

Actividad | 1 | Pérdida de Autenticación y Gestión de Sesiones.

AUDITORÍA INFORMÁTICA.

Ingeniería en Desarrollo de Software.



TUTOR: JESSICA HERNANDEZ ROMERO.

ALUMNO: JONATHAN OSWALDO CARDENAS GARCIA.

FECHA: 24-octubre -2025

Tabla De Contenido

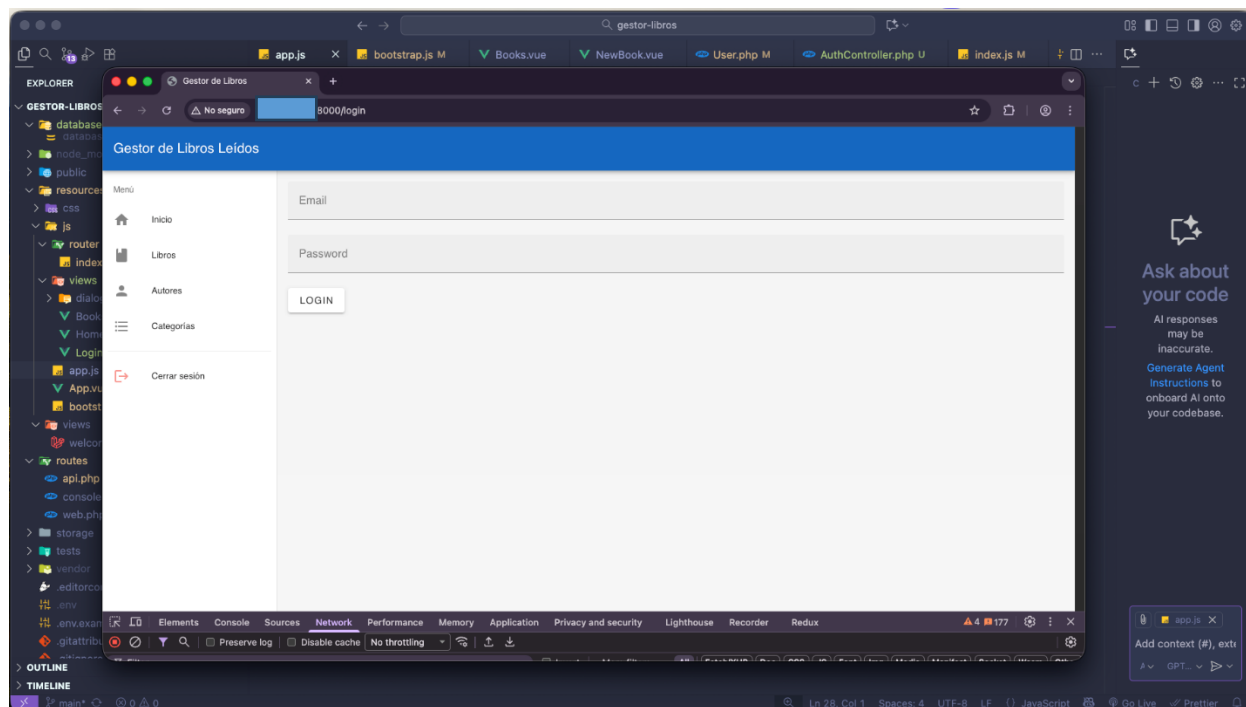
Tabla De Contenido	2
Desarrollo.....	3
Descripción del sitio web.....	3
Ataque al sitio	8
Referencias.....	13

Desarrollo

Descripción del sitio web

Figura 1

Pantalla de inicio

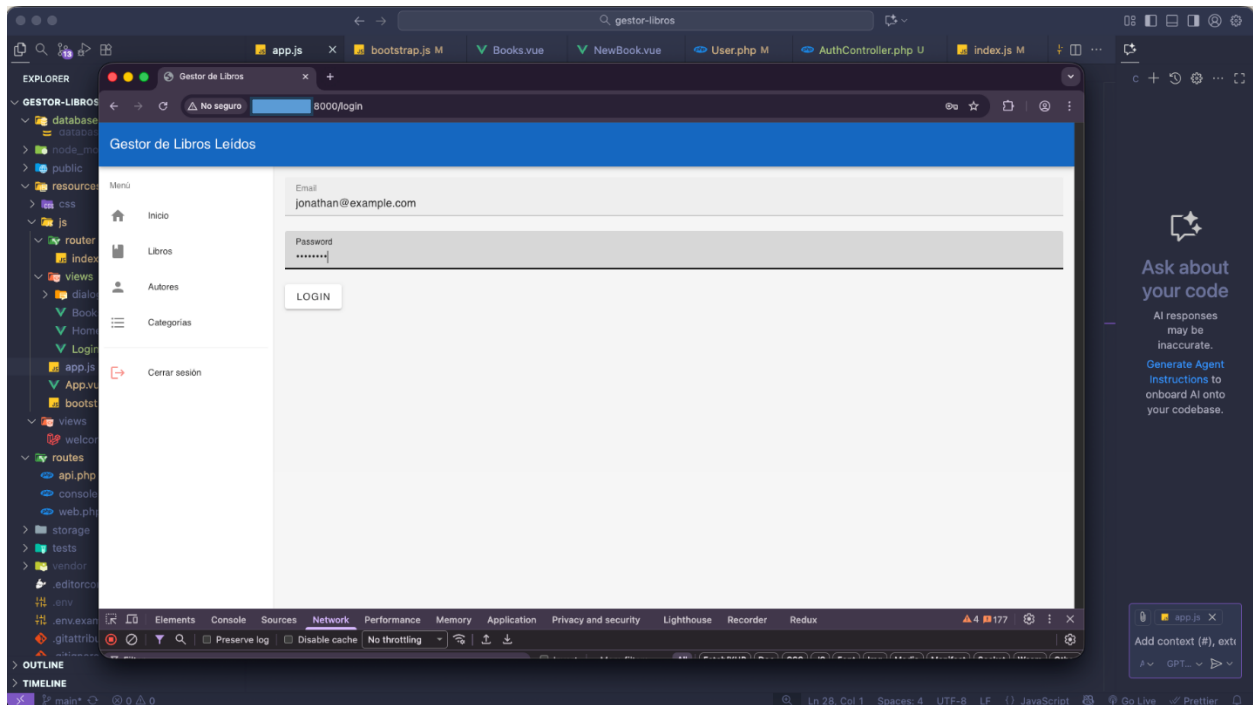


Nota. Este proyecto no lo realice para la carrera, es un proyecto que realice a manera de practica para aplicar conocimientos que ocupaba en mi trabajo, esta realizado con Laravel para back, está conectado a una base de datos SQL y de front esta realizado con Vue que no deja de ser JavaScript y en back PHP, para el diseño tiene Vuetify y los iconos de mdi, para las peticiones usa Axios, en si es un programa que yo tenía para gestionar mis libros leídos, pero no tenía autenticación, así que se la puse para poder hacer estas prácticas, ya tenía la migración de la tabla users así que solo agregue unos cuantos con una contraseña encriptada, como me daba error recupere la contraseña con tinker y esa puse, por los tiempos lo realice en local a pesar de que mi proyecto esta en GitHub

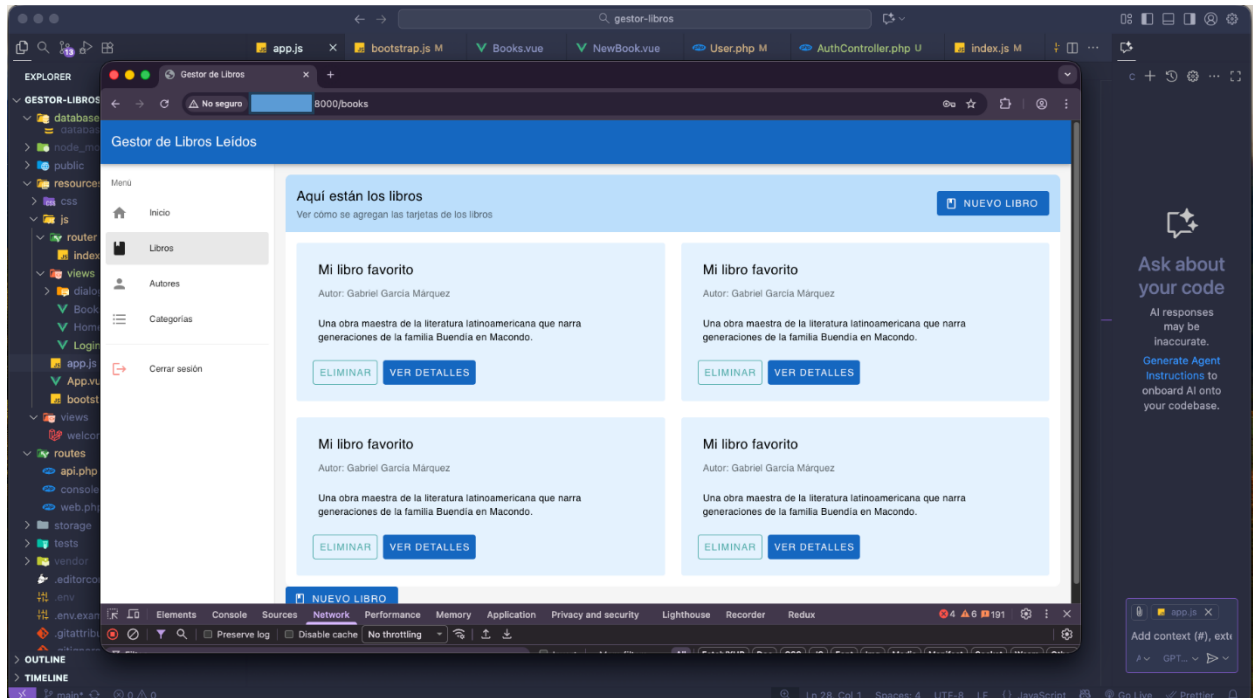
no me daba tiempo a desplegar, como se el funcionamiento pude hacer la actividad en local, agrego más imágenes para que se vea que es un proyecto bastante completo y que no me dio error.

Figura 2

Pantalla de login con datos



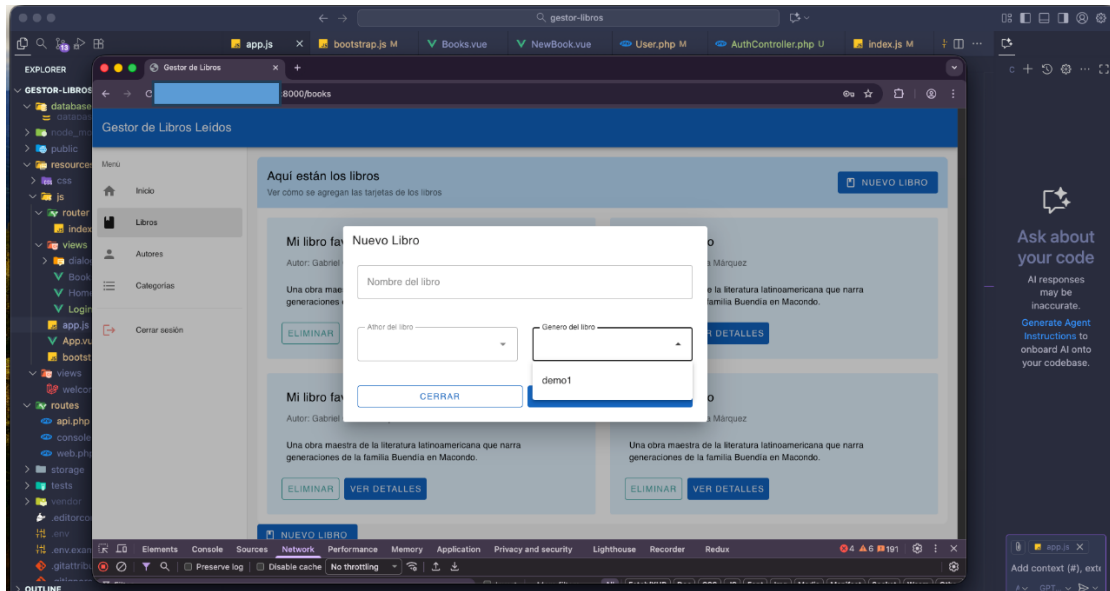
Nota. Se ve como cree una pantalla rápida para hacer el login, el botón dispara la petición axios y se va a api en back, autentica y regresa el token.

Figura 3*Inicio*

Nota. Esta es en si la pantalla inicial, si la autenticación es correcta hace un router.push y manda a esta vista, es la aplicación como tal, en el menú de la izquierda se ve que tengo un botón para cerrar seccion, esto es para hacer pruebas, se presiona y manda al login.

Figura 4

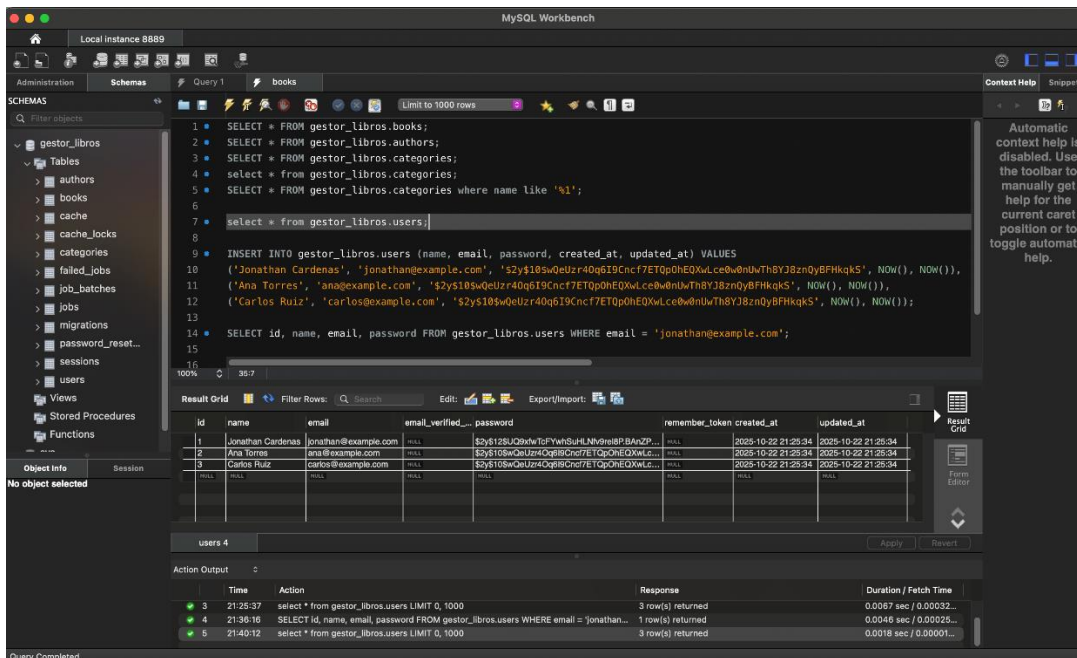
Una vista a un dialog



Nota. Esta una vista a un pequeño dialog para agregar un libro.

Figura 5

Agregar usuarios



Nota. En mi gestor de usuarios se ve como agrego y consulte los usuarios en mi tabla users.

Figura 6

Codigo y consola

The image shows a code editor with a file explorer on the left, a code editor in the center, and a terminal window on the right. The file explorer shows a project structure for 'gestor-libros' with folders like 'app', 'models', 'providers', and 'config'. The code editor displays the 'AuthController.php' file, which contains the following code:

```

1 <?php
2 namespace App\Http\Controllers;
3
4 use App\Http\Controllers\Controller;
5 use Illuminate\Http\Request;
6 use App\Models\User;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Support\Facades\Auth;
9
10 class AuthController extends Controller
11 {
12     public function register(Request $request)
13     {
14         $request->validate([
15             'name' => 'required|string|max:255',
16             'email' => 'required|email|unique:users,email',
17             'password' => 'required|string|min:6|confirmed'
18         ]);
19
20         $user = User::create([
21             'name' => $request->name,
22             'email' => $request->email,
23             'password' => Hash::make($request->password),
24         ]);
25
26         $token = $user->createToken('api-token')->plainTextToken;
27
28         return response()->json([
29             'user' => $user,
30             'token' => $token
31         ], 201);
32     }
33
34     public function login(Request $request)
35     {
36         $request->validate([
37             'email' => 'required|email',
38             'password' => 'required|string'
39         ]);
40
41         if (!Auth::attempt($request->only('email', 'password'))) {

```

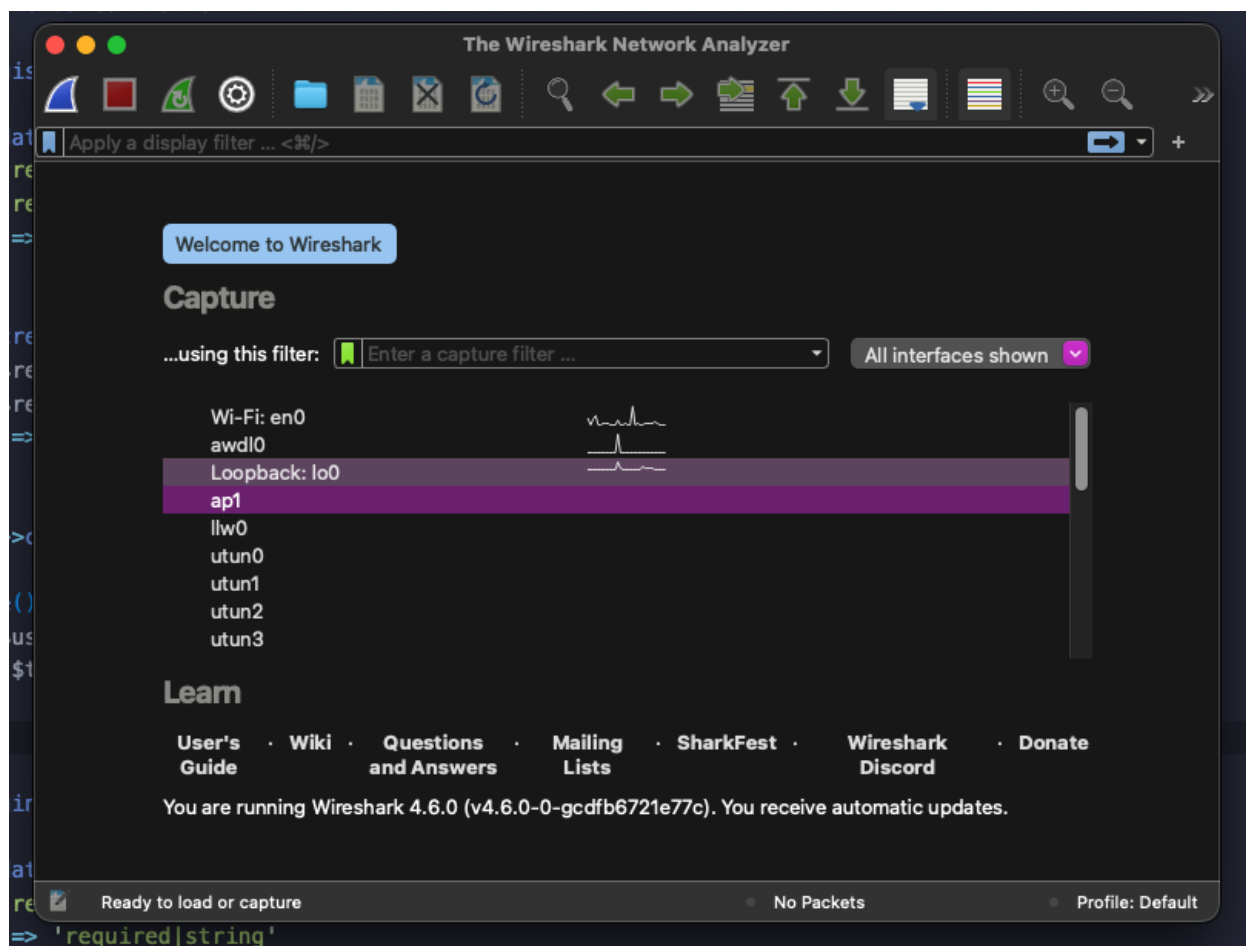
The terminal window shows the command 'php artisan serve --host=0.0.0.0 --port=8000' and its output, which includes the Laravel version (v2.2.0) and a list of files changed, inserted, and deleted. The output also shows the server running on 'http://0.0.0.0:8000'.

Nota. Aquí se ve mi código y como corri el proyecto para que se pudiera ver desde más dispositivos en la red, la verdad no pude conectar el teléfono por falta de tiempo pero era la idea.

Ataque al sitio

Figura 7

Pantalla de inicio



Nota. Esta es la pantalla de inicio donde seleccionamos que escuchar, originalmente quería conectarme con el teléfono y hacer la petición así se podría ver desde wifi pero al ser local se puede ver desde loopback.

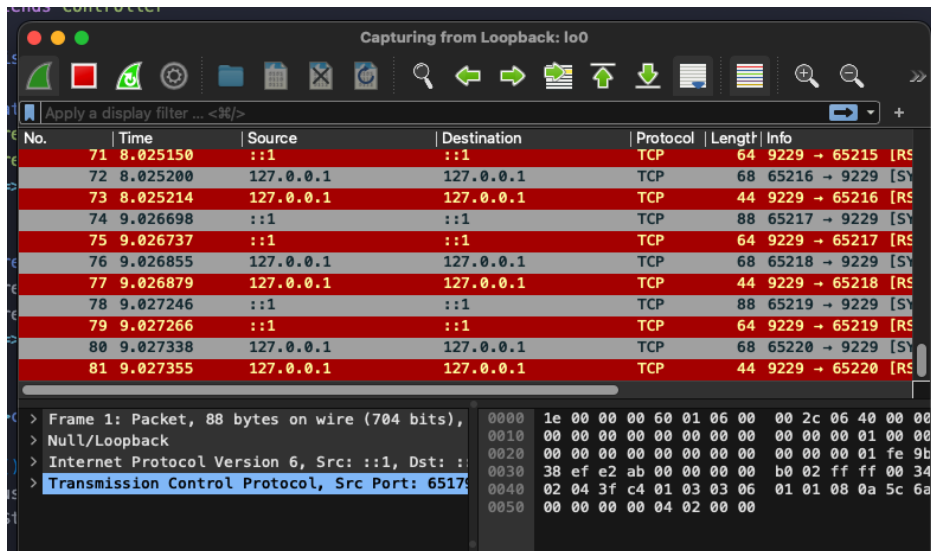
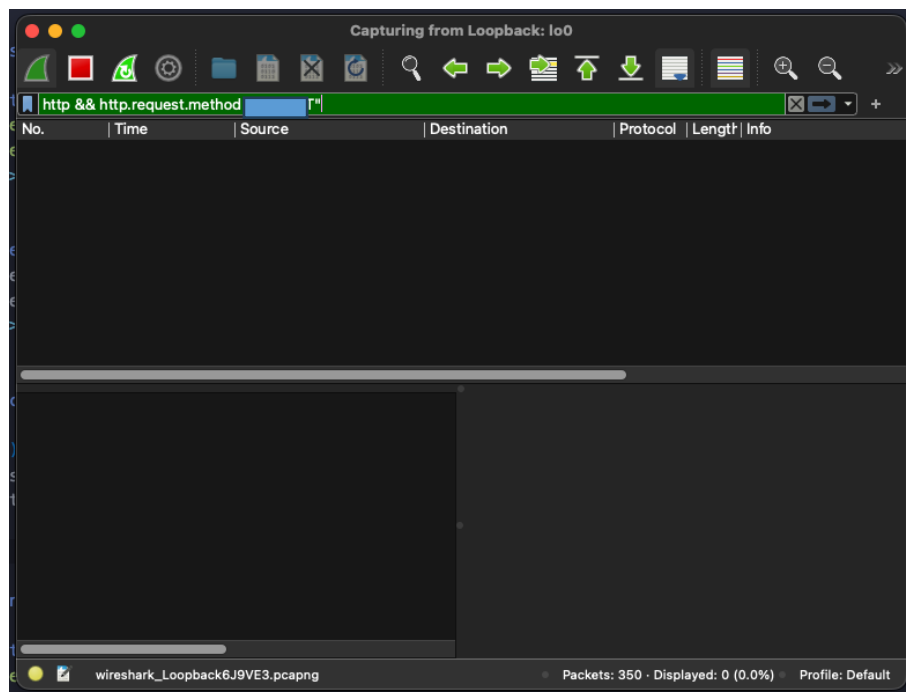
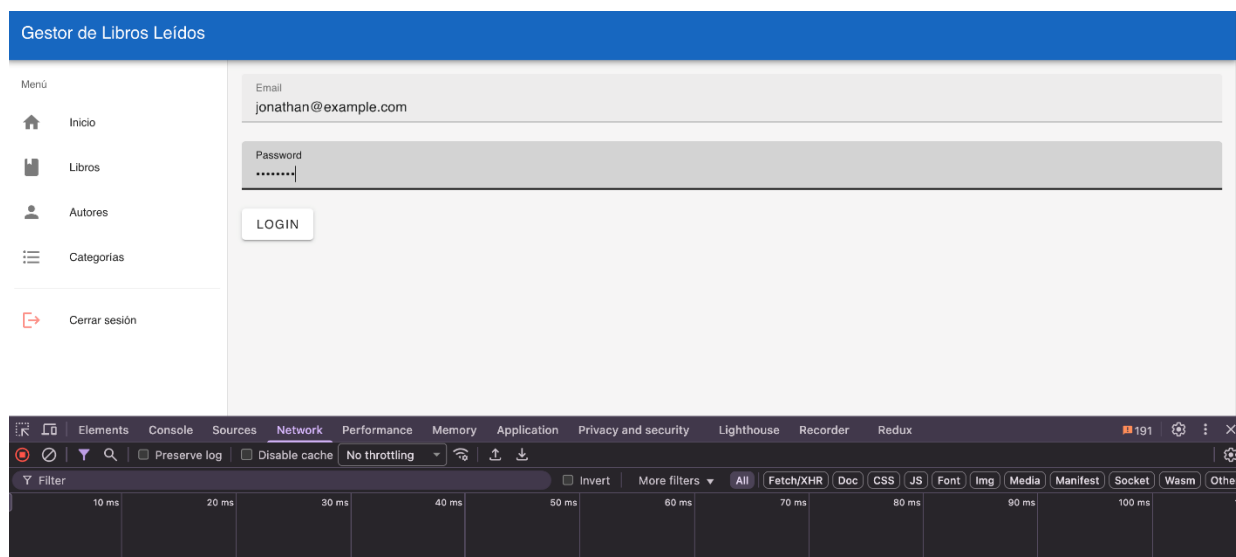
Figura 8*Escuchando la conexión**Nota.* Se ve todo lo que pasa por aquí.**Figura 9***Aplicamos el filtro**Nota.* Aplicamos el filtro para escuchar la petición que aremos.

Figura 10

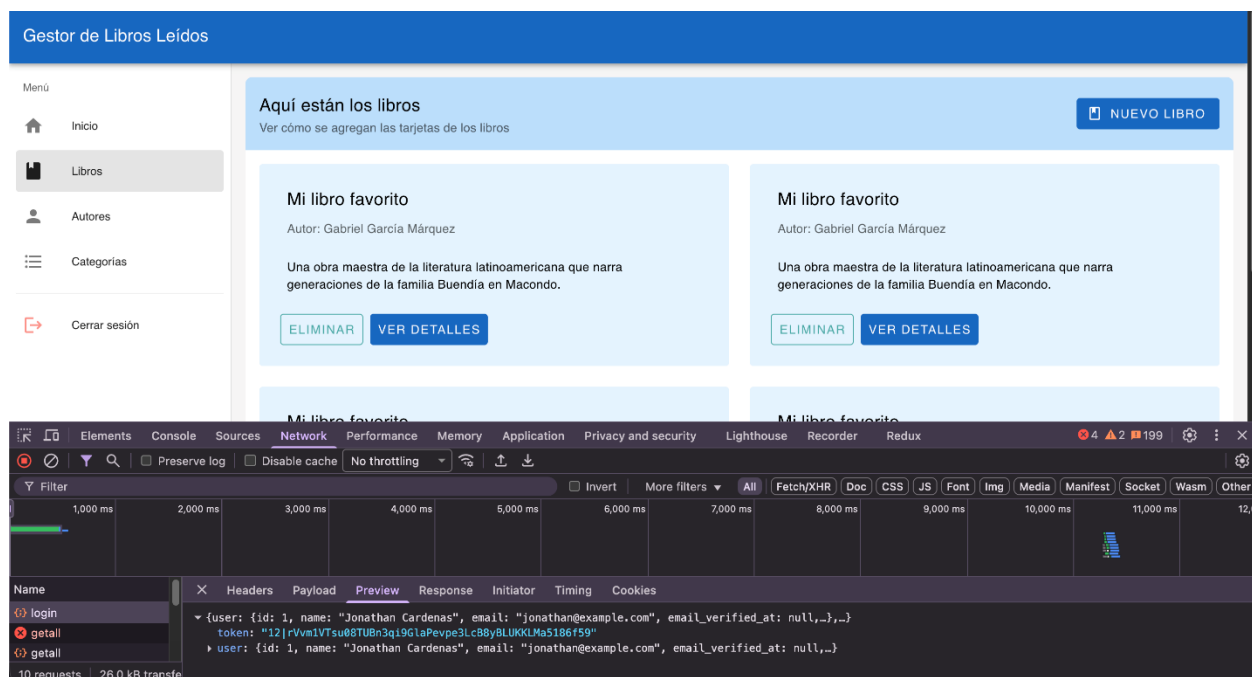
Regresamos a la pagina y ponemos las credenciales



Nota. Esta una vista se aprecia que se ponen las credenciales que están en la base de datos.

Figura 11

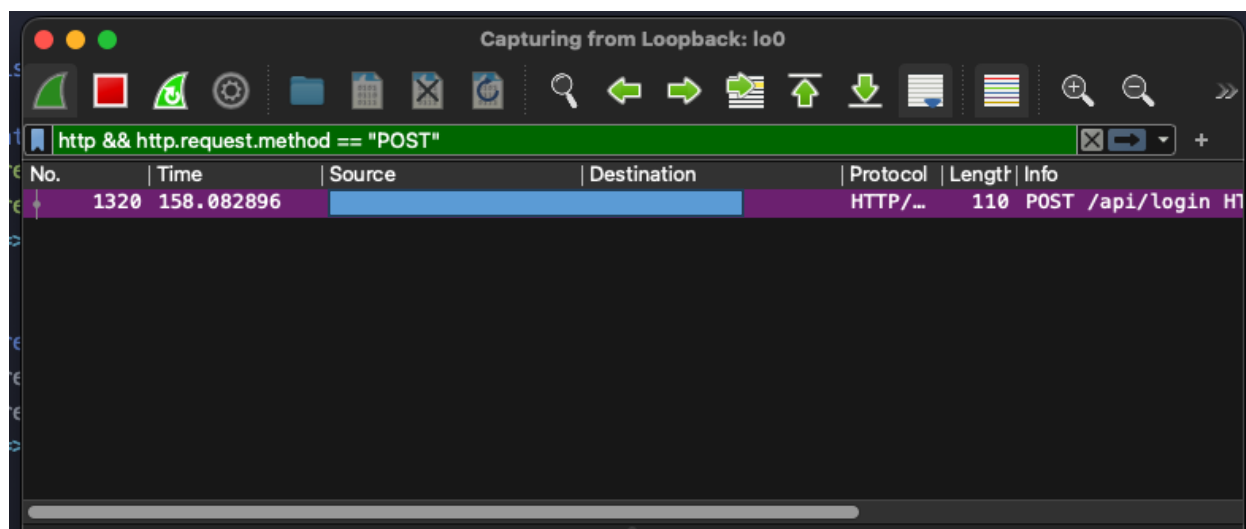
Petición realizada



Nota. Se ve que la petición se realizó con éxito y nos mandó a la otra vista.

Figura 12

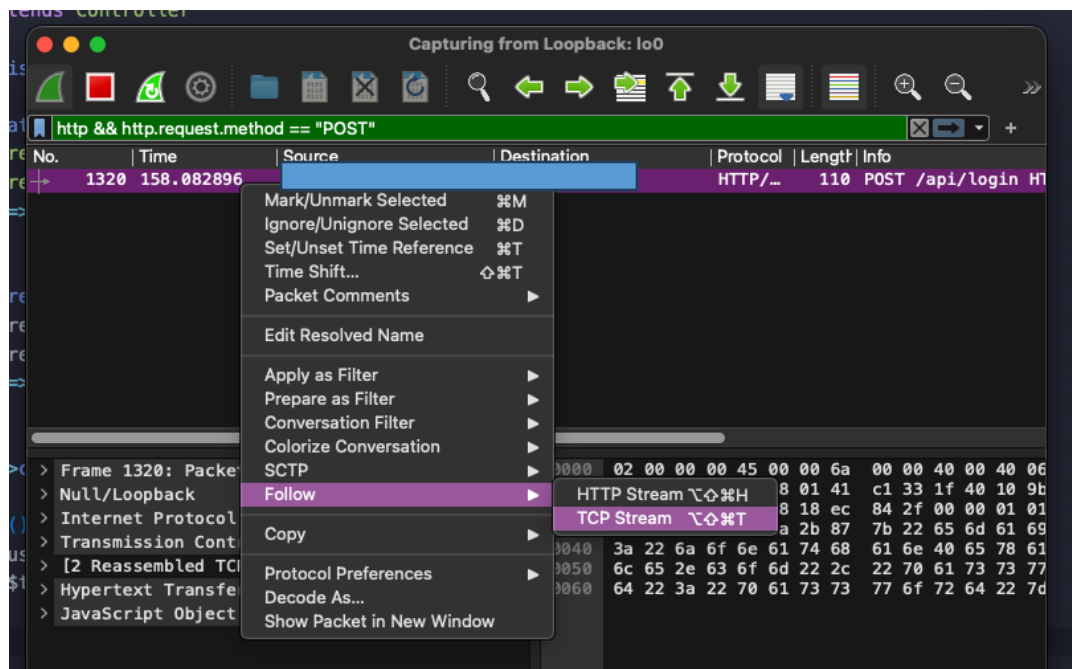
Vemos la petición en la aplicación



Nota. La aplicación con el filtro ve la petición que se realizó.

Figura 13

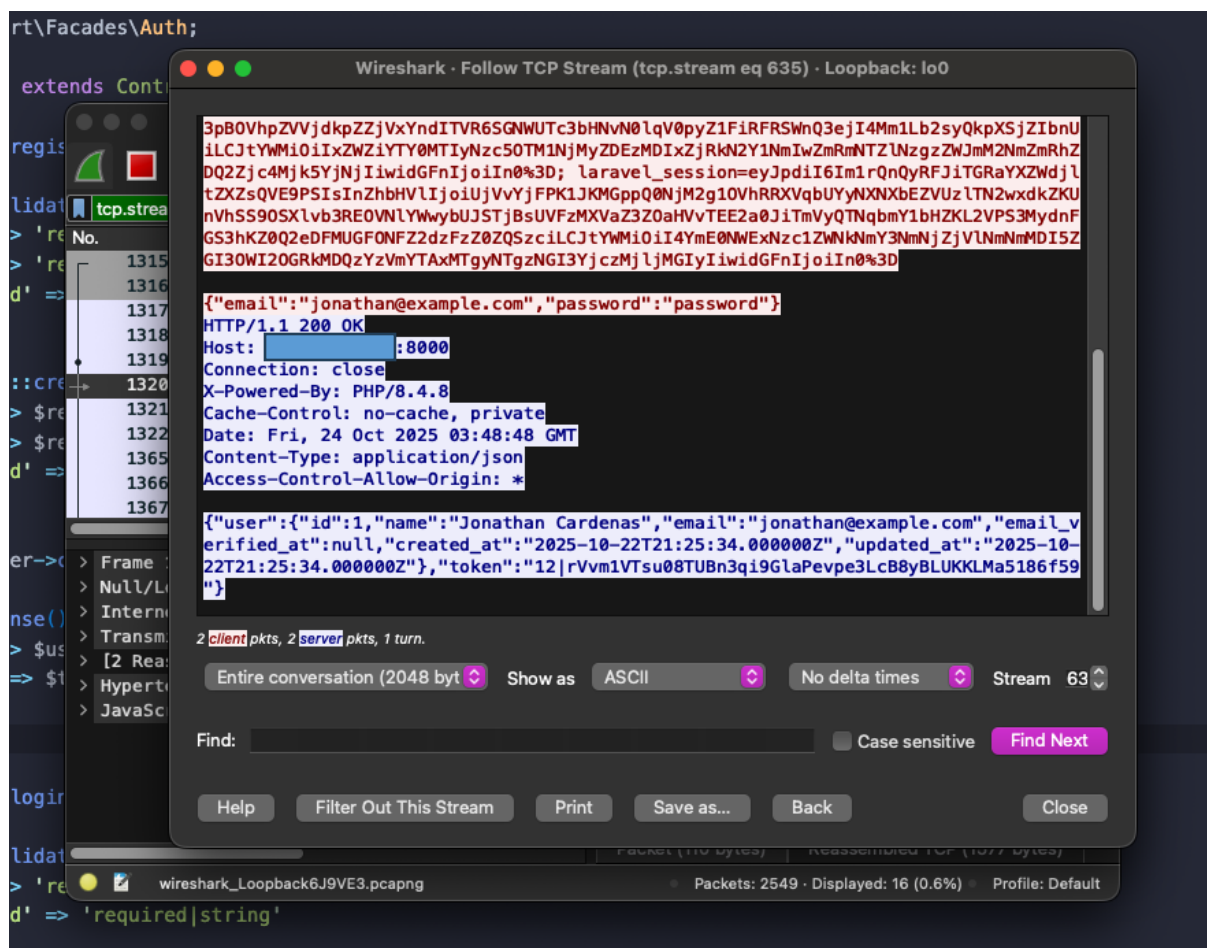
Abrimos lo que se escuchó sobre la petición



Nota. Se aprecia el cómo se abre.

Figura 14

Detalle



Nota. Se aprecian las credenciales con las que iniciamos sesión.

Referencias

<https://github.com/CardinalSG/JonathanCardenasAplicacionesBiometricas>

Me dio unos errores pero las 3 actividades están en este repo, cuando solucione el problema las cambiare a su propio repo.