interesantes en lo que a tecnología y sociedad se refiere. El Software Libre, a mi humilde modo de ver, poco tiene que ver con ciertas situaciones que nos alejan del código y de los programas. Social sí, pero sin código, no. Volviendo al principio de esta reflexión, de la misma manera que el Aikido, sin bien "arte de la paz" como también se lo conoce, no deja de ser un arte marcial.

Más allá de esta breve presentación y para ir concluyendo la propuesta, separando la introducción formar, es preciso decir que la idea del proyecto se arrimaría a un llano: "tomá este trabajo deseando fervientemente que pueda ser de utilidad, que pueda ser modificado, remezclado y retransformado. En el medio el cómo llega a esa persona (ese otro) desde este instante hasta que *el otro* lo encuentra en la maraña de internet. Bueno, en parte para esto está la tecnicatura también. Del mismo modo que yo pude acceder al repositorio de trabajos finales y conocer otros proyectos, lo mismo (arguyo) puede llegar a suceder con este. *No pasa nada,*

clonalo con tranquilidad, debería decir. Instalalo, te aseguro que no vas a romper nada. Es muy sencillo, de hecho te crea una web con la carpeta de tus libros, eso es todo. La finalidad, insisto, es que pueda ser utilizado como mejor le(s) parezca, yo me comprometo a ser lo más claro posible en cuanto a su utilidad.

Notas de noviembre 2023:

Después de las correcciones y sugerencias para este trabajo realizadas por el profesor Juan Pablo Taulamet, en lo que se refiere a la dirección del proyecto, es decir, apuntar a equipos carentes de conexiones a internet, he preferido tachar y permanecer el texto que hace referencia a “internet” en lugar de borrarlo, una forma de mostrar “viva” la continua evolución del mismo.

Introducción

¿Por qué realizar un trabajo de esta índole? En principio puedo decir que porque no encontré algo que pudiera adaptarse al entorno casero, pequeño. Vivimos en un mundo donde todo se piensa, sobre todo lo tecnológico, hacia ámbitos grandes, empresariales, con metodologías ágiles de trabajo y tiempos cortos y urgentes. Por ello se desarrollan marcos de trabajo (del inglés *frameworks*) para que puedan ser sostenidos por un equipo de desarrollo y todo lo demás viene aparejado para desembocar finalmente en esas perspectivas. Este proyecto intenta ir hacia el lado opuesto y no motiva la urgencia de nada en absoluto. Si hablamos de vulnerabilidades a nivel infraestructura requiere lo mínimo indispensable para poder operar, diferenciándose enormemente de los requerimientos de estos marcos mencionados que necesitan de bases de datos y servidores tendientes a la nube para operar, como se estila el día de hoy. Geliba apuesta por la computación autogestionada y de bajo consumo, los sitios estáticos sin demasiados abalorios, la transmisión de datos de pares y lo comunal.

Motivaciones personales de la propuesta

Mi principal motivación es mi pasión por la lectura y los libros, sumado a mi entusiasmo por la internet de baja tecnología, horizontal y de escasa trascendencia, es decir, que con este proyecto no apunto a estar en los resultados del buscador Google. A su vez, también soy entusiasta de la recuperación de equipos viejos con distribuciones de GNU/Linux “ligeras”, que permiten de este modo darle una segunda vida a los equipos que por algunas compañías de software son considerados obsoletos. Esto último motiva mi idea de estar en contra de la obsolescencia programada de las computadoras personales por sobre todos los equipos, y aparejado a algún esbozo que pueda hacer con algunas líneas de código en Python desembocaron en este desarrollo para ofrecer un generador simple que haga todo el trabajo de hacer un sitio web de una página con los enlaces de descarga incluidos, directos y sin tener que pasar por ninguna publicidad o esperas interminables, como se estila en muchos lugares de descargas.

Desde ya que este trabajo puede ser mejorado, dista de estar finalizado y es por ello que el repositorio está disponible para que alguien más avezado pueda usarlo para lo que se le

antoje, estudiarlo y modificarlo, compartirlo y/o distribuir copias mejoradas, cumpliendo así con las 4 libertades del Software Libre.

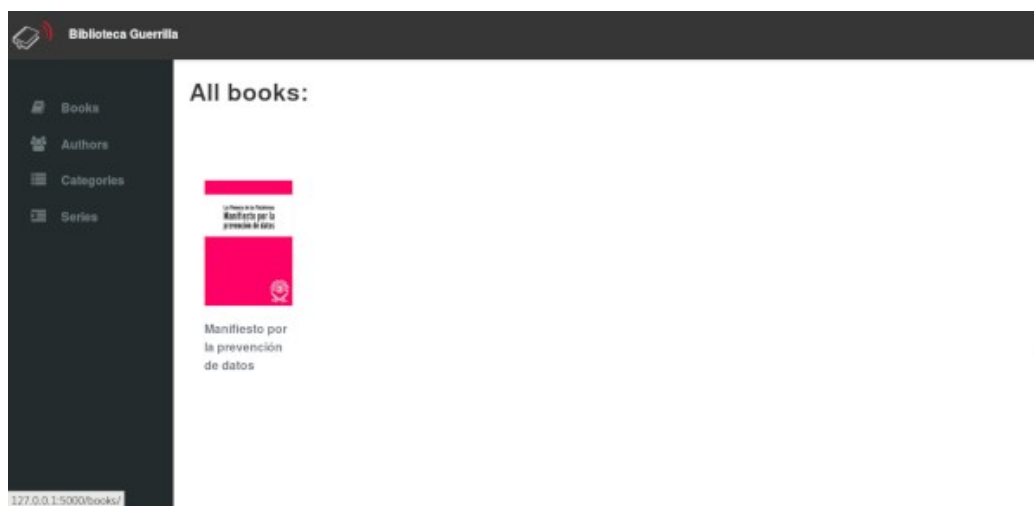
Estado del Arte

Los antecedentes a los que aquí se van a hacer referencia tienen que ver con 4 proyectos disponibles en internet. Veremos a continuación, en base a los datos recabados y la información provista, si con estas herramientas a las que pude acceder en base a búsquedas en la web, era posible fácilmente llegar a un objetivo acorde al proyecto, esto es, ofrecer una biblioteca simple digital generada por un usuario (o similar) sin demasiados conocimientos sobre el manejo de una consola de GNU/Linux o sobre desarrollo orientado a páginas web. Cabe aclarar que si bien los principios técnicos de estos proyectos no son completamente iguales, ofrecen cierto resultado que se acerca al objetivo de la pregunta de investigación, en la cual se trata de ofrecer sin pasos complejos un listado de libros digitales que pueda ser descargado por alguien más, en cualesquiera que sean sus condiciones, es decir, desde una computadora personal conectada a internet, desde un teléfono celular conectado a una red lan, etc.

Las herramientas analizadas como antecedentes son las siguientes:

[Biblioteca Guerrilla](#)

Este proyecto genera a partir del programa *Calibre*, una aplicación de escritorio gestora de bibliotecas digitales, un catálogo web de los libros que el usuario haya preseleccionado en este programa. En principio hemos de decir que este sistema no presenta mayores inconvenientes para generar el sitio, siempre y cuando se posea algún conocimiento sobre clonar repositorios y modificar algún código extra, además de tener organizada correctamente la biblioteca en Calibre con sus tapas correspondientes.



Bibliobox

Ofrece un paso más allá de la anterior Biblioteca Guerrilla. Se trata de generar el resultado tal cual se menciona en el instructivo del anterior proyecto, agregando la posibilidad de “embeberlo” en una Raspberry Pi (en el instructivo sugiere a partir de su versión número 3) y generar a partir de un script una conexión de red lan disponible, lo cual implica portabilidad, siempre y cuando dispongamos de una celda de carga, claro está, u otro dispositivo portable generador de la energía requerida para operar.



Letras Viajeras

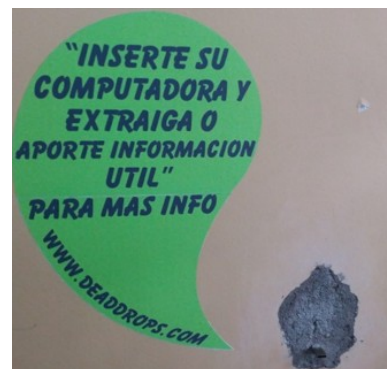
Como reza en su repositorio de internet, es una iniciativa de la *Dirección de Bibliotecas y Promoción de la Lectura del Gobierno de la Provincia de Buenos Aires*, que permitirá a los pasajeros de larga y media distancia tener acceso gratuito a libros digitales.

Es un trabajo realizado por la Cooperativa de Software Libre Gcoop, argumentando que “*La solución consiste en un Router Inalámbrico al que se le instala OpenWRT, un Firmware libre basado en el kernel Linux*”.



Dead Drops

Este proyecto, creado en 2010 por el artista alemán Aram Bartholl, fomenta el intercambio de archivos entre usuarios anónimos sin necesidad de conectarse a Internet (un P2P fuera de línea). El objetivo es muy simple: empotrar un *pendrive* en la pared dejando el conector visible, de manera tal que cualquiera con una conexión usb pueda acceder al mismo con algún dispositivo que posibilite esta conexión.



Nota de referencia:

https://www.bbc.com/mundo/noticias/2016/04/160331_tecnologia_proyecto_dead_drops_usb_muros_red_offline_aram_bartholl_lb

Este es el proyecto más elemental de todos los citados y desde el cual surgió la motivación para este trabajo, la cual fue derivando a través de búsquedas en los demás arriba enumerados.

Descripción del proyecto

¿Qué es?

Con una aplicación de escritorio hecha en *Python* con *Tkinter*, la idea del proyecto es generar, con unos simples pasos, el listado de la carpeta con la biblioteca digital que el usuario elija (en PDF, Epub, etc) y compartirla en una página web estática, anti obsolescente, es decir, que se pueda “levantar” en cualquier computadora con un servidor web instalado como Apache, Nginx, Hiawatha, etc.

El resultado final es un archivo *index.html* (con un diseño minimalista) y la copia del contenido de la carpeta elegida a un directorio en la misma ubicación que el *index* mencionado. El *script* creará una carpeta que se llamará “libros” y contendrá todo lo que la aplicación de Python haya leído desde la ubicación que se haya elegido como origen de los documentos.

Luego de ese resultado, el usuario podrá decidir qué hacer con ello, como se detallará más abajo.

¿Por qué automatizada y qué idea o ideas persigue el proyecto?

Porque con un *script* visual, que no deja de ser un archivo con unas pocas líneas de código, intenta generar un documento HTML leyendo una ruta que el usuario asigne sin que requiera mucho esfuerzo por parte de éste. El proyecto está a favor de una computación de bajo consumo, ya que podría desplegarse en una simple Raspberry Pi instalada en su hogar. Si no se cuenta con un equipo de este estilo puede ser un CPU, si bien no de bajo consumo, de los considerados “obsoletos”, como por ejemplo un Athlon 2500 de más de 10 años de antigüedad. La inspiración sobre estos temas surge en parte de la lectura de estos artículos escritos por miembros del grupo en internet denominado “Cybercirujas”.

<https://cybercirujas.sutty.nl/textos/2022/05/24/por-qu%C3%A9-peleamos-por-una-computaci%C3%B3n-de-bajo-consumo.html>

¿A quién va dirigido y qué busca solucionar?

Está dirigido principalmente a usuarios que carecen de conocimientos técnicos pero que quisieran poder ofrecer su biblioteca de libros digitales ~~en internet~~ en una intranet para poder ser compartida y descargados los archivos libremente.

Este trabajo ofrece una solución sencilla pero efectiva a la creación de una página web con elementos para descargar sin tener que pasar por tediosos tiempos de espera ni dudosos sitios de descarga, apuntando a compartir en círculos pequeños, comunidades amantes de la lectura (propósito primigenio) y por qué no, motivar a que se pueda mejorar dirigiendo el resultado hacia otros horizontes, de cualquier índole o necesidad.

Objetivos

1) Promover un mayor uso de software libre mediante el desarrollo de una aplicación, cumpliendo con las 4 libertades del mismo e instando a que pueda mejorarse de la manera que se lo requiera.

2) Motivar el uso de un equipo antiguo como servidor de este sitio web, favoreciendo así también el *autohosteo* y ~~la internet~~ los accesos de baja tecnología.

3) Alentar y fomentar la lectura en sus diferentes vertientes, en este caso la digital, compartiendo archivos.

4) Compartir documentos con licencias libres.

5) Ofrecer una descarga directa sin publicidades y tiempos de espera tediosos.

Qué quiere decir *autohosteo* e internet de baja tecnología

A medida que internet fue creciendo la demanda de recursos se hizo cada vez mayor. Si en los albores de la red la una página web apenas demandaba un simple html, que no es más que un archivo de texto con algunos hipervínculos y algunas imágenes, al día de hoy la precarga con marcos de trabajo, bases de datos buscando información, cookies, usuarios, claves, fotos y videos de alta resolución, etc. hacen insostenible una conexión austera para poder navegarlas. Las páginas web de baja tecnología, junto con los generadores de sitios estáticos (programas

que facilitan la creación de un entorno web sin casi requerir tener conocimientos de HTML, CSS y Javascript) pretenden acercarse a ese estilo de antaño con páginas que no poseen acceso a bases de datos, imposibilitando la inyección de código muy en boga en estos días y permitiendo cargar el sitio prácticamente en cualquier parte por los bajos requerimientos del mismo.

El *autohosteo* no es más que la posibilidad de ofrecer el sitio web en su propio equipo si así se lo dispone, favoreciendo de este modo una internet autogenerada. Siempre que esté disponible en el hogar una dirección ip pública, a través del proveedor de internet, es posible *autohostear* un sitio web en algún equipo que pueda permanecer encendido el tiempo que requiera el sitio estar en línea. Los equipos Raspberry Pi son propicios para esta actividad por su bajo consumo energético. Son pequeñas computadoras de una sola placa pero con gran rendimiento que posibilitan el alojamiento de nuestros sitios.

Una información introductoria mucho más elaborada acerca de las páginas web de baja tecnología puede leerse en Low Tech Magazine, en esta nota:

<https://solar.lowtechmagazine.com/es/2019/11/how-to-build-a-lowtech-website.html>

Principales desafíos

El resultado de este trabajo, como bien se ha hecho mención desde el principio, es intentar acercar una pequeña y muy humilde colaboración al Software Libre, brindando una idea en forma de *script* que solucione un inconveniente puntual, ya sea para quien carece del conocimiento para resolverlo, o para alguien que se encuentre incursionando en desarrollo y quiera ver de qué va este trabajo, o simplemente se ha topado con este proyecto y lo quiere probar, para que pueda listar sus documentos (los cuales se sugiere fervientemente que **posean licencias libres** para evitar cualquier tipo de futuros inconvenientes, reclamos, etc).

Existen antecedentes como los que se han listado aquí de gran calidad y con resultados tecnológicamente mejores. Este proyecto no busca competir con ellos, si no ofrecer una alternativa mínima, sencilla y aplicable para cualquier tipo de usuario.

Creo, para concluir, que responde a la pregunta inicial que ofició de disparador: que cualquier persona pueda usarlo y con muy poco pueda lograr un resultado práctico, rápido y que le permita hacerlo disponible en diferentes entornos.

Análisis de la necesidad – características del proyecto

Me considero un lector de diversos formatos y ello implica poder encontrar material en webs para poder descargar.

Usualmente existen sitios de descarga que ofrecen libros de autores conocidos, con licencias restrictivas, pero no de noveles con licencias libres.

La necesidad surge de simplificar un proyecto en el que me encuentro trabajando, “*Biblioteca EnlaceLibre*”, donde se pretende aunar libros y revistas sin restricciones de autor o con las licencias libres mencionadas, tanto para su descarga como para su reedición. Este sitio requiere de un registro por parte del usuario y una posterior subida del material, uno a uno. Esto, si bien no tiene complejidad alguna exceptuando el tiempo, para el usuario sin demasiadas pretensiones que solo desea compartir sus libros puede llegar a ser tedioso o aburrido. Además de la idea de fomentar los bajos recursos y por qué no, el uso de del sitio generado en alguna red local sin salida a internet, oficiaron de disparador para llevar a cabo el proyecto.

El problema durante el desarrollo de Ge.Li.B.A. que se me presentó fue el siguiente. Una vez que había desarrollado la aplicación que generaba el *index.html* y reunía la carpeta con los libros me pregunté: bueno, tiene la página y la carpeta... ¿Y ahora? Tiene que tener al menos un lugar para subir por el momento, después verá qué quiere hacer además de eso. Así fue como surgió la alternativa de subida al sitio de Ge.Li.B.A. por [FTP](#).

La idea era más que clara: mantenerlo simple para el usuario y que el programa lo hiciera todo.

Conectar con [EnlaceLibre](#) era la primera opción, es decir, ofrecer un espacio dentro el mismo sitio web para que el usuario subiera su material para compartir. Este proyecto fue desarrollado en Django, un marco de trabajo de Python para la materia Programación 2, de la profesora Ana Laura Almada. En principio estaba hecho en Laravel, un marco de PHP. Sin

detenernos demasiado en esos pormenores técnicos, mi intención era expresar con esto que existía un proyecto anterior que pudo haber sido presentado como Proyecto Final, pero fue descartado o desplazado, mejor dicho, por GeLiBA, por considerar el alcance más ameno a usuarios de cualquier índole o característica técnica.

Las dificultades que se me presentaban intentando migrar los archivos del usuario que hubiera generado eran las siguientes:

1) Generar un modelo nuevo de tablas en la base de datos de EnlaceLibre que implicara obviar algunos datos fundamentales a los cuales estaba orientada la página, como el autor, el título y una imagen de la tapa.

2) Suponiendo que ello fuera sorteado con datos y enlaces a imágenes genéricas, debía de generar además un usuario y clave para que posteriormente se pudiera modificar o bien por el usuario o por alguien más si este no lo desease. Otro incordio.

3) Otro dato no menor era generar una página aparte, donde se recibieran todos los libros precargados desde la carpeta generada y prepararlos para la subida. ¿Qué sentido tendría que el interesado tuviera que volver a seleccionar la carpeta para subir sus libros? Un paso doble que resultaba pesado e innecesario, además de que no había alcanzado el conocimiento técnico para generar esa página con los libros precargados.

En definitiva, después de varias pruebas realizadas, la opción *EnlaceLibre* quedaba descartada por complicada.

La siguiente alternativa era subir los libros a un espacio en la Raspberry Pi que poseo en mi casa, la cual oficia de servidor web para algunos sitios que tengo. El obstáculo a salvar era el siguiente para este caso:

-Que el espacio pudiera ser accedido de forma anónima.

-Que la ruta de acceso imposibilitara escalar, es decir, que no pudiera ir por ejemplo a */etc* o */home/usuario*.

En primer lugar probé con el programa *vsftpd* pero no logré dar en el clavo con las restricciones de usuario. La siguiente prueba la efectué con *proftpd* con mejores resultados. La idea era “enjaular”, como se conoce en la jerga técnica a la limitación de la ruta, un espacio

para que el usuario pudiera subir su carpeta con los libros más el HTML que oficiaría de acceso a éstos, o sea, lo que hubiera creado Ge.Li.B.A. Para ello había que editar el */etc/proftpd/proftpd.conf*

```
# Use this to jail all users in their homes
```

```
DefaultRoot /var/www/html/geliba/libros_externos
```

Y establecer un usuario anónimo:

```
#Configuracion personal
```

```
<Limit LOGIN>
```

```
DenyAll
```

```
</Limit>
```

```
<Anonymous /var/www/html/geliba/libros_externos>
```

```
User ftp
```

```
UserAlias anonymous ftp
```

```
RequireValidShell off
```

```
<Limit LOGIN>
```

```
AllowAll
```

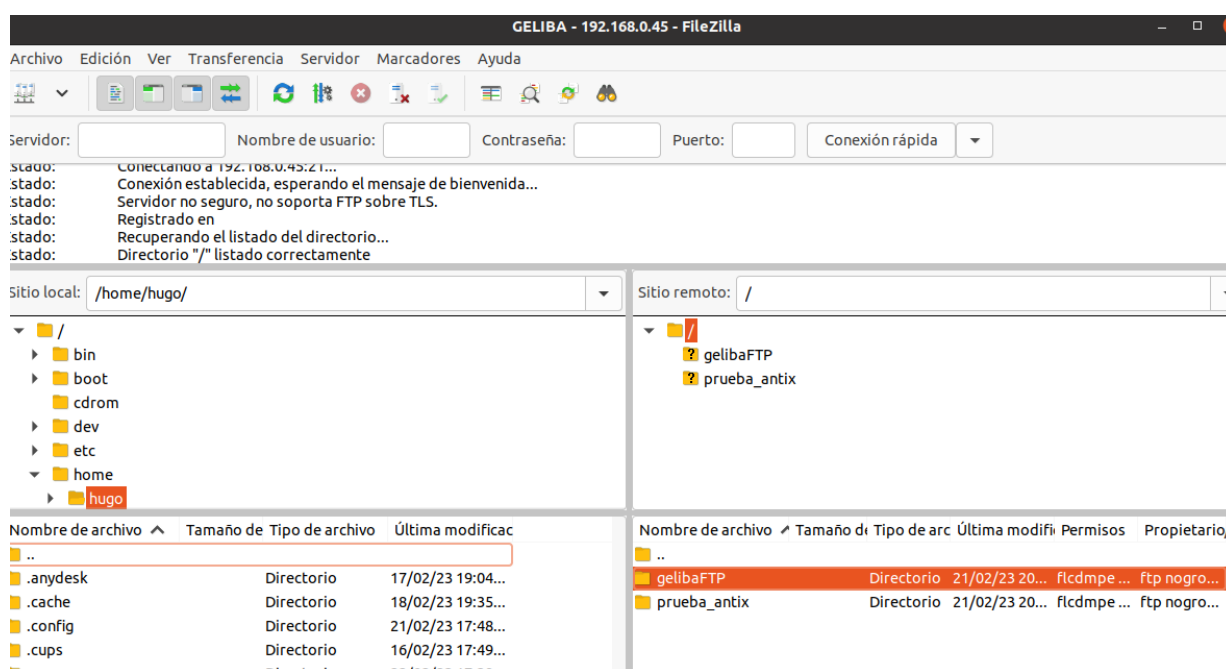
```
</Limit>
```

```
</Anonymous>
```

Con esto configurado, ya se podrían empezar a subir los contenidos. Ahora solo quedaba resolver cómo realizarlo desde la misma aplicación y con algunas (mínimas) preguntas al interesado.

Primero que nada, esto debería ser opcional, no todos iban a querer compartir sus contenidos en una web. Para ello tuve que decidir algunas cosas. Inicialmente debería aparecer un cartel que consultara si deseaba subir el contenido por FTP a Geliba. En caso afirmativo, se solicitaría que se escribiese un usuario o nombre, para que la ruta quedara de la siguiente manera:

https://geliba.enlacepilar.com.ar/libros_externos/gelibaFTP/ (este sitio de ejemplo se encuentra funcionando, se puede seguir el enlace).



(imagen de ejemplo del FTP funcionando con Filezilla)

Acto seguido el complemento `ftplib` de Python haría su magia: subir todo el contenido con algunas líneas de código, crear carpetas, y subidas explorando el contenido, rehaciendo en silencio todo lo realizado previamente, pero sin que el usuario tuviera que hacer esa tediosa operación 2 veces...

Para ello se aparece una ventana diciendo: “¿Querés subir los libros a GeLiBA?”. Si la respuesta es afirmativa, llama a la función `subir_ftp()` y empieza el proceso de leer la carpeta libros y volcarla en un directorio cuyo nombre debe proveer el usuario, indicándolo en un cuadro de diálogo emergente que solicita el nombre.

Después de la copia es posible ver el resultado final abriendo automáticamente el navegador.

Sitio de prueba de Geliba

Modelo web del listado de libros para compartir

Total de libros: 6	
Presioná CTRL + F para buscar	
#	Nombre
1	Los Cipreses del Cementerio Britanico - Hugo Enrique Boulocq
2	Enroque en la ventana 1987 - Hugo Enrique Boulocq
3	Apuntes sobre el Cuento - edicion digital 2020

Licencia

La parte de desarrollo del software está licenciado bajo [GPL3](#); para el contenido web donde se encuentra disponible el archivo de descarga, la licencia es la siguiente: [Licencia de Producción de Pares](#).

Publicación

El trabajo posee su repositorio en Gitlab, el cual puede ser clonado libremente desde:

<https://gitlab.com/enlacepilar/generador-libre-de-bibliotecas-automatizadas>

Geliba tiene además un sitio web alojado en una Raspberry Pi hogareña, promoviendo de este modo el autohosteo, el cual se encuentra disponible accediendo a:

<https://geliba.enlacepilar.com.ar/>

Y un espejo por alguna eventual falla, en:

<https://geliba.netlify.app/>



vie 13 enero 2023 - categoria: [Galería](#)



ANEXO: GALERÍA DE FOTOS CON GELIBA

Generando una galería de fotos con GelLiBA

Etiquetas: [galeria](#) [fotos](#) [imagenes](#)

dom 25 diciembre 2022 - categoria: [FTP](#)



SUBIDA OPCIONAL POR FTP

Subir por FTP tu proyecto a Geliba

Etiquetas: [ftp](#) [subida](#)



```
geliba.py - /home/hugo/Escritorio/Enlace a Trabajo Final/autoinst...
File Edit Format Run Options Window Help
from tkinter import *
import os
from tkinter import messagebox

#colores
letra_geliba="#1E434C"
#fondo ="#C99E10"
fondo = "#EBEBEB"
#subtitulo_fondo="#9B4F0F"
subtitulo_fondo = "#1F8A98"
boton_carpeta="#34C3D5"
letra_carpeta="black"
boton_ayuda="#78D7E3"
letra_ayuda="white"
blanco="white"

caja = Tk()

caja.title("Instalador")
caja.geometry('400x280')
caja.config(background=fondo)
caja.resizable(False, False)

texto0 = Label(caja, text="Ge.Li.B.A.", font=('Tahoma', 18), height=2, fg=letra_
texto0.grid()

texto1 = Label(caja, text="¡Generador Libre de Bibliotecas Automatizadas \n (ant
texto1.grid()

texto2 = Label(caja, text="Hacer clic en el botón para generar el archivo", back
texto2.grid()

### *** FUNCIÓN ABRE HTML ***
def mira_HTML ():
    import webbrowser

    # open html file

Ln: 1 Col: 0
```

The image shows a Python Tkinter application window titled 'geliba.py'. The code defines a window 'caja' with a title 'Instalador' and a geometry of 400x280. It sets a background color and is non-resizable. The window contains three labels: 'Ge.Li.B.A.', '¡Generador Libre de Bibliotecas Automatizadas \n (ant', and 'Hacer clic en el botón para generar el archivo'. A function 'mira_HTML' is defined to open an HTML file. A modal dialog box titled 'Instalador' is overlaid on the window, displaying the text 'Ge.Li.B.A.', '¡Generador Libre de Bibliotecas Automatizadas (anti obsolescentes)!', and 'Hacer clic en el botón para generar el archivo'. The dialog has two buttons: 'Seleccionar carpeta y generar HTML' and 'Ayuda'.

Objetivos alcanzados y problemas sorteados

En el camino de la práctica uno aprende lo que mil teorías no pueden enseñar. Y vaya si uno aprende. Durante este trayecto incorporé a mi saber no solo algunas vertientes del código de Python que me eran desconocidas, si no también del generador de sitios estáticos Pelican y cómo modificar los temas visuales preestablecidos y una de las cosas más interesantes de éste, cómo crear uno propio.

De cómo la idea de un pequeño espacio web sin pretensiones derivó en un estructura con tema propio

El mundo de los generadores de sitios estáticos me resultó al principio irrelevante, después extraño y al final (cuando me encuentro escribiendo estas líneas) apasionante, a tal punto de dedicarle muchas horas al día para crear uno propio para el sitio de Geliba. Pero vayamos por partes. Empezando por los generadores de sitios estáticos, la idea que establecen es la de volver a la raíz de los primeros sitios web de la primera etapa de la World Wide Web, esquivando diseños con cargas excesivas muy comunes hoy, solventado gracias a que los proveedores de servicios de internet ofrecen un servicio muy distinto al que se podía llegar a poseer a finales de la década del 90. Muchos megas para procesar muchas cargas de datos, videos a 4k, más de una computadora familiar, celulares, tabletas, todo en un mismo hogar. Tienen como premisa evitar bases de datos y de este modo son más seguros, ya que los ataques en su mayoría van dirigidos a ese destino. Es verdad, por otro lado, que se sacrifica cierto dinamismo con esta parte, pero el resultado lo vale.

Impacto de la implementación y observaciones finales

Creo que el proyecto nunca podría ser considerado como concluido, y además de los testeos realizados por mi compañero Walter, la idea es someterlo a prueba en diversos foros para comprobar si verdaderamente resulta de interés esta iniciativa o serían necesarias algunas modificaciones, o cambiar radicalmente de perspectiva si fuese necesario. Este paso claramente requiere de un proceso prolongado, por lo cual considero que el tiempo dará una prueba concluyente acerca de si puede llegar a ser de utilidad o no. Si al menos a una persona pudiera servirle para cumplir con sus objetivos, ya sea una web por afición, o una lista de libros para compartir en una Lan escolar, o en una organización, me consideraré sumamente afortunado.

Anexo

¡Geliba como generador de galería de fotos!

Un script que no pueda ser transformado y adaptado por otro a sus necesidades, discúlpeme mi transgresión, no sirve para nada. Acá les transmito una inquietud que plasmé directamente en un archivo PY. ¿Si en vez de subir libros hacemos unas pequeñas, pequeñísimas modificaciones al código y en vez de leer PDF, ODT, TXT, leyerá JPG o JPEG, PNG y BMP? Y si en vez de generar un listado de texto, junto con el complemento o biblioteca PIL de Python generásemos miniaturas y transformásemos la página en una galería de imágenes? Al momento de imaginarlo me sonó interesante y con unos retoques en el código original pude obtener un resultado bastante aceptable para lo que buscaba.

La idea consiste en generar una galería con fotos en miniatura en una tabla, con enlace a las imágenes de tamaño real para visualizar al hacer clic y de este modo reducir la carga de la página.



Anexo II

Durante agosto de 2023 realicé una serie de cambios en el *script* de GeLiBA con la intención de simplificar el contenido y “modularizar” determinados aspectos internos del código con el fin de que sea legible y proyectando que alguien más pueda adaptarlo a sus necesidades sin demasiadas complicaciones. También he desechado determinadas funciones que solo prestaban a confusión, optando porque el fin sea lo más claro posible: crear un sitio web con los documentos elegidos y levantarlos en una Red de Área Local (LAN).

La función subida por FTP al sitio que he creado como informativo fue eliminada para evitar ambigüedades en el destino del resultado generado.

Además de ello, agregué el conocido marco de trabajo Bootstrap, en su versión 4, para agilizar y mejorar cuestiones de diseño y Datatables, una herramienta escrita en Javascript que agiliza la búsqueda y despliegue de los resultados con interesantes funcionalidades.

Modifiqué el archivo de ayuda de Geliba que se encontraba desactualizado, por este texto:

Este menú de ayuda pretende, aunque sea someramente, guiarte por el breve y sencillo proceso de generar tu listado de libros para compartir en una página web estática, anti obsolescente y que pueda ser servida en cualquier computadora.

Geliba está pensado principalmente para generar un sitio web y, haciendo uso previo del script instalador, alojar con Apache el sitio web en una red local.

¿Cómo empiezo?

- 1) *Seleccioná la carpeta donde se encuentren tus documentos para compartir.*
- 2) *Escribí un título y una breve descripción.*
- 3) *Se va a abrir tu navegador con el resultado.*
- 4) *En la carpeta vas a tener un documento "index.html" que es el sitio web, una carpeta "estáticos" con los complementos del sitio y una carpeta "libros", donde se copiaron los documentos que habías seleccionado previamente.*

5) Copiá estos 3 elementos a un pendrive si así lo deseás o lleválos (si es esta PC la que vas a usar) a `"/var/www/html"` en esta misma computadora. Si no tenés nada allí pero existe un `"index.html"`, reemplazalo (sobrescribir), es el que coloca por defecto Apache para asegurarse de que todo anduvo bien al instalarlo

6) Ahora abrí el navegador y en la barra de direcciones tipeá `"localhost"`. Debería aparecer el sitio de GeLiBA :)

¿Posibles conflictos de derechos?

Si por algún motivo no te deja ver la página o descargar los libros, vamos a asegurarnos de darle derechos de escritura a la carpeta `"html"`. Para ellos nos paramos en `/var/www` y abrimos la consola con el botón derecho del mouse: abrir una terminal o simplemente abrimos la terminal y tipeamos `"cd /var/www"`.

Allí le daremos derechos a la carpeta `755` (lecto-escritura y ejecución para propietario, lectura y ejecución para grupo e invitados)

```
"sudo chmod -R 755 html/"
```

Quizás el tema más complejo que se me presentaba quizás sería cómo saber si la PC sin internet que iba a ofrecer el sitio tuviera instalado Apache... pero... ¿qué es Apache?

Apache es un software que oficia de "servidor" de páginas web que al instalarse acepta en esa PC las peticiones de terceros para tu dirección ip (en este caso la ip local de tu máquina), accediendo desde el navegador, y que más de uno pueda hacerlo en simultaneo.