**Ace Books**

**Alejandro Leal Bermúdez**

IES Vega de Mijas

2018/2019

**1. Introducción.**

Actualmente hay distintas páginas web que intentan ofrecer un servicio de lectura de documentos, sin embargo, todas estas se quedan cortas, sea por no soportar varios tipos de documentos, carecer de la posibilidad de organizar tus documentos e incluso de no tener un buen lector de documentos.

Ace Books te da la capacidad de leer tus archivos pdf, organizar estos en distintos directorios e incluso interactuar con ellos.

**2. Descripción del proyecto.**

**2.1 Objetivos.**

Los objetivos del proyecto son los siguiente:

* Ofrecer al usuario las herramientas necesarias para organizar sus documentos en colecciones.
* Poder personalizar los documentos y colecciones (nombre, descripción y portada).
* Un lector de documentos.
* Gestionar tu información (nombre usuario, imagen de perfil, contraseña e idioma de la web).

**2.2 Requisitos mínimos y restricciones.**

Los usuarios tendrán que registrarse para usar los distintos servicios que son ofrecidos. Se necesitará acceso a internet y un navegador moderno.

**3.Métodos y herramientas de desarrollo.**

**3.1 Hardware.**

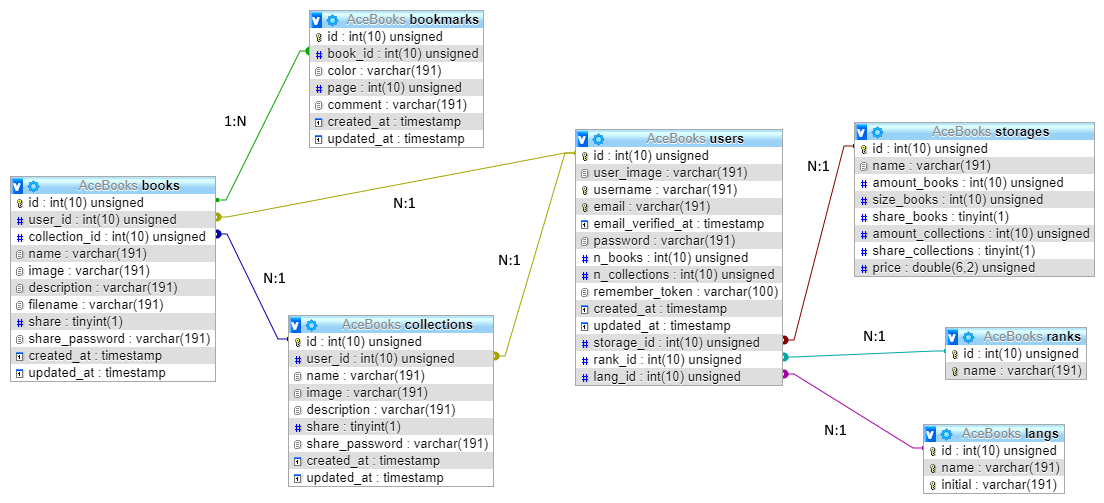
Para el desarrollo del proyecto he usado mi PC que tiene los siguientes componentes:

* Procesador: AMD Ryzen 5 2600.
* Placa base: MSI B450-A PRO.
* RAM: 8GB DDR4.
* Tarjeta gráfica: GTX 660 de 2GB.
* Discos duros: 1 TB + 3 TB.

**3.2 Software.**

* Sistema operativo: Windows 10.
* Editor de código: Visual Studio Code.
* Control de versiones: Git (repositorio en GitHub).
* Para documentar el proyecto: Microsoft Office 2019.
* Editores de imágenes: Gimp y Paint.NET.
* Contenedor de software: Docker (en el he usado PHP 7.3, phpmyadmin, MySQL y nginx).
* Gestores de paquetes: npm, yarn y composer.
* Framework en el backend: Laravel
* Framework en el frontend: Vue.JS (usado en momentos puntuales, la mayor parte del frontend es con Blade, el motor de plantillas de laravel).

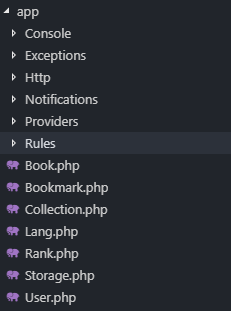
**4. Diseño.**

**4.1 Diagrama entidad-relación y tablas normalizadas.**

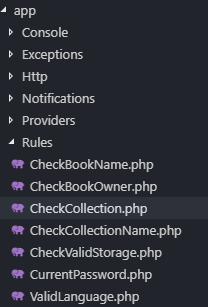
**4.2 Estructura de la aplicación.**

A continuación, explicaré la estructura del proyecto, me centraré en las partes importantes, ya que laravel tiene su estructura de directorios.

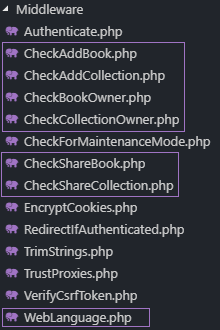
* En **app** tenemos los modelos, estos archivos son los encargados de manipular la información de la base de datos.



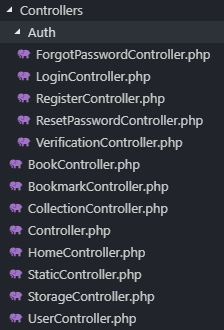
* En **app/Rules** tenemos normas personalizadas para validar los distintos formularios, todos ellos se encargan de comprobar que “x” valor recibido exista en la base de datos y/o el usuario sea el dueño de dichos datos.



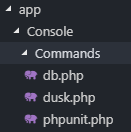
* En **app/Http/Middleware** tenemos distintos filtros que se aplicaran al acceder a distintas páginas, por ejemplo, uno que compruebe que el usuario ha iniciado sesión o ha verificado su correo (entre otras cosas), yo he creado los siguientes (resaltados en la imagen), principalmente son para controlar los límites de la tarifa del usuario, además hay otros que se encargan de restringir el acceso a documentos y colecciones que no son de su propiedad.



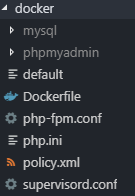
* En **app/Http/Controllers** tenemos los controladores, estos son los encargados de usar las funciones proporcionadas por los modelos para tratar la información y enviar los resultados a las vistas.



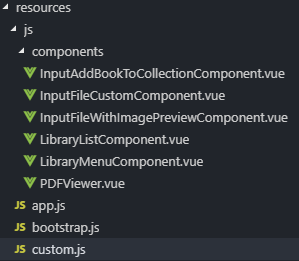
* En **app/Console/Commands** tenemos una serie de comandos personalizados, creados para acelerar las pruebas durante el desarrollo de la página web, el primero se encarga de importar la base de datos, los otros dos son para hacer tests con dusk y phpunit.



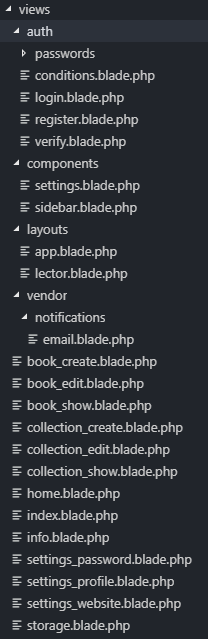
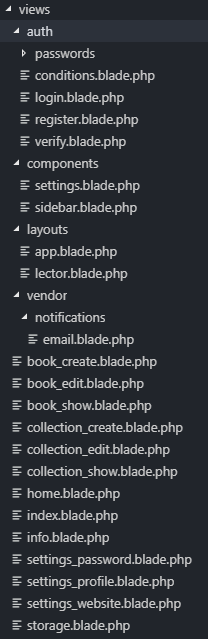
* En **database/migrations** tenemos la estructura de la base de datos y en **database/seeds** una serie de datos base que serán importados junto a la estructura de la base de datos.
* En **docker** tenemos los archivos necesarios para configurar el contenedor usado para desarrollar la página web.



* En **public/images** se almacenarán las imágenes de la web, incluidas las portadas de las colecciones y los documentos.
* En **public/books** se almacenarán los documentos de los usuarios.
* En **resources/js** tenemos los archivos encargados de empaquetar todo el JavaScript de nuestra web en un solo archivo, además de custom.js que será el archivo que contendrá la lógica del sidebar en la versión para pantallas pequeñas.
* En **resources/js/components** tenemos los distintos componentes de Vue.JS



* En **resources/lang** tenemos los distintos archivos que proveen la traducción en castellano e inglés de la página web.
* En **resources/sass** tenemos las hojas de estilos
* En **resources/views** tenemos las distintas vistas de la página web, todas ellas usan el motor de plantillas blade de laravel.



* En **routes/web.php** podemos encontrar todas las rutas de la página web.
* En **la raíz del proyecto** tenemos docker-compose.yml este archivo se encarga de crear el contenedor de docker, usando los ajustes que tiene en su interior y los archivos del directorio docker. Además, tenemos los distintos “.env”, “.env.docker” es usado para iniciar el proyecto con docker, mientras que “.env.example” es para iniciar el proyecto en un servidor o en local usando por ejemplo XAMPP, WAMPP o LAMPP.

**4.3 Flujo de control de cada una de las páginas PHP que componen el proyecto.**

**5. Conclusiones**

Tras realizar este proyecto, me he parado a pensar que características que debería pulir, incluso cuales añadir (ya hablaré de esto en el siguiente punto).

Cuando tuve que pararme a pensar que proyecto hacer lo tenía más que claro, Ace Books es una idea que siempre ha estado en mi cabeza, he usado un montón de lectores de documentos, pero todos tenían algo que los hacían difíciles de usar, desde consumir muchos recursos hasta faltarles características que para mí eran indispensables.

Tras comenzar el proyecto seguí los plazos que me marcaba, he tenido mis dificultades, el lector ha sido reescrito tres veces, si, como lees, aún así hay cosas que pulir en él. Incluso más de una vez se me ha pasado por mi cabeza la idea de reescribir parte del backend para que las sesiones funcionen con tokens y así poder crear una aplicación con react native.

He aplicado cosas que me han enseñado en la empresa, una de ellas es docker, me alegro de haberlo implementado, ya que a última hora el servidor del instituto de hizo incompatible con mi proyecto.

Por mi cuenta he aprendido a como usar vue.js ya que el lector de documentos requería de este framework, si no hubiese usado esto el código del lector sería muy difícil de mantener.

**6. Mejoras**

En lo que mejoras se refiere, me centraría en mejorar el código del lector y pulir ciertos controladores que se han quedado un poco mal organizados. Pero por falta de tiempo esto no ha sido posible.

En cuanto nuevas funciones tengo unas cuantas en mente:

* Reescribir parte del backend y el frontend para que las sesiones funcionen con tokens, usaría laravel passport, esto implicaría que el frontend fuese completamente en vue.js.
* Terminar la sección de tarifas, ofrecer un servicio premium con alojamiento ilimitado para así sustentar los servidores de la página web y ofrecer nuevas funciones a los usuarios.
* Soporte a más formatos, actualmente solo se leen documentos pdf.
* Añadir la posibilidad de compartir documentos y colecciones, esto implicaría añadir la función de reportar contenido que infrinja las leyes de copyright.
* Crear un panel de administración propio, ya que actualmente se usa voyager y este limita la gestión la web.
* Añadir un sistema de notificaciones, se informará al usuario de la retirada de documentos y colecciones compartidas que infrinjan las leyes de copyright, además de proporcionar información de rebajas en las tarifas de la web.
* Posibilidad de recomendar libros a otros usuarios.
* Un sistema para publicar libros, ya sean de pago o gratuitos.
* Añadir tests e integración continua, para evitar subir a producción versiones con errores.
* Eliminar completamente jQuery, se usaría bootstrap junto a vue.js.
* Crear una aplicación de la web con react native, para ello se tendría que realizar antes el primer punto de esta lista.

**7. Bibliografía.**