



# ECOBOOK

29/05/2024

---

Hugo Sanchis Belda

2n DAW 23/24

I.E.S Lluís Simarro

Edgar Carbonell Martinez





# RESUMEN

En este proyecto, he creado una aplicación web para gestionar un Banco de libros en un instituto, utilizando PHP para el backend y React para el frontend. Esta experiencia nos ha permitido fortalecer nuestros conocimientos en PHP y aprender a usar React de manera efectiva.

En la primera fase, me dediqué a investigar cómo React puede conectarse con un backend hecho en PHP. Esta investigación fue esencial para entender cómo React puede crear interfaces de usuario dinámicas y cómo se puede integrar con PHP para manejar la lógica del servidor y las interacciones con la base de datos.

Después de adquirir estos conocimientos, diseñe y creé la base de datos utilizando HeidiSQL. Planifiqué una estructura eficiente para manejar información sobre libros y estudiantes, asegurando que los datos se mantuvieran ordenados y consistentes.

En resumen, Ecobook no solo ha sido un proyecto para aprender a usar React y PHP, sino también una oportunidad para crear una aplicación útil para el sistema educativo. Este proyecto nos permitió combinar una gestión eficiente del backend con la creación de interfaces de usuario atractivas y funcionales, dejando un impacto positivo en la comunidad educativa y demostrando cómo el software bien diseñado puede resolver problemas reales.



# ECOBOK

Hugo Sanchis Belda



# ÍNDICE

<b>1. INTRODUCCIÓN.....</b>	<b>5</b>
<b>2. OBJETIVOS.....</b>	<b>6</b>
<b>3. ENTORNO DE TRABAJO.....</b>	<b>8</b>
3.1 Entorno Cliente (FrontEnd).....	8
3.2 Entorno Servidor (BackEnd).....	8
3.3 React.....	9
3.4 Flujo de datos.....	9
<b>4. DESARROLLO DEL PROYECTO.....</b>	<b>12</b>
4.1 ANÁLISIS Y DISEÑO.....	12
<b>5. IMPLEMENTACIÓN.....</b>	<b>23</b>
5.1 XAMPP.....	23
5.2 Dominio.....	24
5.3 Base de Datos.....	25
5.4 Archivos de configuración.....	27
5.5 Build de la App.....	28
5.6 Filezilla.....	30
<b>6. VISTAS Y SUS ACCIONES.....</b>	<b>32</b>
6.1 Login.....	32
6.2 Inicio.....	33
6.3 Estudiantes.....	34
6.4 Cursos.....	36
6.5 Asignaturas.....	38
6.6 Libros.....	40
6.7 Ejemplares/Copias.....	43
6.8 Asignar.....	46
6.9 Desasignar.....	49
6.10 Otras vistas fuera de flujo.....	51
<b>7. CONCLUSIÓN.....</b>	<b>52</b>
<b>8. RECURSOS.....</b>	<b>53</b>
<b>9. BIBLIOGRAFÍA.....</b>	<b>54</b>



# 1. INTRODUCCIÓN

En los institutos de educación, una de las necesidades más urgentes que se ha detectado es la carga económica que representa para las familias la compra de libros nuevos cada año. Este problema no solo impacta la economía familiar, sino que también tiene repercusiones ambientales significativas debido a la producción y el desperdicio de materiales educativos.

Hasta ahora, diversas soluciones han intentado abordar este problema. Algunas escuelas han implementado programas de préstamo de libros, donde los estudiantes pueden tomar prestados libros por un período determinado y luego devolverlos. También se han explorado iniciativas de donaciones, donde los alumnos que ya no necesitan los libros los donan a la escuela para uso futuro. Además, se han desarrollado plataformas en línea que permiten la compra y venta de libros de segunda mano entre estudiantes. Sin embargo, estas soluciones presentan ciertas limitaciones como la dificultad en la gestión logística o el control del estado de los libros.

En este contexto, se plantea ECOBOOK como una nueva solución innovadora. ECOBOOK es una aplicación web diseñada para gestionar de manera eficiente la red de libros en un instituto, fomentando el intercambio y la reutilización de materiales educativos.

Los beneficios de esta solución son múltiples. En primer lugar, ECOBOOK reduce significativamente el gasto en libros para las familias, ya que facilita el intercambio a bajo costo de libros usados. En segundo lugar, promueve valores de sostenibilidad y responsabilidad ambiental entre los estudiantes, ya que reutilizar libros ayuda a reducir el desperdicio de recursos.

Sin embargo, también presenta algunos inconvenientes. La implementación inicial de la plataforma puede requerir una inversión de tiempo y recursos por parte del instituto. Además, es necesario asegurar que todos los participantes utilicen la plataforma de manera adecuada para garantizar la disponibilidad y el estado de los libros.



## 2. OBJETIVOS

ECOBOOK tiene como objetivo clave desarrollar una aplicación web de gestión de libros de un centro educativo con los siguientes beneficios:

### **Objetivo Económico:**

El aspecto económico de ECOBOOK es uno de sus pilares fundamentales. En muchos casos, los costos de los libros de texto representan una carga significativa para las familias, especialmente cuando hay varios hijos en edad escolar. ECOBOOK ayuda a reducir esta carga creando un sistema donde los libros son prestados y utilizados año tras año. Esto significa que una familia no tendrá que desembolsar grandes cantidades de dinero al inicio de cada año escolar para adquirir los libros requeridos.

### **Sostenibilidad Ambiental:**

Desde el punto de vista medioambiental, ECOBOOK tiene un impacto positivo significativo. La producción de libros de texto implica el uso de recursos naturales, como el papel y la tinta, además de la energía utilizada en los procesos de impresión y transporte. Al fomentar la reutilización de libros, ECOBOOK reduce la demanda de nuevos ejemplares, disminuyendo así la presión sobre los recursos naturales y reduciendo la huella de carbono asociada con la producción y distribución de libros nuevos.

### **Responsabilidad Comunitaria:**

ECOBOOK también promueve la responsabilidad comunitaria entre los estudiantes. Al participar en el intercambio de libros, los estudiantes aprenden a valorar y cuidar los recursos que comparten con sus compañeros. Esta práctica fomenta una cultura de responsabilidad y respeto mutuo, donde cada estudiante entiende que su comportamiento impacta a los demás. Los estudiantes se convierten en custodios de los libros, asegurándose de que se mantengan en buenas condiciones para que otros puedan usarlos en el futuro.



## **Impacto Educativo y Social:**

El acceso equitativo a los libros de texto es crucial para asegurar que todos los estudiantes tengan las mismas oportunidades de éxito académico. ECOBOOK ayuda a nivelar el terreno de juego garantizando que todos los estudiantes, independientemente de su situación económica, tengan acceso a los materiales educativos que necesitan. Esto no solo mejora los resultados académicos, sino que también fortalece la cohesión social dentro del instituto, creando un ambiente donde todos los estudiantes se sienten apoyados y valorados.



## 3. ENTORNO DE TRABAJO

**Arquitectura:** Para el desarrollo de ECOBOOK, se ha adoptado un modelo de arquitectura cliente-servidor de tres capas. Este modelo se divide en las siguientes capas:

### 3.1 Entorno Cliente (FrontEnd)

Representa la parte con la que los usuarios interactúan, es la interfaz de usuario en el navegador. Las tecnologías utilizadas son:

- **Bootstrap 5:** Framework para crear interfaces adaptables y consistentes.
- **React:** Biblioteca de JavaScript para construir interfaces de usuario interactivas y dinámicas.
- **Vite:** Herramienta de construcción rápida, hace el esqueleto del proyecto.

### 3.2 Entorno Servidor (BackEnd)

Procesa las solicitudes del cliente y proporciona los recursos solicitados. Las tecnologías utilizadas son:

- **MySQL:** Sistema de gestión de bases de datos para almacenar y gestionar datos de manera eficiente y segura.
- **PHP:** Lenguaje de programación del lado del servidor para la lógica de la aplicación.





### 3.3 React

React está diseñado de manera que la interfaz de usuario (UI) se construya a partir de piezas pequeñas y reutilizables llamadas componentes. Los componentes gestionan su propio estado interno. Además, React utiliza un enfoque donde los datos solo van en una dirección: desde los componentes padres hacia los componentes hijos. Esto se hace pasando datos como propiedades, conocidas como "props", de un componente a otro.

- **Componentes:** Pequeñas piezas de código que representan partes de la interfaz de usuario.
- **Props:** Propiedades que se pasan de un componente padre a un componente hijo.

### 3.4 Flujo de datos

La comunicación entre el frontend React y el backend PHP se ha realizado mediante solicitudes HTTP a través de APIs REST. La comunicación se realiza principalmente mediante solicitudes POST/GET para enviar y recibir datos del servidor PHP.

- **Solicitud HTTP:** React envía una solicitud HTTP al backend PHP.
- **Procesamiento en el Servidor:** PHP valida y procesa los datos, interactuando con la base de datos si es necesario.
- **Respuesta:** PHP envía una respuesta de vuelta a la aplicación React.

**Actualización del Estado:** React maneja la respuesta y actualiza el estado de la aplicación en consecuencia.

## ➤ Solicitud desde React:

```
const handleSubmit = () => {
  if (checkForm()) {
    request("post", getEndpoint(Endpoints.Copies.createCopy.create), {
      ...data,
      book_guid,
    })
    .then(() => {
      successNotification(ViewStrings.messages.copyCreated);
      push(replacePaths(Paths[Views.copies].path, [{ book_guid }]));
    })
    .catch((err) => errorNotification(err.message));
  } else errorNotification(ViewStrings.messages.inputError);
};
```

Aquí, request es un custom hook que hace uso de **axios** para enviar una solicitud POST al servidor PHP.

- **Axios:** Es un módulo que te permite hacer peticiones desde React a PHP.

## ➤ Recepción en PHP:

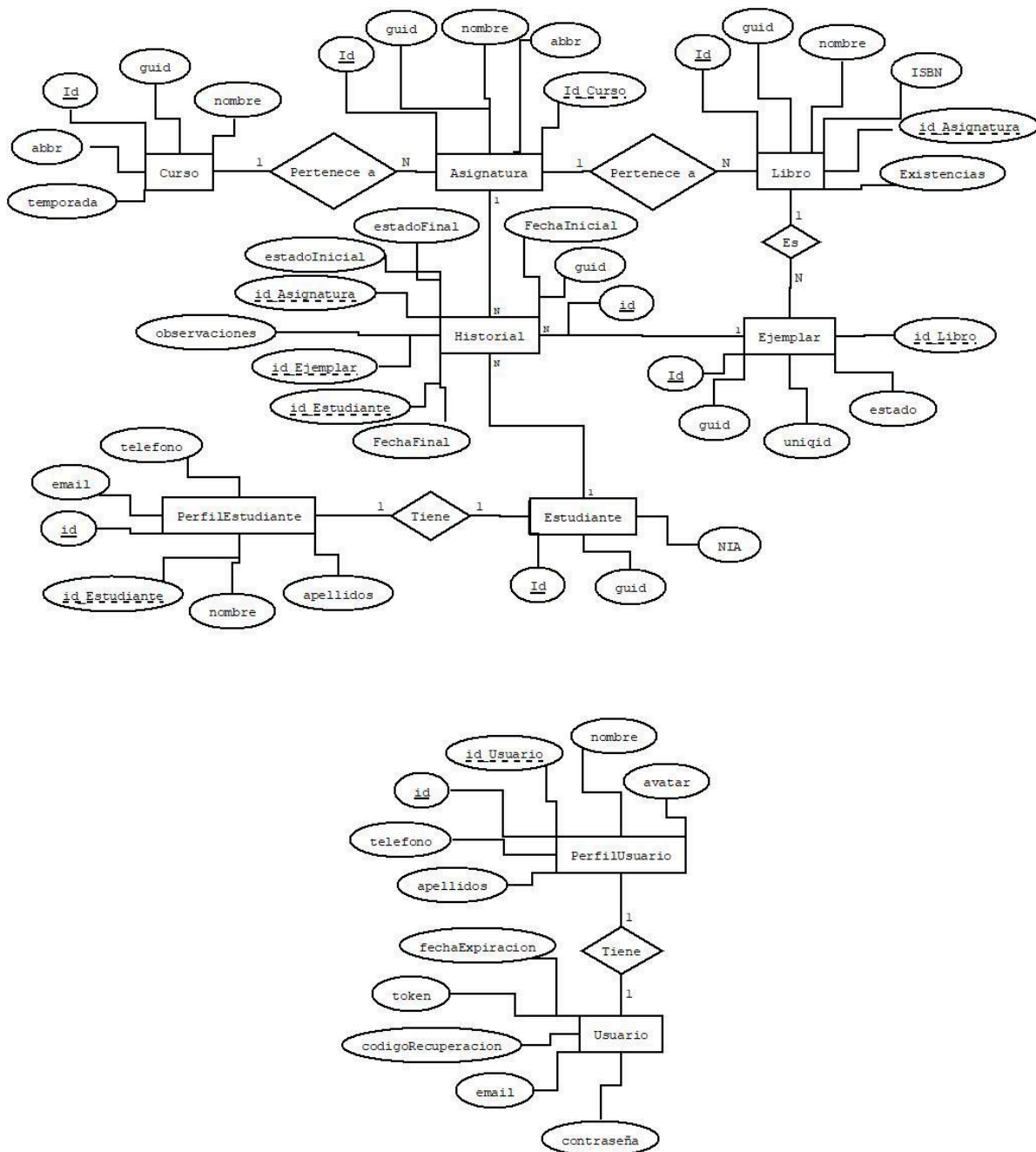
```
api > endpoints > copies > create.php > ...
1  <?php
2
3  include_once '../config/config.php';
4
5  $database = new Database();
6  $db = $database->getConnection();
7
8  $data = postInput();
9  try {
10     $db->beginTransaction();
11     checkAuth();
12
13     $input = validate($data, [
14         'state' => 'required|numeric',
15         "book_guid" => "required|string"
16     ]);
17
18     $book = Book::getByGuid($db, $input->book_guid);
19     $copy = new Copy($db);
20     $copy->state = $input->state;
21     $copy->book_id = $book->id;
22     $copy->store();
23
24     $db->commit();
25     Response::sendResponse();
26 } catch (\Exception $th) {
27     $db->rollBack();
28     print_r(json_encode(array("status" => false, "message" => $th->getMessage(), 'code' => $th->getCode())));
29 }
```

- **Recepción de Datos:** Los datos enviados desde React son recibidos usando `postInput()`.
- **Validación y Procesamiento:** Se validan los datos y se procesa la lógica (buscar el libro por `GUID`, crear una nueva copia, etc.).
- **Respuesta:** Se le devuelven los datos al cliente usando: `Response::sendResponse()`.

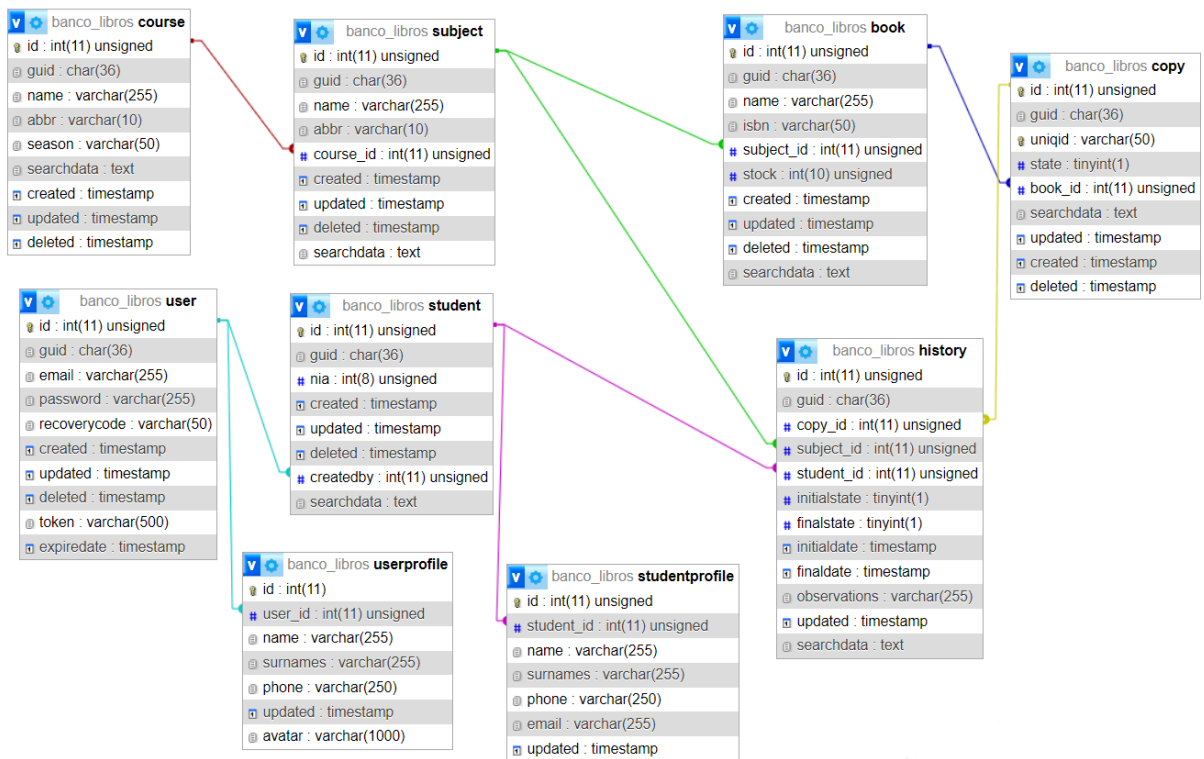
## 4. DESARROLLO DEL PROYECTO

### 4.1 ANÁLISIS Y DISEÑO

- Diagrama de Entidad Relación

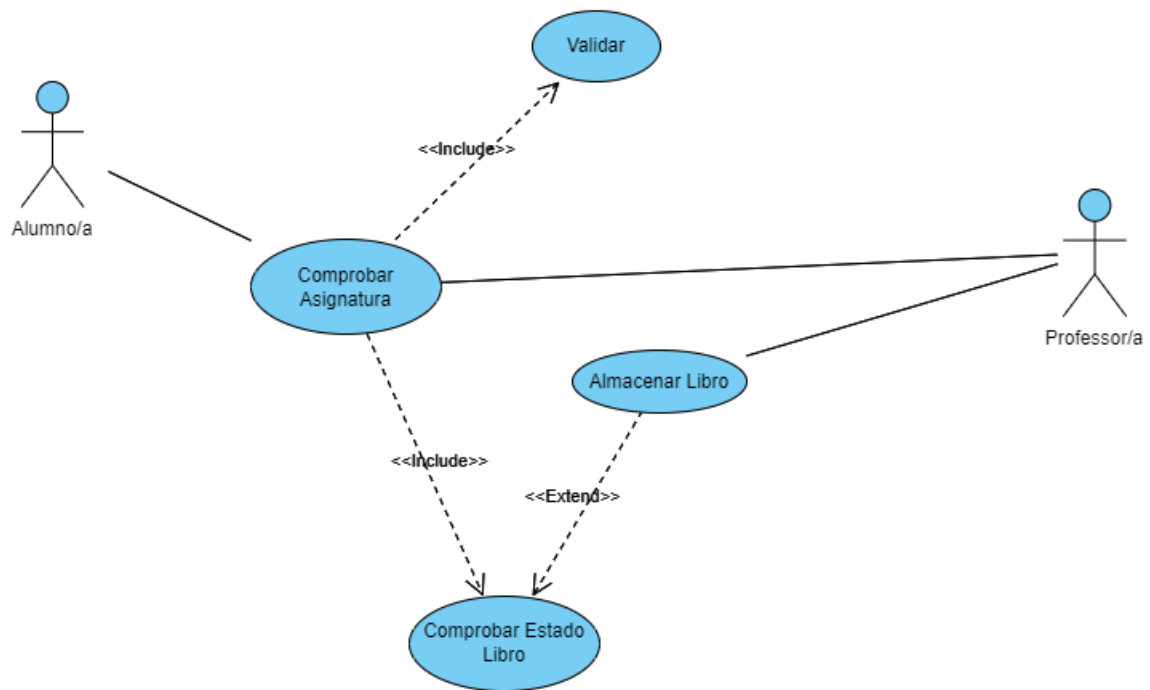


- Diagrama de clases



- Casos de uso

- Realizar devolución del libro



<b>Nom:</b> Validar
<b>Descripción:</b> El profesor valida el NIA del alumno comprobandolo en la Base de Datos.
<b>Actores:</b> Alumno y Profesor
<b>Precondiciones:</b> Se requiere que el alumno esté registrado en la Base de Datos.
<b>Curso normal del caso de uso:</b> <ol style="list-style-type: none"> <li>1. El alumno entrega su carnet donde podemos obtener el NIA del mismo.</li> <li>2. El profesor introduce el NIA del alumno en el sistema para poder comprobarlo.</li> <li>3. El sistema compara el NIA introducido con los existentes en la Base de Datos.</li> </ol>
<b>Postcondiciones:</b> El sistema confirma si el NIA corresponde a un alumno del centro para poder seguir con el procedimiento de devolución.
<b>Alternativas:</b> <ol style="list-style-type: none"> <li>1. El sistema no encuentra ningún alumno con el NIA introducido.</li> <li>2. Fin del caso de uso.</li> </ol>

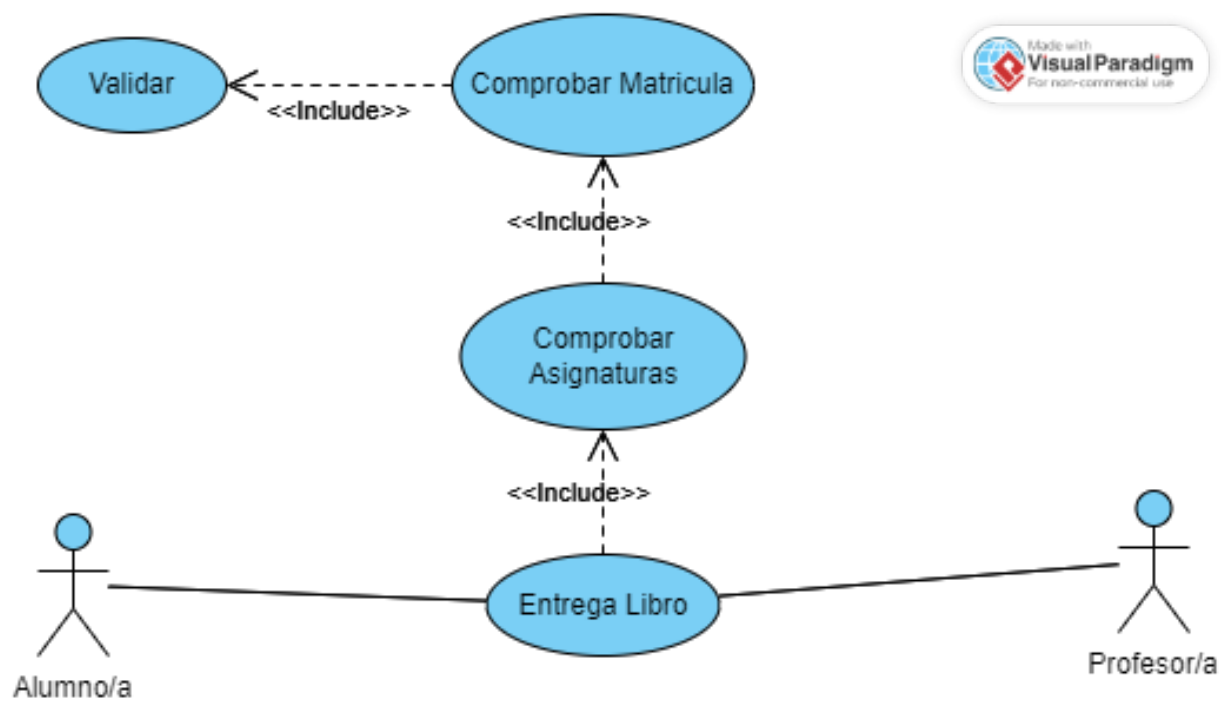
<b>Nom:</b> Comprobar Asignaturas
<b>Descripción:</b> El profesor comprueba todas las asignaturas que tiene el alumno para devolver.
<b>Actores:</b> Alumno y Profesor
<b>Precondiciones:</b> El alumno ha de devolver todos los libros prestados.
<b>Curso normal del caso de uso:</b> <ol style="list-style-type: none"> <li>1. El profesor consulta las asignaturas del alumno que tiene para devolver.</li> <li>2. El profesor comprueba que los libros devueltos correspondan con las asignaturas que ha cursado el alumno.</li> </ol>
<b>Postcondiciones:</b> Una vez comprobados los libros el profesor puede seguir con el caso de uso.
<b>Alternativas:</b> <ol style="list-style-type: none"> <li>1. No es posible el acceso a los resultados académicos del alumno.</li> </ol>

<b>Nombre:</b> Comprobar estado libro (Ejemplar)
<b>Descripción:</b> El profesor realizará la comprobación del estado del libro para saber si se puede volver a prestar.
<b>Actores:</b> Profesor
<b>Precondiciones:</b> El alumno debe haber entregado los libros y el profesor comprobar que sean los correctos
<b>Curso normal del caso de uso:</b> <ol style="list-style-type: none"> <li>1. El profesor comprueba el estado de los libros.</li> <li>2. El profesor introduce el nuevo estado de los libros en la sección correspondiente del sistema.</li> <li>3. El sistema actualizará el estado del libro en la Base de Datos.</li> </ol>
<b>Postcondiciones:</b> Una vez comprobados los libros el profesor puede seguir con el caso de uso.
<b>Alternativas:</b>

<b>Nombre:</b> Almacenar Libro (Ejemplar)
<b>Descripción:</b> El profesor almacenará el libro para posteriormente volver a prestarlo a un nuevo alumno.
<b>Actores:</b> Profesor
<b>Precondiciones:</b> El profesor deberá registrar el nuevo estado del ejemplar
<b>Curso normal del caso de uso:</b> <ol style="list-style-type: none"> <li>1. El profesor almacena el libro en el lugar correspondiente del instituto.</li> <li>2. El profesor indica en el sistema que el libro ha sido almacenado.</li> <li>3. El sistema realizará la gestión de almacenar el libro en la base de datos.</li> </ol>
<b>Postcondiciones:</b> Una vez almacenado el libro terminará el caso de uso.
<b>Alternativas:</b>



➤ Realizar asignación del libro



<b>Nombre:</b> Validar
<b>Descripción:</b> El profesor introducirá el NIA del alumno en la aplicación para comprobar que el alumno exista en la base de datos.
<b>Actores:</b> Profesor y Alumno
<b>Precondiciones:</b> El NIA debe corresponder al alumno.
<b>Curso normal del caso de uso:</b> <ol style="list-style-type: none"> <li>1. El profesor consultará en el sistema el NIA indicado por el alumno.</li> <li>2. El sistema hará una consulta a la Base de Datos para comprobar que el NIA indicado por el alumno sea válido.</li> <li>3. El sistema indicará al profesor si el NIA es válido o no.</li> </ol>
<b>Postcondiciones:</b> Una vez puesto el NIA el profesor podrá seguir con el caso de uso.
<b>Alternativas:</b> <ol style="list-style-type: none"> <li>1. El NIA del alumno no es el que le corresponde.</li> </ol>

<b>Nombre:</b> Comprobar Matrícula
<b>Descripción:</b> El profesor deberá comprobar la matrícula del estudiante para saber que curso va a realizar.
<b>Actores:</b> Profesor y Alumno
<b>Precondiciones:</b> El alumno debe haber entregado la matrícula.
<b>Curso normal del caso de uso:</b> <ol style="list-style-type: none"> <li>1. El profesor consultará en el sistema del centro la matrícula del alumno.</li> </ol>
<b>Postcondiciones:</b> Una vez comprobada la matrícula el profesor pasará al siguiente caso de uso.
<b>Alternativas:</b> <ol style="list-style-type: none"> <li>1. El alumno no ha entregado la matrícula.</li> </ol>

<b>Nombre:</b> Comprobar asignaturas
<b>Descripción:</b> Una vez comprobada la matrícula, el profesor comprobará las asignaturas que ha de cursar el alumno.
<b>Actores:</b> Profesor y Alumno
<b>Precondiciones:</b> La matrícula debe haber sido entregada y comprobada.
<b>Curso normal del caso de uso:</b> <ol style="list-style-type: none"> <li>1. El profesor consultará en el sistema las asignaturas que el alumno va a cursar.</li> <li>2. El sistema consultará en la Base de Datos las asignaturas que el alumno tiene que cursar y las mostrará para que el profesor pueda consultarlas.</li> </ol>
<b>Postcondiciones:</b> Una vez consultadas las asignaturas, el profesor pasará al siguiente caso de uso.
<b>Alternativas:</b> <ol style="list-style-type: none"> <li>1. Las asignaturas no corresponden al alumno.</li> </ol>

<b>Nombre:</b> Entrega libro
<b>Descripción:</b> Una vez comprobadas las asignaturas, el profesor entregará los libros correspondientes al alumno.
<b>Actores:</b> Profesor y Alumno
<b>Precondiciones:</b> Hay que tener un stock mínimo para cubrir las necesidades de las matrículas.
<b>Curso normal del caso de uso:</b> <ol style="list-style-type: none"> <li>1. El profesor entregará los libros al alumno.</li> <li>2. El profesor indicará en el sistema que ha realizado la entrega de los libros al alumno.</li> <li>3. El sistema realizará la gestión de asignar los libros al alumno para el curso actual en la Base de Datos.</li> </ol>
<b>Postcondiciones:</b> Una vez entregados los libros al alumno, finaliza el caso de uso.
<b>Alternativas:</b> <ol style="list-style-type: none"> <li>1. No existen suficientes libros para abastecer la matrícula.</li> </ol>



## ➤ Otros Casos de Uso de la Aplicación

### 1. Iniciar Sesión

- **Actor:** Profesor
- **Descripción:** El profesor ingresa sus credenciales para acceder a la aplicación.

### 2. Cerrar Sesión

- **Actor:** Profesor
- **Descripción:** El profesor cierra su sesión en la aplicación.

### 3. Recuperar Contraseña

- **Actor:** Profesor
- **Descripción:** El profesor puede solicitar recuperar su contraseña si la ha olvidado.

### 4. Editar Perfil

- **Actor:** Profesor
- **Descripción:** El profesor puede editar su información personal, como nombre, dirección de correo electrónico y foto de perfil.

### 5. Cambiar Contraseña

- **Actor:** Profesor
- **Descripción:** El profesor puede cambiar su contraseña por una nueva.

### 6. Cambiar Correo Electrónico

- **Actor:** Profesor
- **Descripción:** El profesor puede cambiar su dirección de correo electrónico.

### 7. Elegir Idioma

- **Actor:** Profesor
- **Descripción:** El profesor puede elegir el idioma de preferencia para la interfaz de la aplicación.

## 8. Gestionar Estudiantes

- Subcasos de Uso:

- **Dar de Alta a Estudiantes**

- **Actor:** Profesor

- **Descripción:** El profesor puede agregar nuevos estudiantes a la base de datos.

- **Editar Datos del Estudiante**

- **Actor:** Profesor

- **Descripción:** El profesor puede modificar la información de los estudiantes.

- **Dar de Baja a Estudiantes**

- **Actor:** Profesor

- **Descripción:** El profesor puede eliminar a estudiantes de la base de datos.

- **Ver Historial de Estudiantes**

- **Actor:** Profesor

- **Descripción:** El profesor puede ver el historial de libros asignados y devueltos por cada estudiante.

## 9. Gestionar Libros

- Subcasos de Uso:

- **Crear Libros en la Base de Datos**

- **Actor:** Profesor

- **Descripción:** El profesor puede agregar nuevos libros a la base de datos.

- **Modificar Libros**

- **Actor:** Profesor

- **Descripción:** El profesor puede modificar la información de los libros existentes.

- **Eliminar Libros**

- **Actor:** Profesor

- **Descripción:** El profesor puede eliminar libros de la base de datos.



## 10. Gestionar Cursos

- Subcasos de Uso:

- Crear Cursos

- Actor: Profesor

- Descripción: El profesor puede agregar nuevos cursos a la base de datos.

- Modificar Cursos

- Actor: Profesor

- Descripción: El profesor puede modificar la información de los cursos existentes.

- Eliminar Cursos

- Actor: Profesor

- Descripción: El profesor puede eliminar cursos de la base de datos.

## 11. Gestionar Asignaturas

- Subcasos de Uso:

- Crear Asignaturas

- Actor: Profesor

- Descripción: El profesor puede agregar nuevas asignaturas a la base de datos.

- Modificar Asignaturas

- Actor: Profesor

- Descripción: El profesor puede modificar la información de las asignaturas existentes.

- Eliminar Asignaturas

- Actor: Profesor

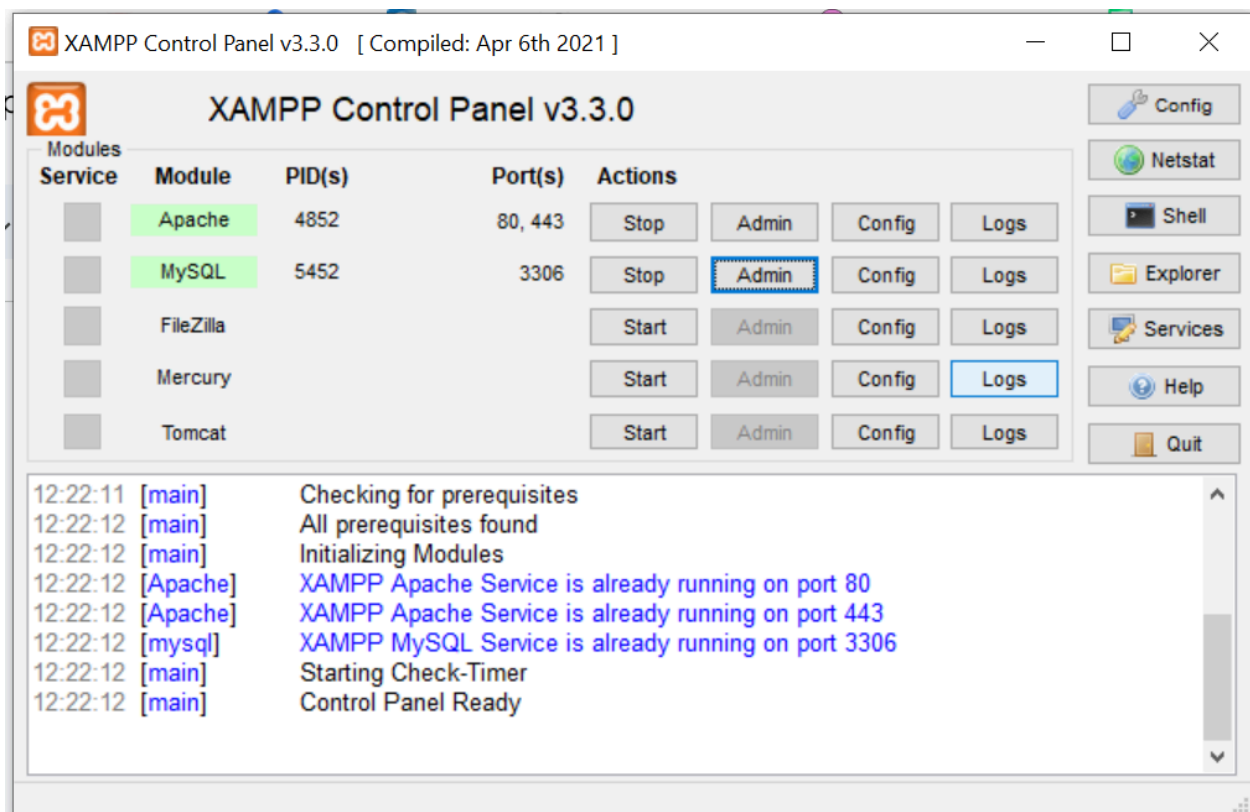
- Descripción: El profesor puede eliminar asignaturas de la base de datos.

## 5. IMPLEMENTACIÓN

Vamos a hablar sobre la implementación de este proyecto desde sus inicios hasta el paso final.

### 5.1 XAMPP

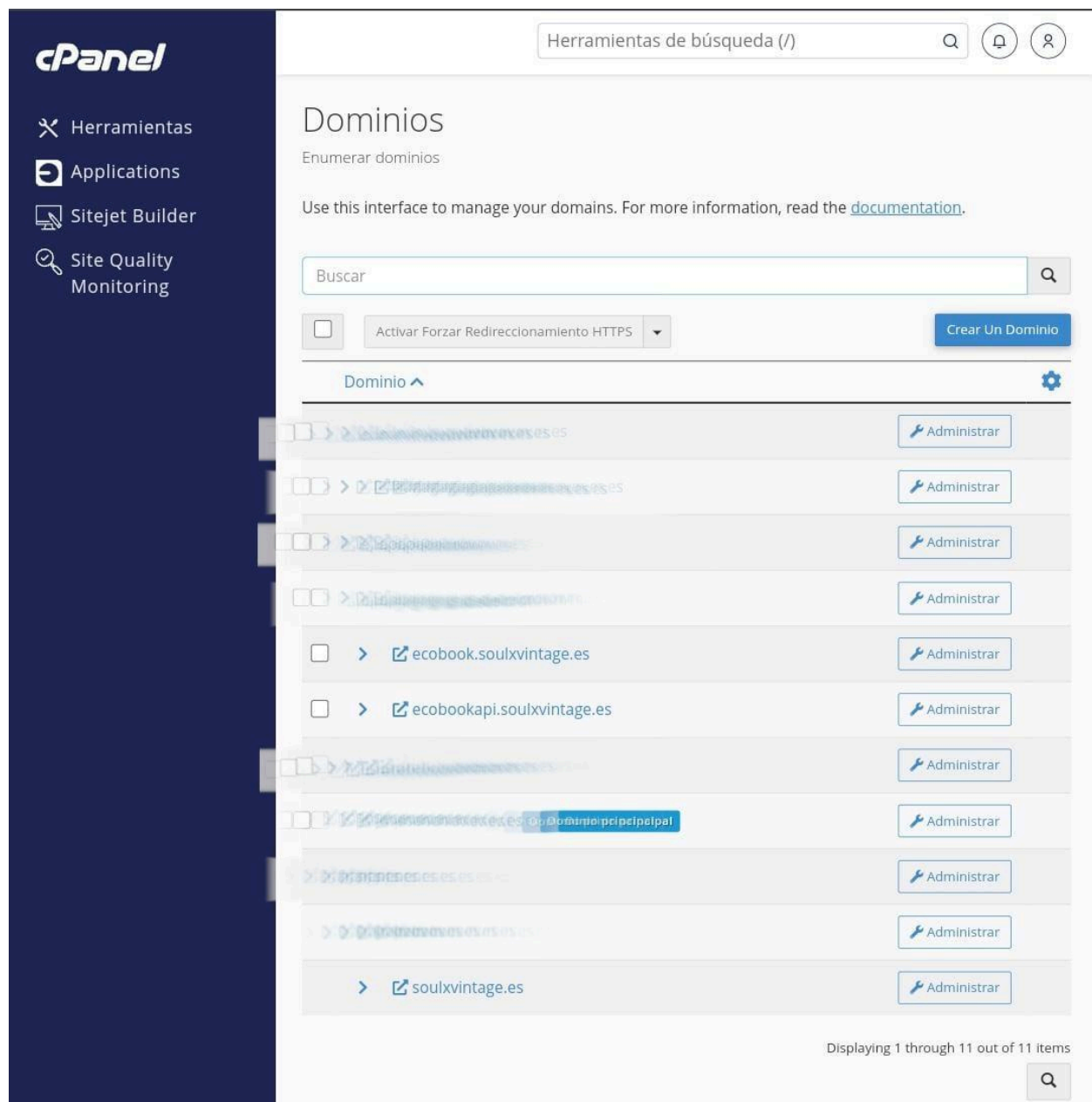
Durante el desarrollo de este proyecto, se ha trabajado en un servidor local utilizando XAMPP, he hecho pruebas, ajustes y también he creado la base de datos.



## 5.2 Dominio

Una vez terminado el desarrollo de la aplicación web hay que proceder a implementar esta dicha web en un servidor real y una base de datos conectada a dicho servidor.

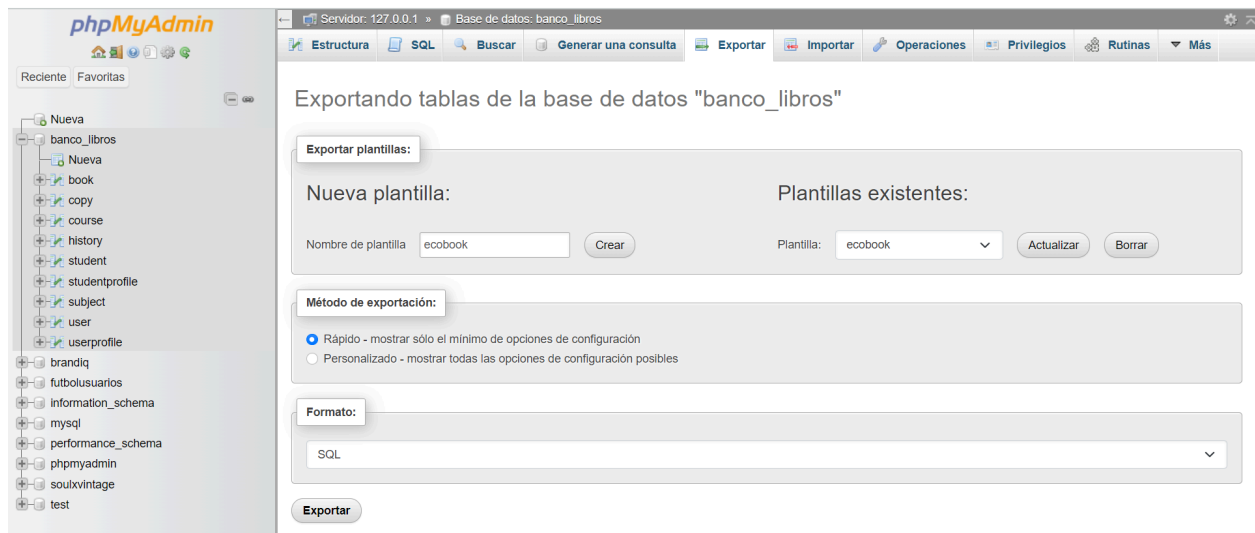
Para ello utilizaré un hosting que tenía ya y crearé dos subdominios, uno para alojar la api y otro para la app, así podré acceder a la web desde cualquier lugar.



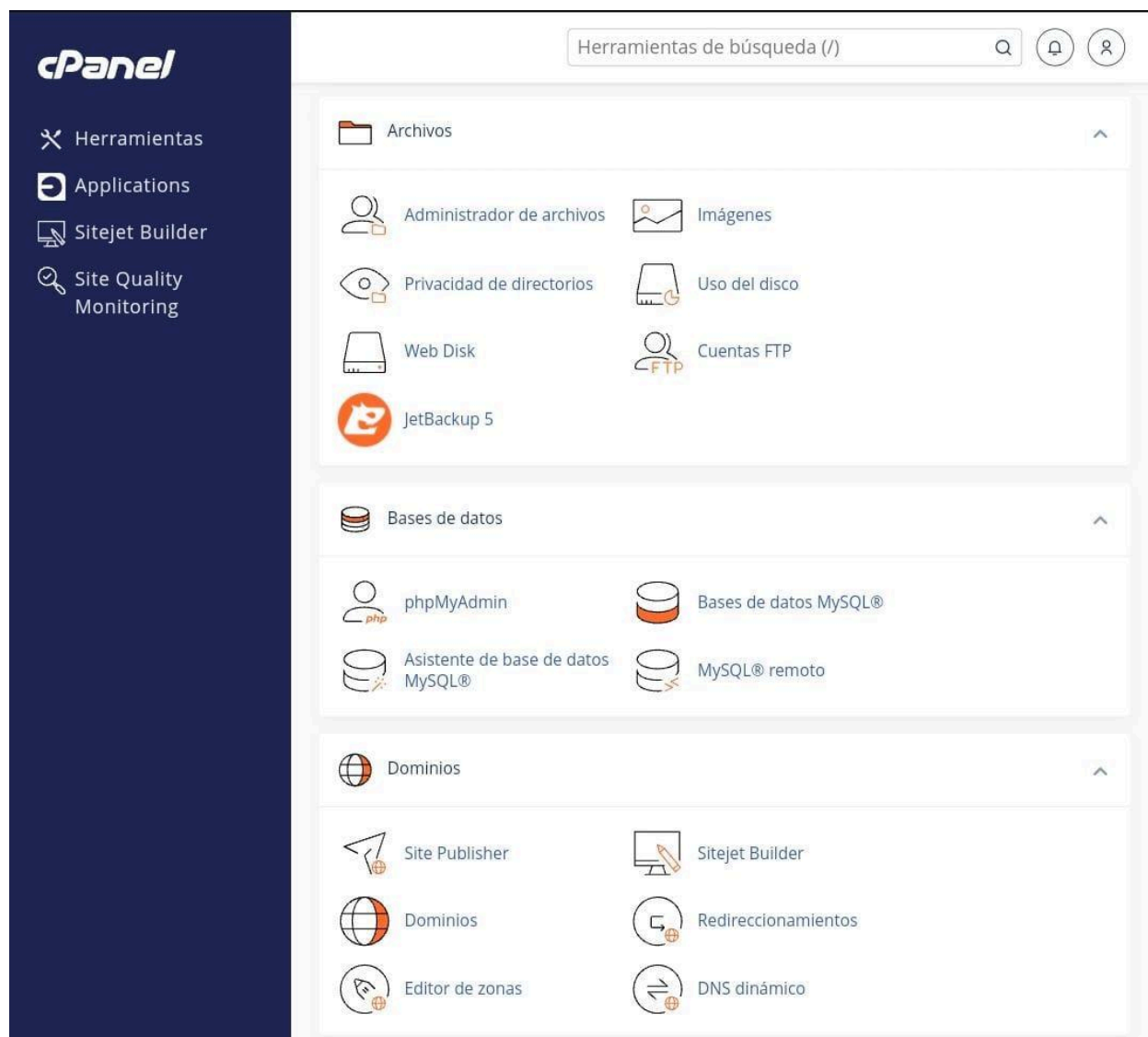


## 5.3 Base de Datos

Una vez dentro del servidor nos pondremos a crear nuestra base de datos con mySQL exportando desde nuestro servidor local el archivo SQL.



Y ahora hacemos la importación de la base de datos al servidor.

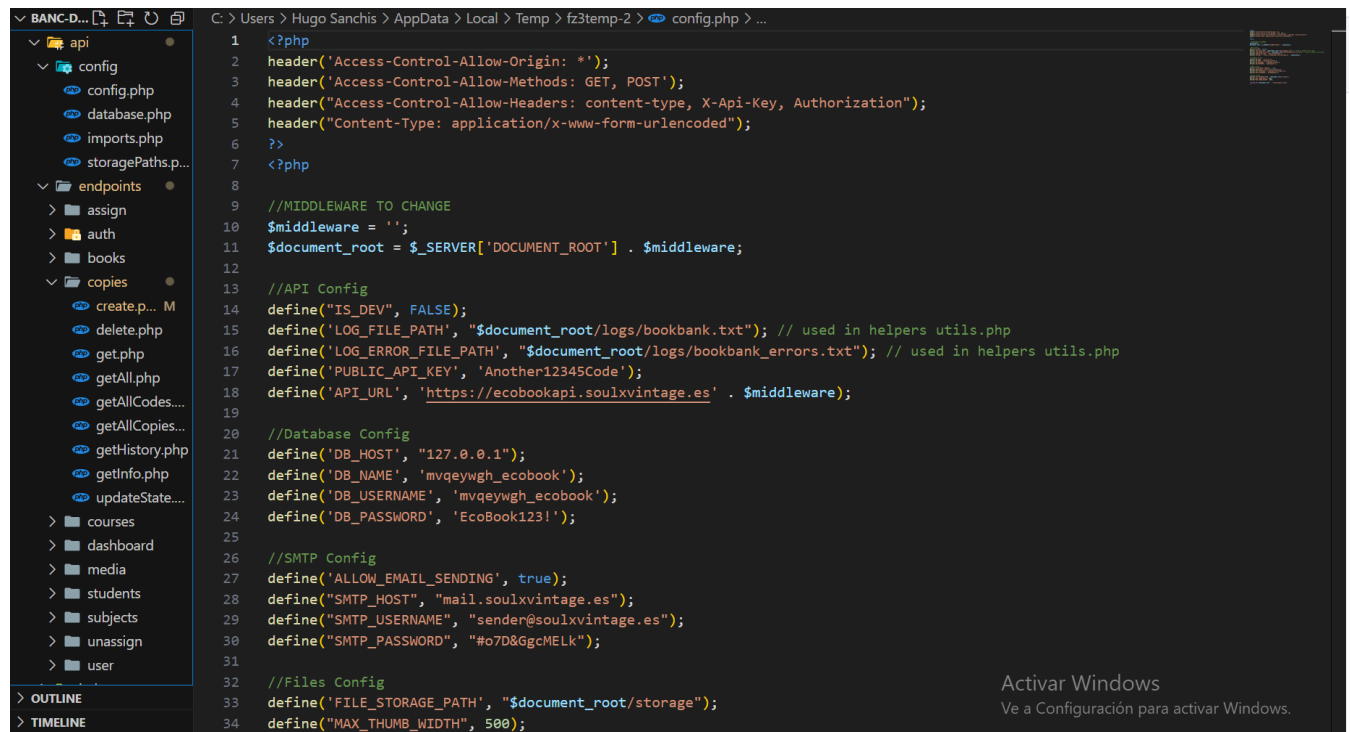


## 5.4 Archivos de configuración

Solo nos quedaría modificar nuestro código de configuración, tanto desde el frontend el archivo “.env” como desde el backend el archivo “config.php”.

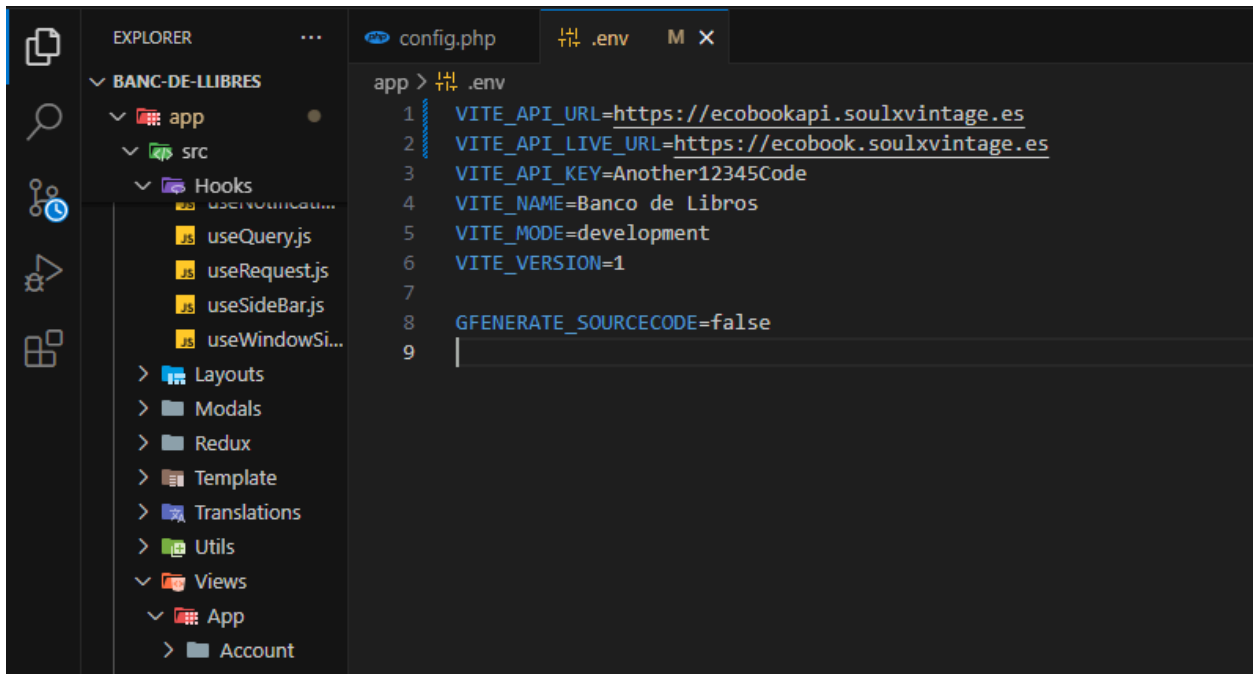
Así quedarían:

### conf.php



```
1 <?php
2 header('Access-Control-Allow-Origin: *');
3 header('Access-Control-Allow-Methods: GET, POST');
4 header('Access-Control-Allow-Headers: content-type, X-API-Key, Authorization');
5 header('Content-Type: application/x-www-form-urlencoded');
6 ?>
7 <?php
8
9 //MIDDLEWARE TO CHANGE
10 $middleware = '';
11 $document_root = $_SERVER['DOCUMENT_ROOT'] . $middleware;
12
13 //API Config
14 define('IS_DEV', FALSE);
15 define('LOG_FILE_PATH', "$document_root/logs/bookbank.txt"); // used in helpers utils.php
16 define('LOG_ERROR_FILE_PATH', "$document_root/logs/bookbank_errors.txt"); // used in helpers utils.php
17 define('PUBLIC_API_KEY', 'Another12345Code');
18 define('API_URL', 'https://ecobookapi.soulxvintage.es' . $middleware);
19
20 //Database Config
21 define('DB_HOST', "127.0.0.1");
22 define('DB_NAME', 'mvqeywgh_ecobook');
23 define('DB_USERNAME', 'mvqeywgh_ecobook');
24 define('DB_PASSWORD', 'EcoBook123!');
25
26 //SMTP Config
27 define('ALLOW_EMAIL_SENDING', true);
28 define('SMTP_HOST', "mail.soulxvintage.es");
29 define('SMTP_USERNAME', "sender@soulxvintage.es");
30 define('SMTP_PASSWORD', "#o7D&GgcMELk");
31
32 //Files Config
33 define('FILE_STORAGE_PATH', "$document_root/storage");
34 define('MAX_THUMB_WIDTH', 500);
```

## .env



## 5.5 Build de la App

Una vez modificados y listos para subir, hay que hacer la build de la parte del frontend que se realizaria ejecutando el siguiente comando:

```
PS C:\proyectoFinal\Banc-de-Llibres\app> npm run build

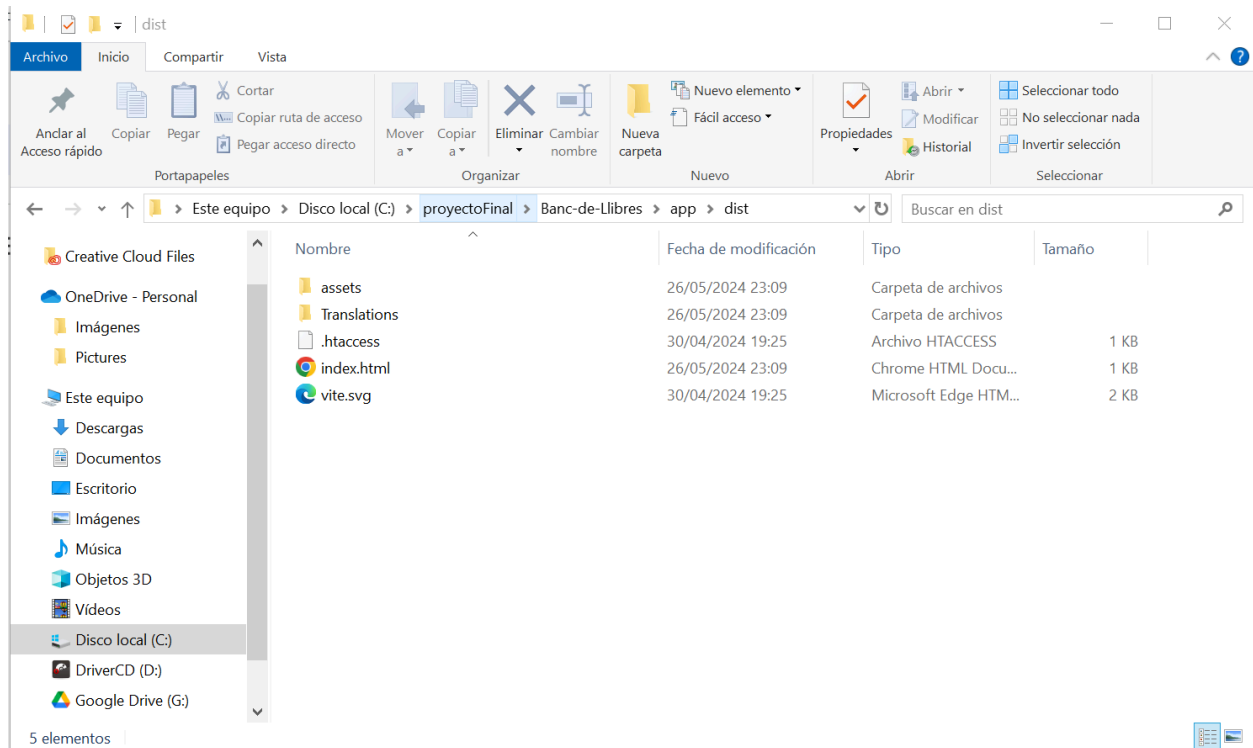
> app@0.0.0 build
> vite build

vite v4.5.2 building for production...
transforming (1) index.htmlDeprecation Warning: Passing percentage units to the global abs() function is deprecated.
In the future, this will emit a CSS abs() function to be resolved by the browser.
To preserve current behavior: math.abs(100%)
To emit a CSS abs() now: abs({100%})
More info: https://sass-lang.com/d/abs-percent

57 |   $dividend: abs($dividend);
    |               ~~~~~
    |               |
    |               v
node_modules\bootstrap\scss\vendor\_rfs.scss 57:14      divide()
node_modules\bootstrap\scss\mixins\_grid.scss 59:12    row-cols()
node_modules\bootstrap\scss\mixins\_grid.scss 85:13    @content
node_modules\bootstrap\scss\mixins\_breakpoints.scss 68:5 media-breakpoint-up()
node_modules\bootstrap\scss\mixins\_grid.scss 72:5    make-grid-columns()
node_modules\bootstrap\scss\_grid.scss 38:3           @import
node_modules\bootstrap\scss\bootstrap.scss 20:9       @import
src\Assets\styles\styles.scss 11:9                   root stylesheet

✓ 2497 modules transformed.
```

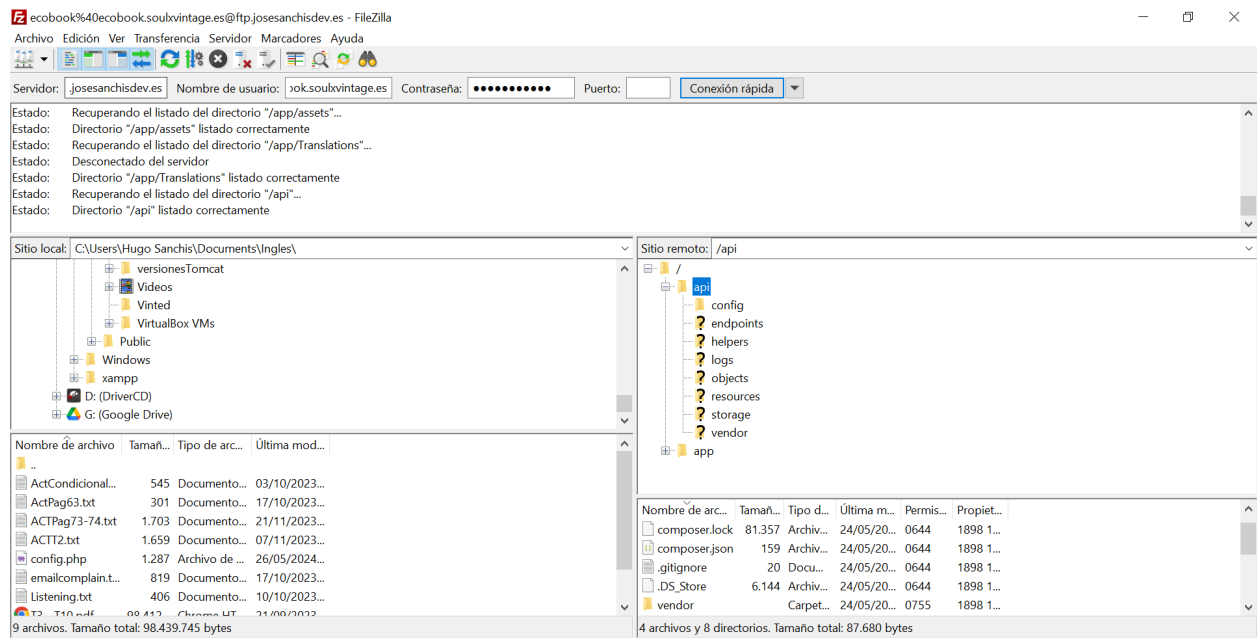
Esta build lo que hará es crear una carpeta llamada dist. Esta carpeta contiene todos los archivos optimizados y minificados listos para ser desplegados en el servidor.



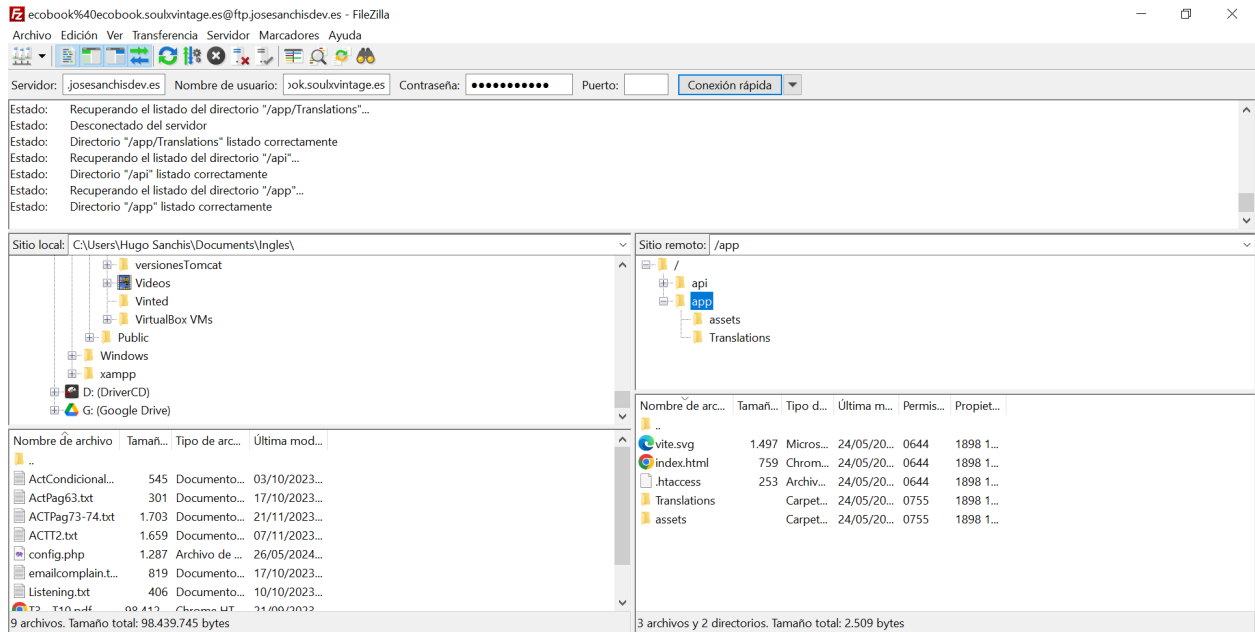
## 5.6 Filezilla

Por último nos conectaremos al servidor a través de Filezilla, para realizar una conexión con FTP y subir los archivos al servidor:

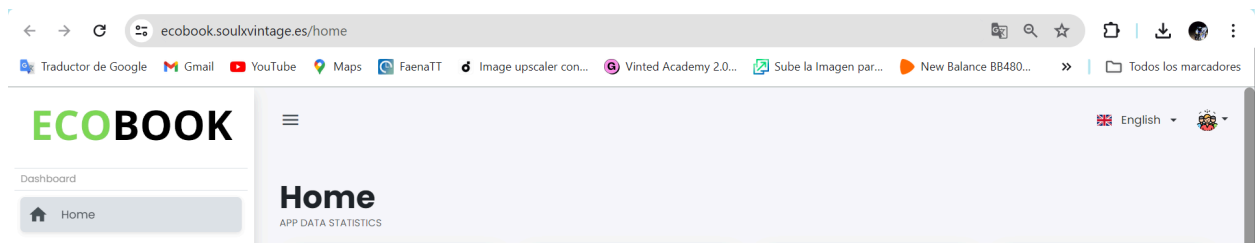
### ➤ Carpeta API con todo el Backend



## ➤ Carpeta APP con todo el frontend (build)



## ➤ La url de la web quedaría tal que así: [ecobook.soulxvintage.es](http://ecobook.soulxvintage.es)



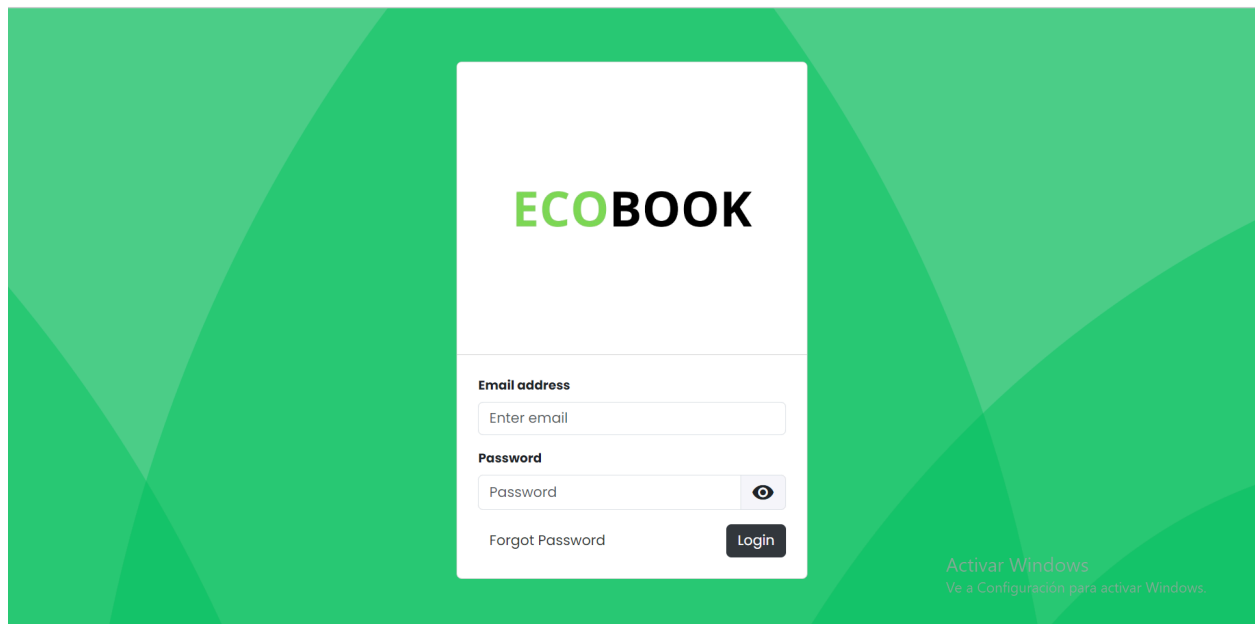
Como podemos ver nuestra web funciona con éxito.

## 6. VISTAS Y SUS ACCIONES

Ahora vamos a realizar una serie de pruebas para comprobar que todo funciona como corresponde.

### 6.1 Login

Cuando entramos a la aplicación se abrirá automáticamente la página para el login.





## 6.2 Inicio

Una vez hemos hecho login, nos redirigirá al Inicio. En esta sección de la aplicación podremos ver las siguientes estadística de la aplicación:

- Total de cursos
- Total de asignaturas
- Total de libros
- Total de estudiantes
- Recuento de ejemplares dependiendo del estado
- Total de ejemplares buenos
- Total de ejemplares malos
- Lista de ejemplares para reemplazar
- Gráfico de los estudiantes dados de alta en los últimos 12 meses
- Gráfico de las copias dadas de alta en los últimos 12 meses



## 6.3 Estudiantes

Siguiendo el flujo de la aplicación ahora nos iremos a la parte de estudiantes, esta vista muestra una lista de todos los estudiantes dados de alta.

En esta vista podemos realizar varias acciones:

- **Crear un nuevo estudiante:**
  - Para esto hará falta rellenar el siguiente formulario con la información correcta del estudiante:

[←](#) **Nuevo Estudiante**

**Identificación del Estudiante**

NIA: \*52556255

**Perfil del Estudiante**

Nombre: \*Hugo

Apellidos: \*Sanchis

Teléfono: \*665958525

Correo Electrónico: \*hugo@gmail.com

Crear

- Editar información del estudiante:

← Estudiante

Información Historial

Identificación del Estudiante

NIA:

96326632

Perfil del Estudiante

Nombre:

Hugos

Apellidos:

Sanchis

Teléfono:

666595858

Correo Electrónico:

hugolluissimarro03@gmail.com

Actualizar

Activar Windows

- Ver historial de libros del estudiante:

← Historial de Libros del Estudiante

Información Historial

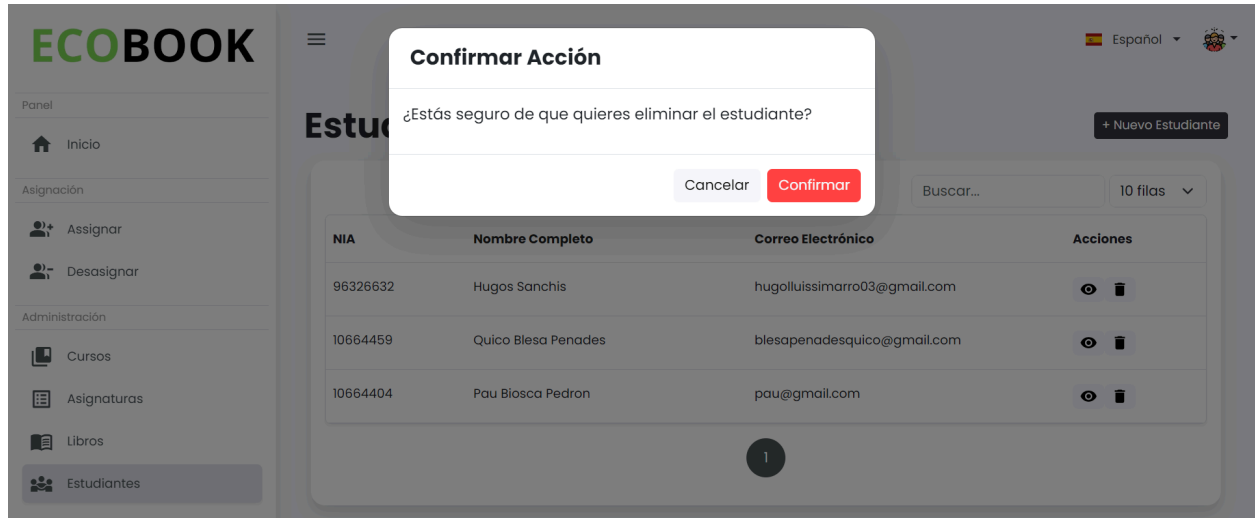
Buscar...

10 filas ▾

Libro	Código	Estado Inicial	Estado Final	Fecha Inicial	Fecha Final
MathematicsI	0340198645516	Nuevo	Bueno	24-05-2024	24-05-2024

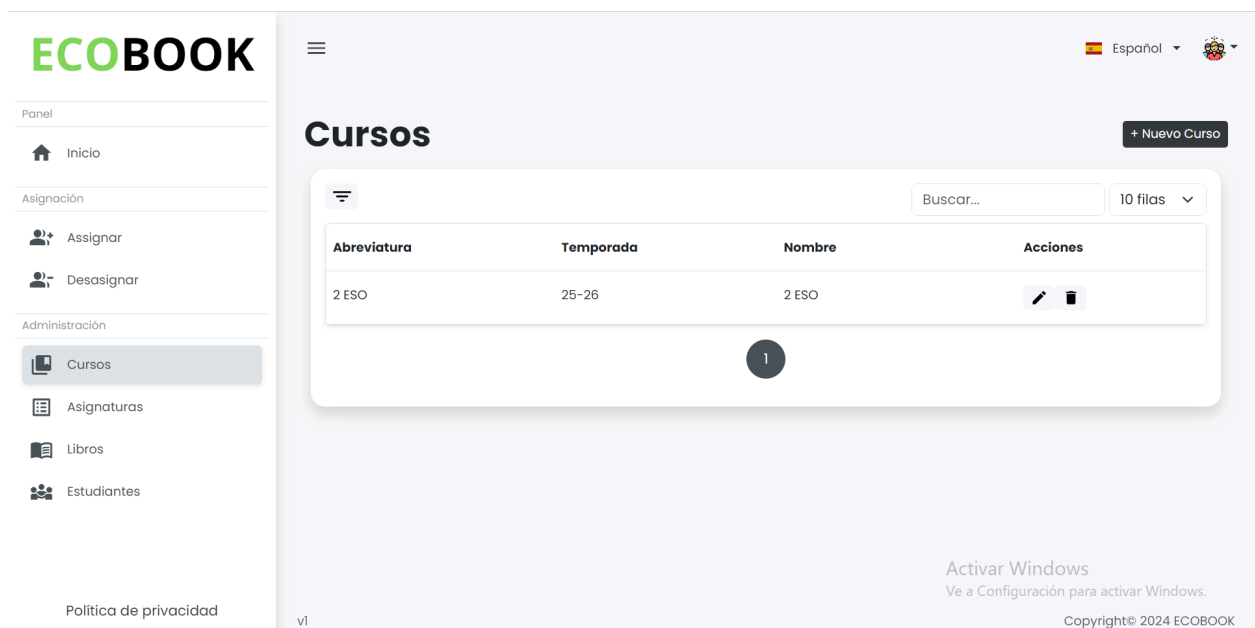
1

- **Eliminar estudiante:**



## 6.4 Cursos

Una vez creado el estudiante, siguiendo el flujo de la aplicación iremos a la vista de cursos donde nos saldrá una lista de los cursos disponibles, en caso de que no tengamos disponibles hay que crear uno para seguir con el flujo.



En esta vista podremos realizar las siguientes acciones:

- **Crear el curso:**
  - Para esto hará falta rellenar el siguiente formulario con la información correcta del curso:

**← Nuevo Curso**

**Información del Curso**

Abreviatura: *	2 ESO
Nombre: *	2 Educación secundaria obligatoria
Temporada: *	23/24

Crear

- **Editar la información de el curso:**

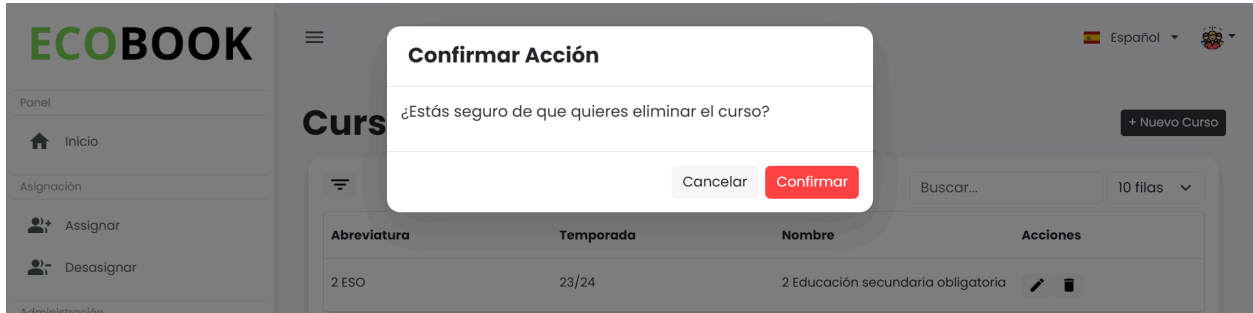
**← Editar Curso**

**Información del Curso**

Abreviatura: *	2 ESO
Nombre: *	2 Educación secundaria obligatoria

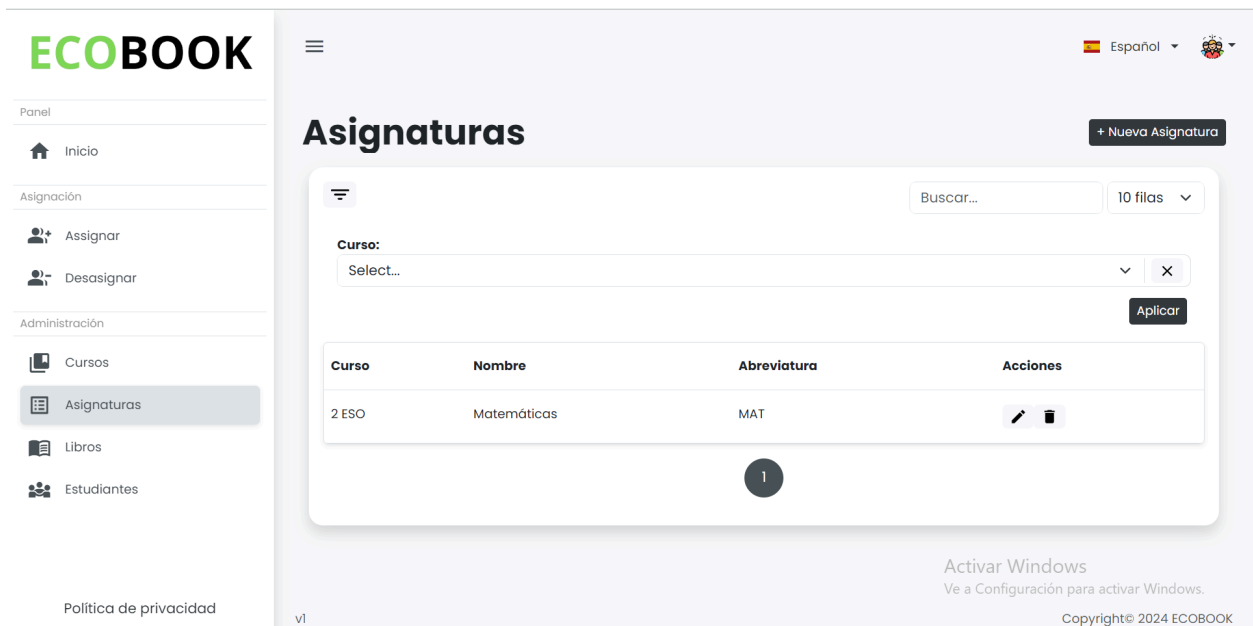
Actualizar

- **Eliminar el curso:**



## 6.5 Asignaturas

Una vez creado el curso seguiremos con las asignaturas para ese curso, así que nos dirigimos al apartado de asignaturas y aquí nos muestra otra lista con todas las asignaturas, y en esta tabla podremos filtrar asignaturas por curso, primero vamos a crear una asignatura y explorar las acciones de esta vista.



- Crear asignatura:

## ← Nueva Asignatura

### Subject Info

Nombre: \* Matemáticas

Abreviatura: \* MAT

Curso: \* 2 ESO

Crear

- Editar información de la asignatura:

## ← Editar Asignatura

### Informacion de la Asignatura

Nombre: \* Matemáticas

Abreviatura: \* MAT

Actualizar

- Eliminar asignatura:

**ECOBOOK**

Panel

- Inicio

Asignación

- Asignar
- Desasignar

Administración

- Cursos
- Asignaturas



**Asignaturas**

Confirmar Acción

¿Estás seguro de que quieres eliminar la asignatura?

Cancelar Confirmar

Curso Nombre Abreviatura Acciones

2 ESO	Matemáticas	MAT	 
-------	-------------	-----	---

1

Español




+ Nueva Asignatura

Buscar... 10 filas

## 6.6 Libros

Una vez tengamos creado el curso y la asignatura quedaría crear los libros para cada asignatura y para ello debemos irnos a la vista de libros.

The screenshot shows the 'Libros' (Books) management page in the ECOBOOK application. The interface includes a sidebar with navigation options: Inicio, Asignación, Administración, Libros (selected), and Estudiantes. The main content area displays a table of books with columns for Nombre, ISBN, Asignatura, Existencia, and Acciones. A single book entry is visible: 'Matemáticas' with ISBN '0-7645-2641-3' and 'Existencia' of 4. The table is paginated, showing 1 of 10 rows. A '+ Nuevo Libro' button is located in the top right corner. The footer includes a Windows activation notice and a copyright notice for ECOBOOK 2024.

Nombre	ISBN	Asignatura	Existencia	Acciones
Matemáticas	0-7645-2641-3	Matemáticas	4	  



En esta vista podemos hacer las siguientes opciones:

- **Crear libros:**
  - Para esto requiere haber creado previamente un curso y una asignatura.

[←](#) **Nuevo Libro**

**Información del Libro**

Nombre: *	Matemáticas
ISBN: *	0-7645-2641-3
Curso: *	2 ESO <span>▼</span> <span>✕</span>
Asignatura: *	Matemáticas <span>▼</span> <span>✕</span>
Existencia: *	30 <span>▲▼</span>

Crear

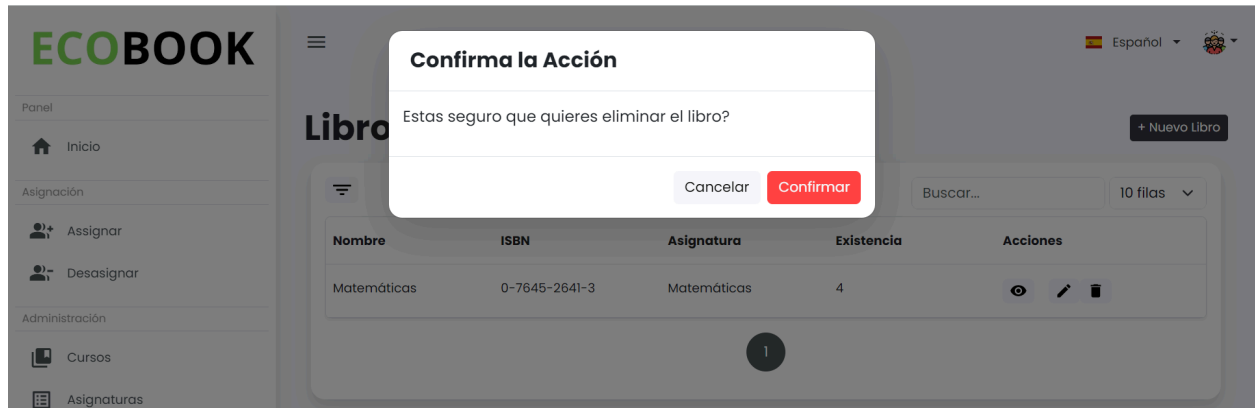
- **Editar información del libro:**

[←](#) **Editar Libro**

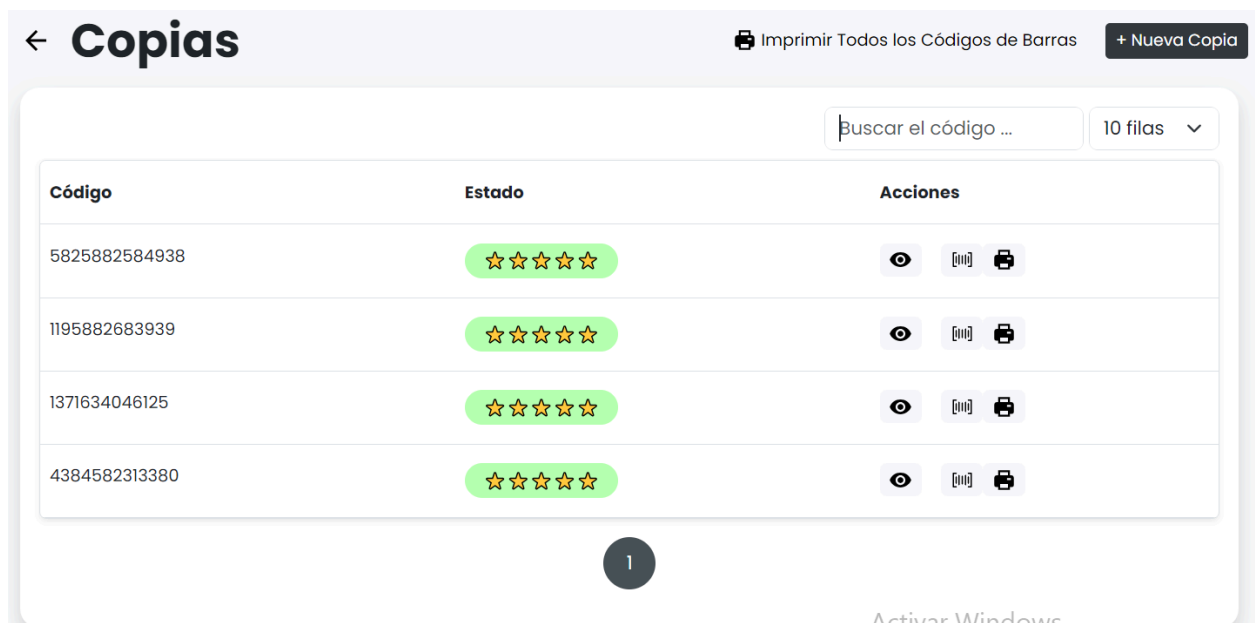
Nombre:	Matemáticas
ISBN:	0-7645-2641-3

Actualizar

- Eliminar libro (y con ello sus ejemplares):



- Ver los ejemplares de cada libro:



## 6.7 Ejemplares/Copias

Si entramos a la vista de ejemplares/copias, también podemos realizar múltiples acciones:

- **Ver información detallada de cada ejemplar/copia:**

← **Información de la Copia**

Información   Historial

**Información General**

**Código de Copia:**

5825882584938

**Curso:**

2 Educación secundaria obligatoria

**Asignatura:**

Matemáticas

**Nombre del Libro:**

Matemáticas

**Estado:**

★★★★★

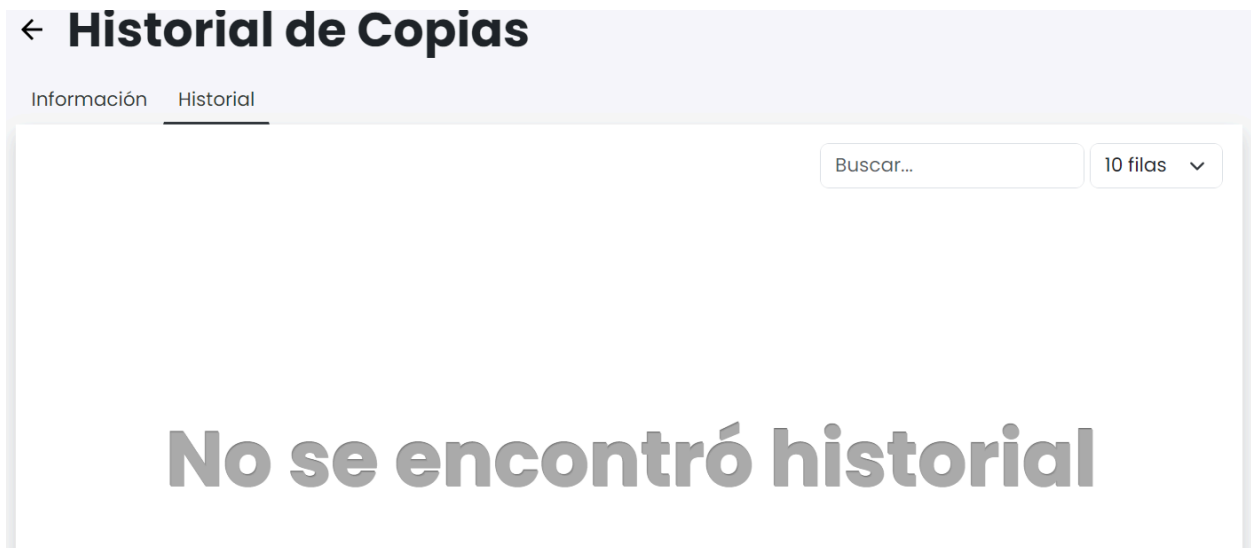
**Código de Barras**



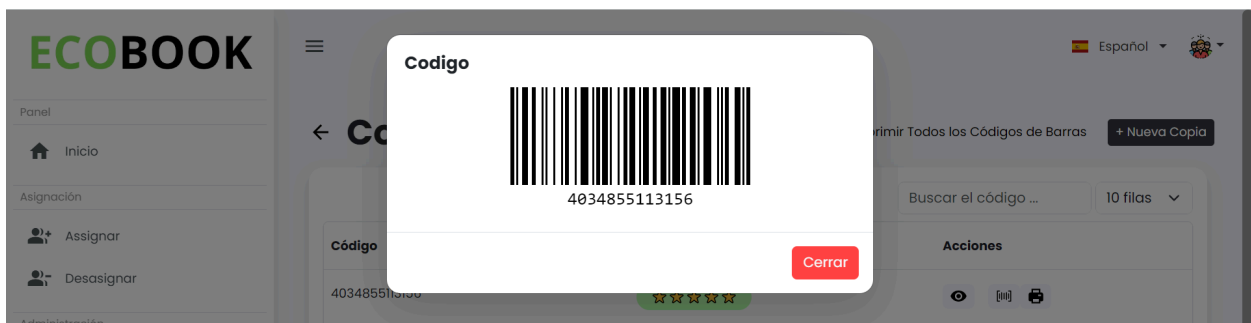
5825882584938

Activar Windows  
Ve a Configuración para activar Windows.

- **Ver historial de cada ejemplar/copia:**
  - En este caso este ejemplar/copia aún no lo ha tenido ningún alumno



- **Ver el código de cada ejemplar:**



- Imprimir tanto el código de cada ejemplar por individual, o todos los ejemplares:



La impresión sacará el siguiente PDF:



Esto será las etiquetas que pegaremos a cada ejemplar que tengamos, para así poder hacerle el seguimiento y saber información que haya almacenada sobre él.


Una vez todos los pasos anteriores estén cumplidos, quiere decir que tenemos toda la información guardada en nuestra base de datos. Ahora ya decidiremos qué hacer.


## 6.8 Asignar

Si es el periodo de inicio de curso deberemos asignarle a todos los alumnos los ejemplares que pertenezcan a las asignaturas que va a realizar el estudiante. Para ello nos dirigimos al apartado de asignar

### Asignar

**Identificación del Estudiante**

 Escanear con lector

 Escanear con cámara web

**NIA:**

**Asignación de Materias**

**Curso:**

Asignar

En esta vista deberemos introducir el nia del estudiante y elegir el curso que va a realizar. Cuando hayamos introducido estos datos. la aplicación automáticamente va a mostrar todas las asignaturas que puedes asignarle al estudiante en ese curso.

### Asignar

**Identificación del Estudiante**

 Escanear con lector

 Escanear con cámara web

**NIA:**

**Asignación de Materias**

**Curso:**

**Pendientes de asignar**

Matemáticas


**Asignados**

Selecciona alguna materia de la lista de la anterior

Asignar

El siguiente paso será pulsar sobre las asignaturas que queramos asignarle al estudiante e introducir el código del ejemplar que vayamos a darle. Para este paso he decidido facilitar el trabajo del usuario, dejándole utilizar lectores de códigos de barras o incluso la webcam para que el código lo escriba automáticamente

Cuando tengamos introducido el código en el formulario, solamente le daremos a confirmar y este automáticamente lo pasará a la columna de asignados. En el caso que nos engañemos y no hayamos asignado bien el ejemplar podemos corregirlo pulsando sobre la asignatura en la parte de Asignados y esta automáticamente volverá a la columna de Pendientes de asignar.



Cuando tengamos ya todas las asignaturas en la columna Asignados pulsaremos el botón asignar y automáticamente el sistema almacenará esta información en su base de datos. Esto registrará también información en el historial.

Si es el periodo de final de curso deberemos desasignar a todos los alumnos los ejemplares que pertenezcan a las asignaturas que han realizado. Para ello nos dirigimos al apartado de desasignar.



## 6.9 Desasignar

**ECOBOK**

Panel

[Inicio](#)

Asignación

[Asignar](#)

**[Desasignar](#)**

Administración

[Cursos](#)

[Asignaturas](#)

[Libros](#)

[Estudiantes](#)

[Política de privacidad](#)

Desasignar

Identificación del Estudiante

Escanear con lector

Escanear con cámara web

NIA: \*

Escribe el NIA del estudiante...

Siguiente

Activar Windows

Ve a Configuración para activar Windows.

Copyright © 2024 ECOBOOK

En esta vista solo deberemos introducir el nia del estudiante que vaya a realizar la devolución y pulsar el botón de siguiente.

**Desasignar**

Identificación del Estudiante

Escanear con lector

Escanear con cámara web

NIA: \*

96326632

Siguiente

Esto nos redirige a la lista de asignaturas que tiene pendiente de devolver el alumno.

# Desasignar

Estudiante: **Hugos Sanchis**

Libros pendientes de devolución

Escanear con lector

Escanear con cámara web

Matemáticas

Para hacerla devolución simplemente pulsamos el botón de la derecha de la asignatura y se abrirá una ventana donde deberemos poner obligatoriamente el estado en el que el libro ha sido devuelto y si queremos podemos añadir una observación.

**ECOBOOK**

Panel

Inicio

Asignación

Asignar

Desasignar

Administración

Cursos

Asignaturas

Libros

Estudiantes

Política de privacidad

Desasignar

Libros pendientes de devolución

Matemáticas

Matemáticas

Código

4034855113156

Estado

☒

☹

☺

😊

🌟

Observaciones

El libro está en perfectas condiciones.

Cancelar

Confirmar

Estudiante: **Hugos Sanchis**

Escanear con lector

Escanear con cámara web

Activar Windows

Ve a Configuración para activar Windows.

Copyright © 2024 ECOBOOK

Una vez pulsemos confirmar el ejemplar se registra en la base de datos como que está devuelto.

## 6.10 Otras vistas fuera de flujo

- **Perfil**

- Aquí el usuario de la aplicación puede cambiar su información como puede ser su imagen de perfil, el nombre, los apellidos o el número de teléfono.

**ECOBOOK**

Panel

- Inicio

Asignación

- Asignar
- Desasignar

Administración

- Cursos
- Asignaturas
- Libros
- Estudiantes

**Perfil**

Imagen de Perfil

Información básica del usuario

Nombre: Hugo Mascapitos

Apellidos: Sanchis

Teléfono: 666595858

Actualizar

- **Cuenta**

- Esta vista le servirá al usuario para cambiar su correo electrónico o cambiar su contraseña.

**ECOBOOK**

Panel

- Inicio

Asignación

- Asignar
- Desasignar

Administración

- Cursos
- Asignaturas
- Libros
- Estudiantes

**Cuenta**

Correo Electrónico

Correo Electrónico Actual: Escribe tu correo electrónico actual...

Nuevo Correo Electrónico: Escribe tu nuevo correo electrónico...

Repite el nuevo correo electrónico: Escribe tu nuevo correo electrónico...

Confirmar Cambio

Contraseña

Contraseña Actual: Escribe tu contraseña actual...

Nueva Contraseña: Escribe tu nueva contraseña...

Repite la nueva contraseña: Escribe tu nueva contraseña...

Cambiar Contraseña



## 7. CONCLUSIÓN

Al desarrollar una aplicación web como ECOBOOK mis conocimientos sobre el desarrollo de aplicaciones web se ha enriquecido bastante. He aprendido conceptos nuevos de programación así como el uso del framework REACT. Además gracias a ayudas externas como la del tutor o mi tutor de prácticas he hecho cosas que nunca podría imaginar, como la implementación de la Webcam para leer códigos de barras.

Me ha parecido un proyecto muy interesante para enriquecer mis conocimientos en el ámbito web. Iniciar un proyecto desde cero y gestionarlo de manera independiente es una de las experiencias que nunca había vivido y me ha aportado un aprendizaje significativo.

Quisiera darle las gracias al Instituto IES Dr. Lluís Simarro por brindarme esta oportunidad única y con un valor inexplicable.



## 8. RECURSOS

**Visual Studio Code:** Es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, completación inteligente de código, fragmentos y refactorización de código. También es personalizable, permitiendo a los usuarios cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo licencia de software privativo e incluye características personalizadas por Microsoft.

**HeidiSQL:** Es una herramienta para gestionar bases de datos, diseñada para trabajar con MySQL, MariaDB, PostgreSQL y Microsoft SQL Server. Permite crear y editar datos y estructuras, exportar datos y ejecutar consultas SQL. Es útil para administrar y visualizar bases de datos de manera eficiente.

**Node.js:** Es un entorno de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado para ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.

**GitHub:** Es una plataforma para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador, permitiendo la colaboración en equipo y el seguimiento de versiones. GitHub facilita la gestión de proyectos y la integración de las contribuciones de varios desarrolladores.

## 9. BIBLIOGRAFÍA

A continuación se describirán las páginas y documentación usada para la realización del proyecto.

- Documentación de react-bootstrap  
<https://react-bootstrap.netlify.app/>
- Dudas puntuales  
<https://es.stackoverflow.com/>
- Iconos Bootstrap  
<https://icons.getbootstrap.com/>
- Canva para realizar las miniaturas de las apariencias y logos:  
<https://www.canva.com/>
- Librerías externas  
<https://github.com/>