

Actividad | 3 | Cross Site Scripting (XSS).

AUDITORÍA INFORMATICÁ.

Ingenieria en Desarrollo de Software.



TUTOR: JESSICA HERNANDEZ ROMERO.

ALUMNO: JONATHAN OSWALDO CARDENAS GARCIA.

FECHA: 24-octubre -2025

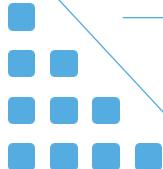


Tabla De Contenido

Tabla De Contenido	2
Introducción	3
Descripción	4
Justificación	5
Desarrollo I	6
Descripción del sitio web.....	6
Ataque al sitio	11
Desarrollo Actividad II	16
Ataque al sitio	16
Desarrollo III.....	25
Ataque al sitio	25
Conclusión	30
Referencias.....	30

Introducción

El entender la importancia de los datos, las vulnerabilidades de un sistema y todos los temas relacionados con la seguridad informática es crucial hoy en día ya que en un mundo tan conectado como en el que actualmente vivimos el perder, filtrar o manipular datos son cosas que pueden pasar en el día a día y afectara a una cantidad inimaginable de personas el hacer mal uso de los mismo, por eso en este trabajo se contemplan las tres actividades realizadas, la primera tiene una contextualización de un proyecto desarrollado por mi mismo que es una aplicación de gestión de libros leído, en la primera parte se ve el tráfico de la página donde desde el login podemos obtener las credenciales como correo y contraseña de una página que no es segura, la segunda actividad se realizó un laboratorio para cambiarnos los permisos, eliminar un usuario y entender cómo funciona la deserialización, en esta tercera actividad se observa el tráfico de nuestro primer proyecto, luego se interviene y se modifican las credenciales del login para mandar credenciales distintas, una incorrecta y otra correcta para observar esa interacción con la página y las respuestas.

Descripción

Este trabajo constara del desglose de los dos primeros a manera de resumen y recapitulación, el primer trabajo vimos el trafico de una pagina web que es de nuestra autoría, entramos a un login, proporcionamos credenciales y vimos como al ver este tráfico podríamos encontrar la petición y ver dichas credenciales en texto plano, claro siempre y cuando la página no tuviera seguridad, el ver estas credenciales era la actividad número uno, la segunda constaba de hacer un laboratorio donde entrabamos a una página, nos logueábamos y por el método de deserialización tomábamos la cookie, la decodeábamos a url, a base 64, modificábamos permisos de administrador, volvíamos a reconstruirla e interceptábamos la petición para meter nuestra cookie modificada y tener los permisos suficientes para eliminar un usuario, la actividad culminaba con el laboratorio en finalizado al eliminar al usuario Carlos y esta tercera actividad consta de interceptar el login de nuestra página y mandar distintas credenciales para obtener distintas respuestas.

Justificación

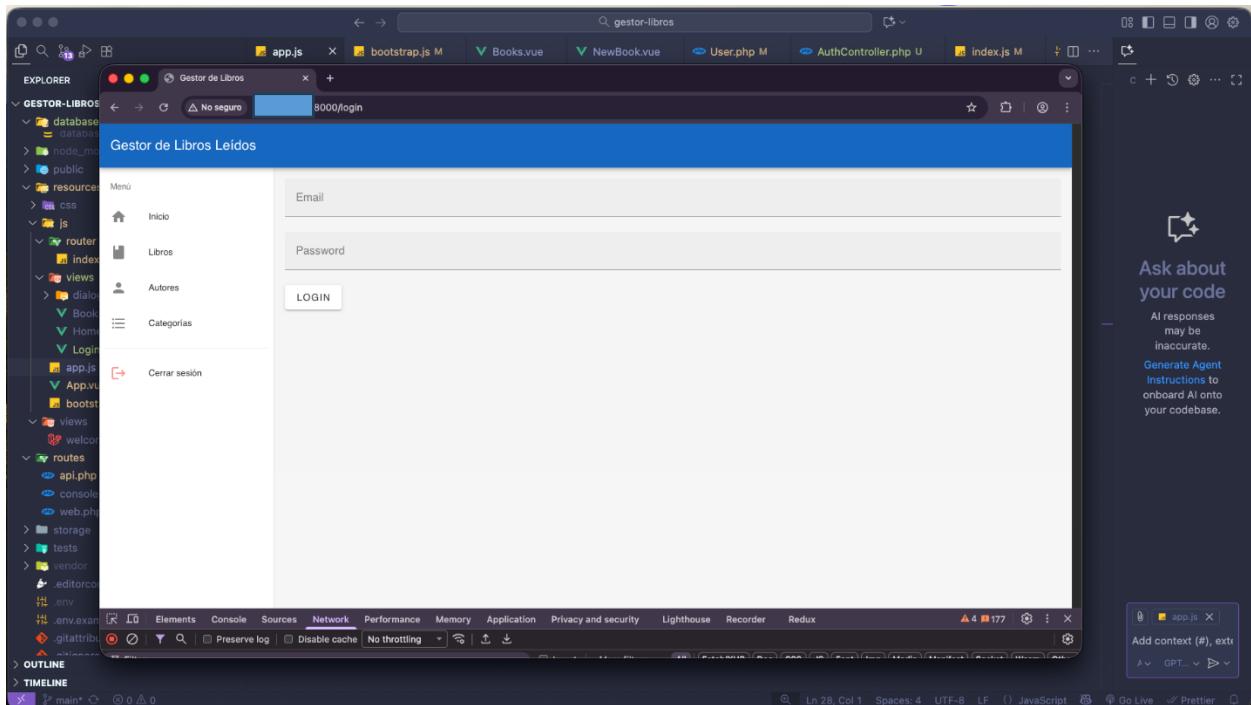
La justificación de este trabajo radica en la importancia de entender porque la información que manejamos en los sistemas se tiene que resguardar bien y con los mecanismos correctos, ya que en el mundo de hoy en día en el que todo está conectado y la información que muchas veces tiene partes sensibles fluye de un lado a otro y termina en las manos equivocadas donde pueden hacer un mal uso de la misma, partiendo desde esta primicia, al entender cómo se puede vulnerar una página podremos generar conciencia y mejores mecanismos de defensa que partirían desde unas buenas practicas para el desarrollo que al final de cuentas es donde impacta todo, porque si desarrollamos mal podríamos dejar sistemas que usan muchas personas de una forma vulnerable así que entender las formas más comunes de vulnerar y manipular esta información nos ayudara a ver la problemática desde otro punto de vista más objetivo y consciente de la situación.

Desarrollo I

Descripción del sitio web

Figura 1

Pantalla de inicio

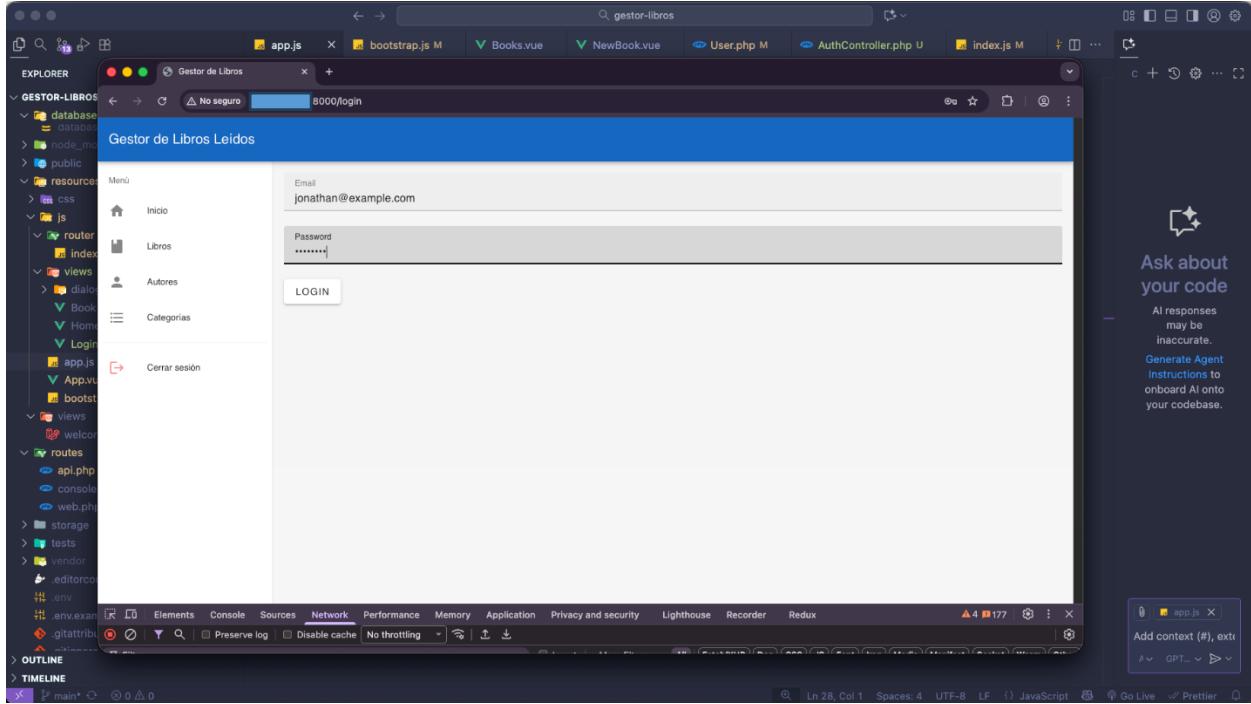


Nota. Este proyecto no lo realice para la carrera, es un proyecto que realice a manera de practica para aplicar conocimientos que ocupaba en mi trabajo, esta realizado con Laravel para back, esta conectado a una base de datos SQL y de front esta realizado con Vue que no deja de ser JavaScript y en back 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 autentificación, 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 recuperé 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

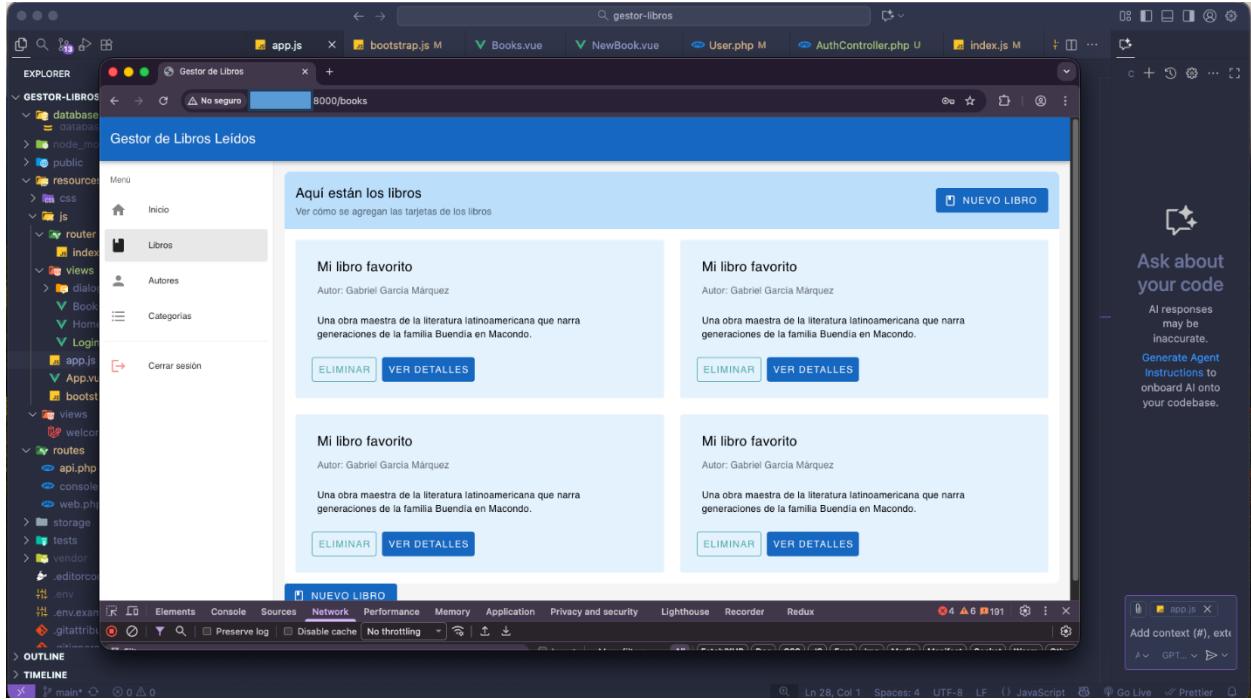
Pantalla de login con datos



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

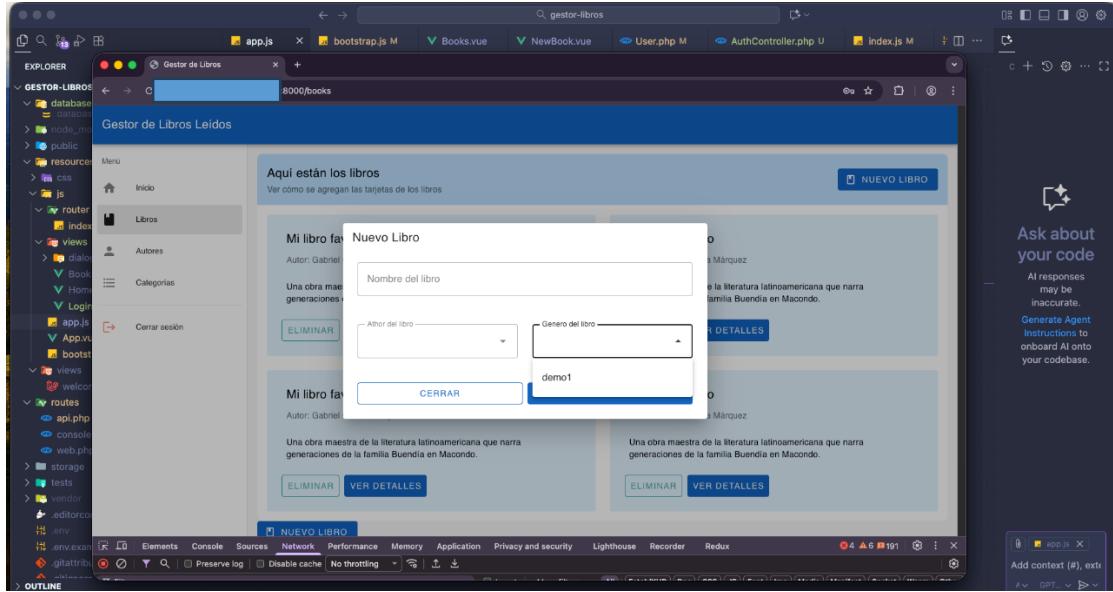
Inicio



Nota. Esta es en si la pantalla inicial, si la autentificación 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 sección, esto es para hacer pruebas, se presiona y manda al login.

Figura 4

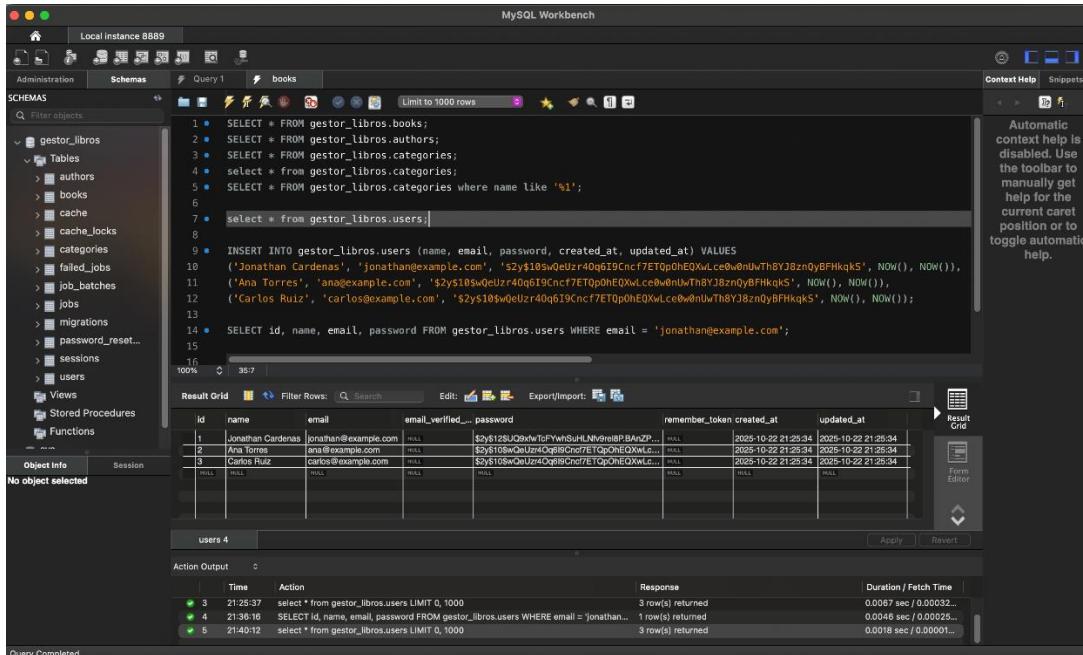
Una vista a un dialog



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

Figura 5

Agregar usuarioss



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

Figura 6

Código y consola

The screenshot shows a code editor interface with several files open. On the left, the file structure is visible under 'EXPLORER' for a project named 'GESTOR-LIBROS'. The 'Http\Controllers' folder contains 'AuthController.php', which is the active file. The code in 'AuthController.php' is a Laravel controller for authentication, including methods for registration and login. To the right of the code editor is a terminal window titled 'php' showing logs from a Laravel application. The logs indicate requests for various routes like '/favicon.ico', '/api/login', and '/api/categories/getall', along with timestamps and response times. The terminal also shows the command 'php artisan serve --host=0.0.0.0 --port=8000'.

```

<?php
namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    public function register(Request $request)
    {
        $request->validate([
            'name' => 'required|string|max:255',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|string|min:6|confirmed'
        ]);

        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
        ]);

        $token = $user->createToken('api-token')->plainTextToken;

        return response()->json([
            'user' => $user,
            'token' => $token
        ], 201);
    }

    public function login(Request $request)
    {
        $request->validate([
            'email' => 'required|email',
            'password' => 'required|string'
        ]);

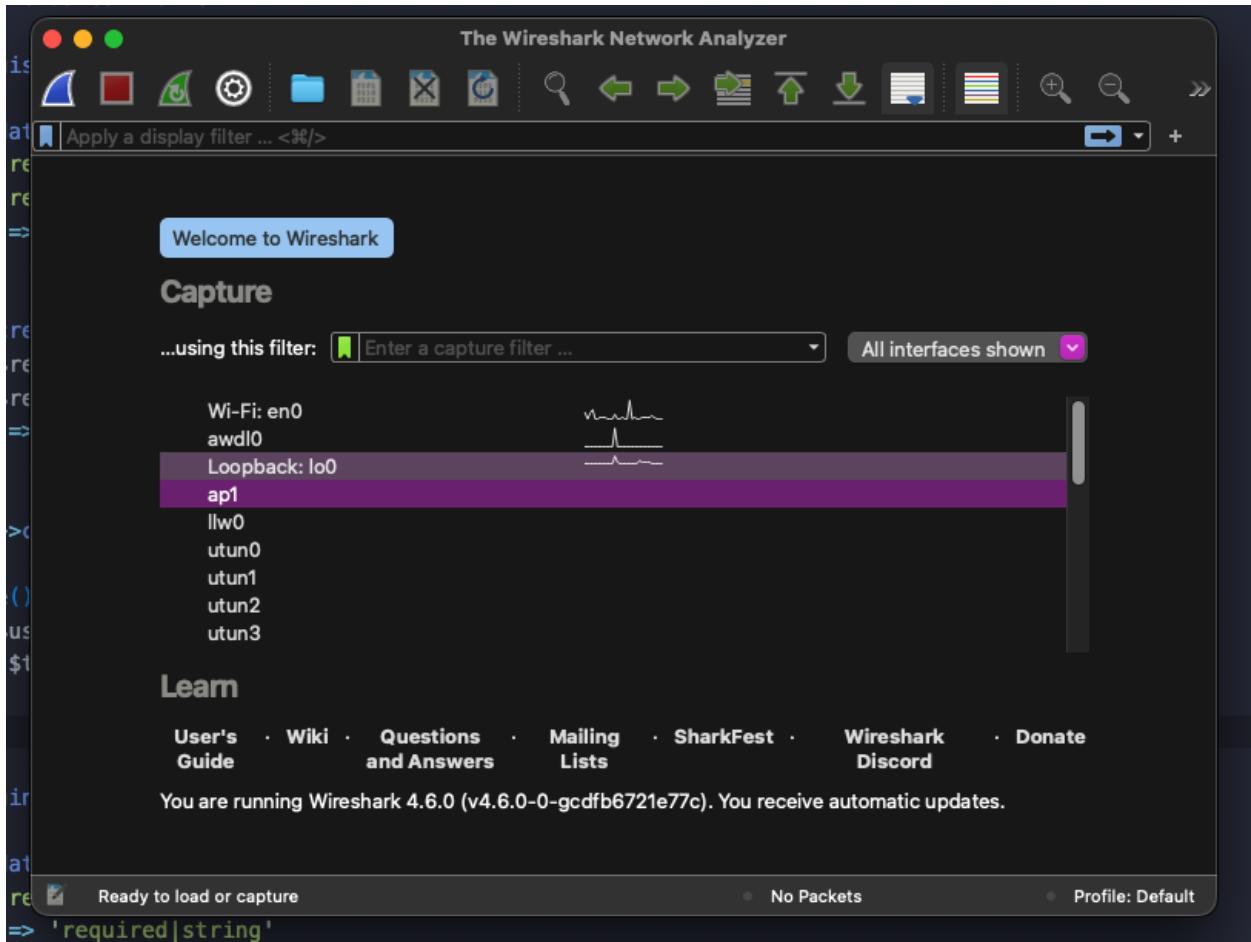
        if (!Auth::attempt(['$request->only('email','password')])) {
    
```

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

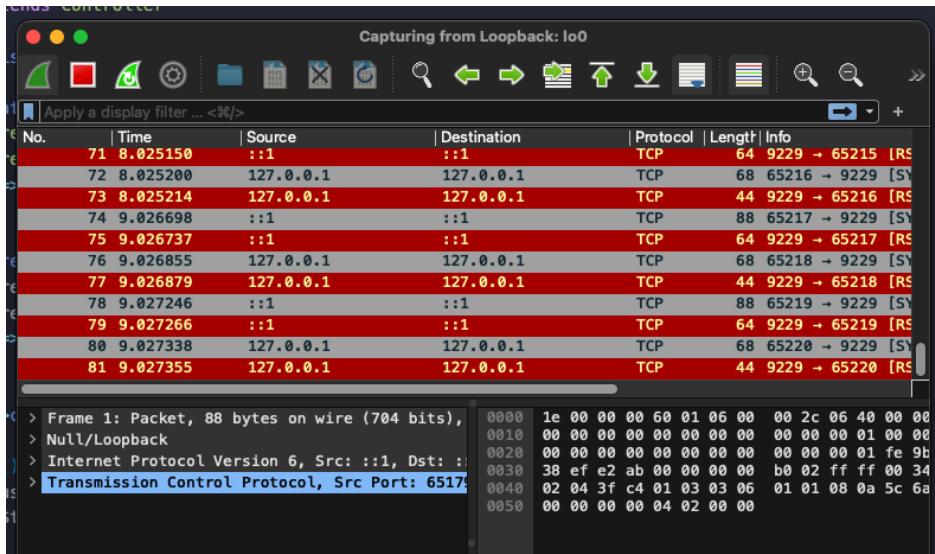
Pantalla de inicio



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

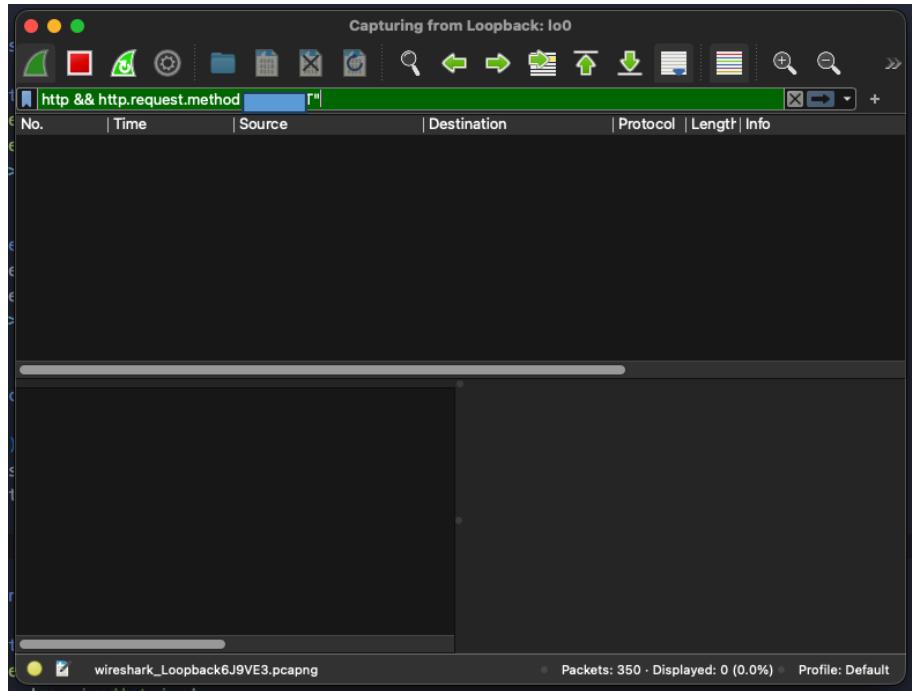
Escuchando la conexión



Nota. Se ve todo lo que pasa por aquí.

Figura 9

Aplicamos el filtro



Nota. Aplicamos el filtro para escuchar la petición que queremos.

Figura 10

Regresamos a la pagina y ponemos las credenciales

The screenshot shows the 'Gestor de Libros Leídos' application. On the left, a sidebar menu includes 'Inicio', 'Libros' (which is highlighted), 'Autores', 'Categorías', and 'Cerrar sesión'. The main content area has two input fields: 'Email' with 'jonathan@example.com' and 'Password' with '.....'. Below them is a 'LOGIN' button. At the bottom, a developer tools Network tab shows a single request labeled 'login' with a status of '191'.

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

Figura 11

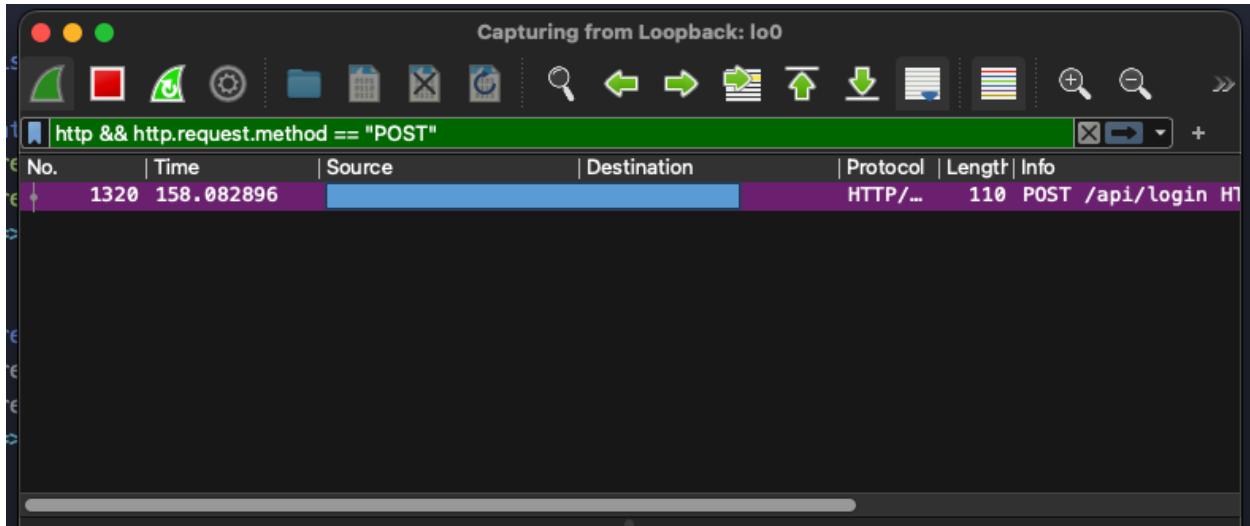
Petición realizada

The screenshot shows the 'Gestor de Libros Leídos' application. The sidebar menu is identical to Figure 10. The main content area displays two book entries: 'Mi libro favorito' by Gabriel García Márquez and another 'Mi libro favorito' entry. Both have 'VER DETALLES' buttons. At the bottom, a developer tools Network tab shows a successful 'login' request with a status of '4' and a response body containing JSON data for a user named 'Jonathan Cardenas'.

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

Figura 12

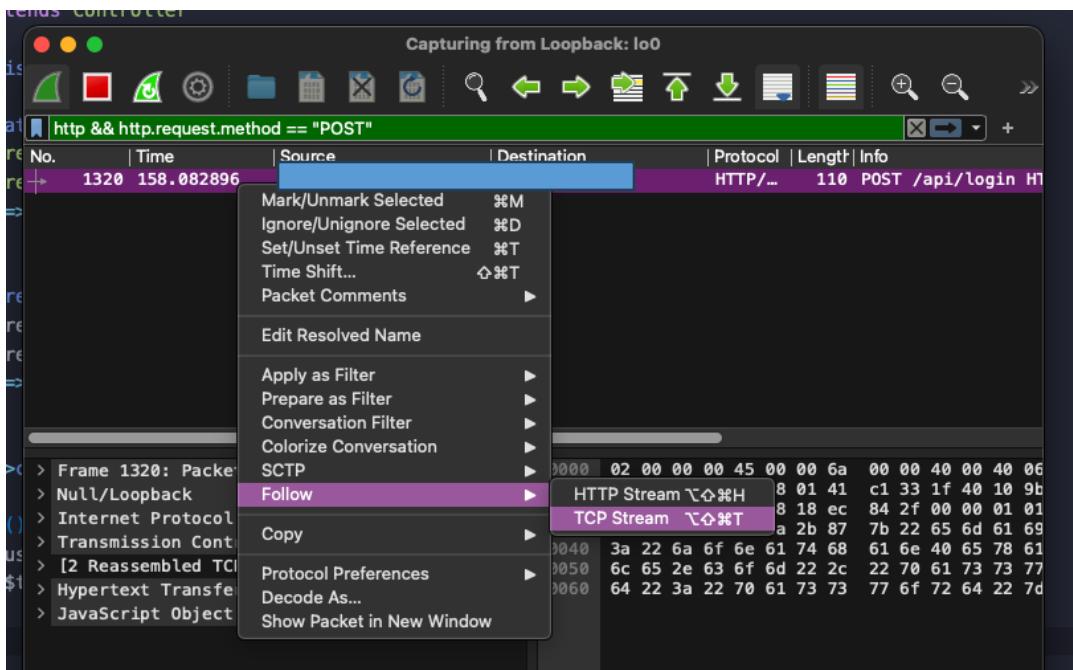
Vemos la petición en la aplicación



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

Figura 13

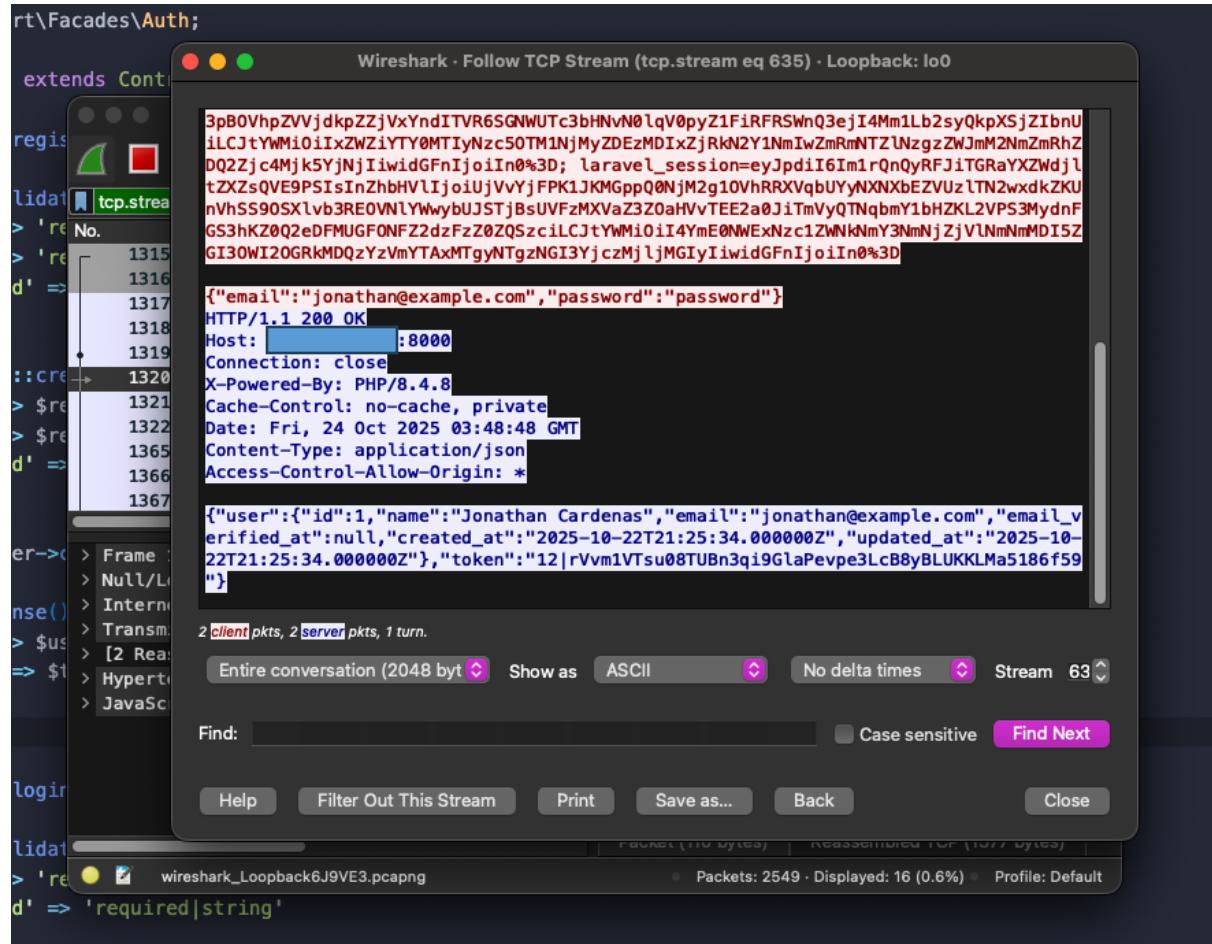
Abrimos lo que se escuchó sobre la petición



Nota. Se aprecia el cómo se abre.

Figura 14

Detalle



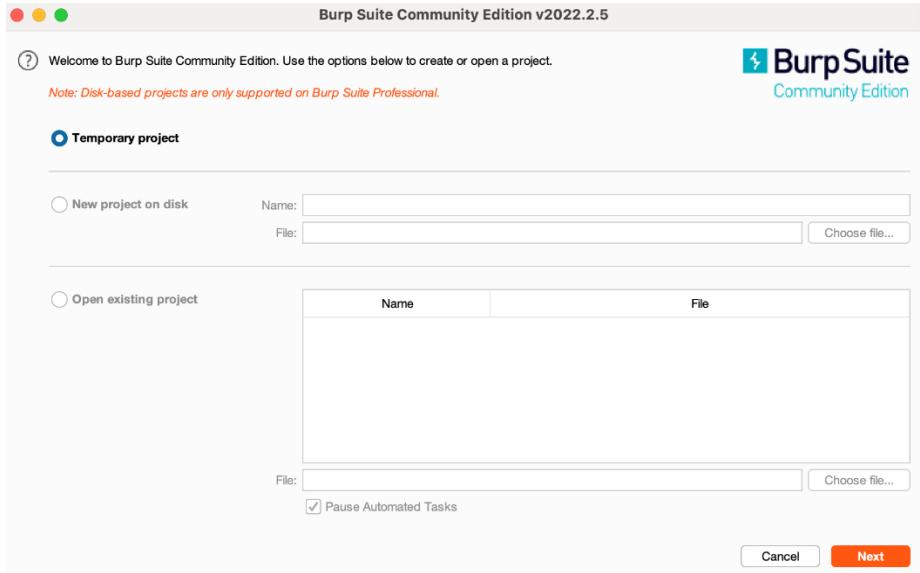
Nota. Se aprecian las credenciales con las que iniciamos sesión.

Desarrollo Actividad II

Ataque al sitio

Figura 45

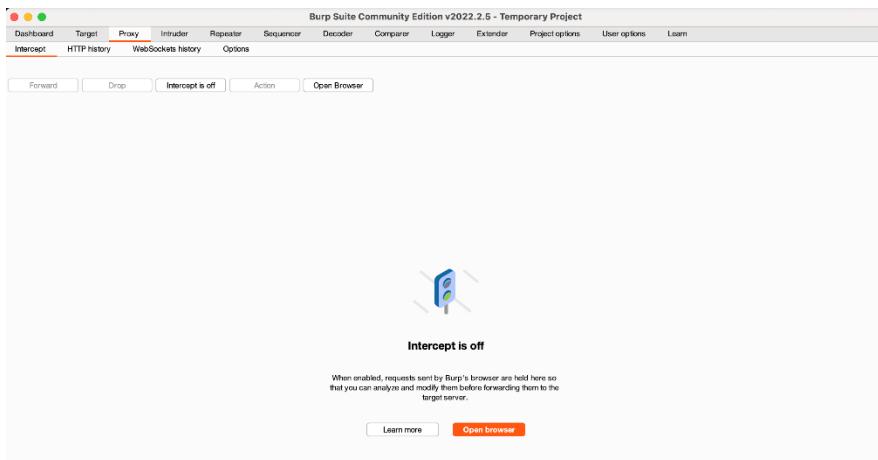
Primera pantalla



Nota. Aquí se ve la primera pantalla que sale cuando abres el programa.

Figura 16

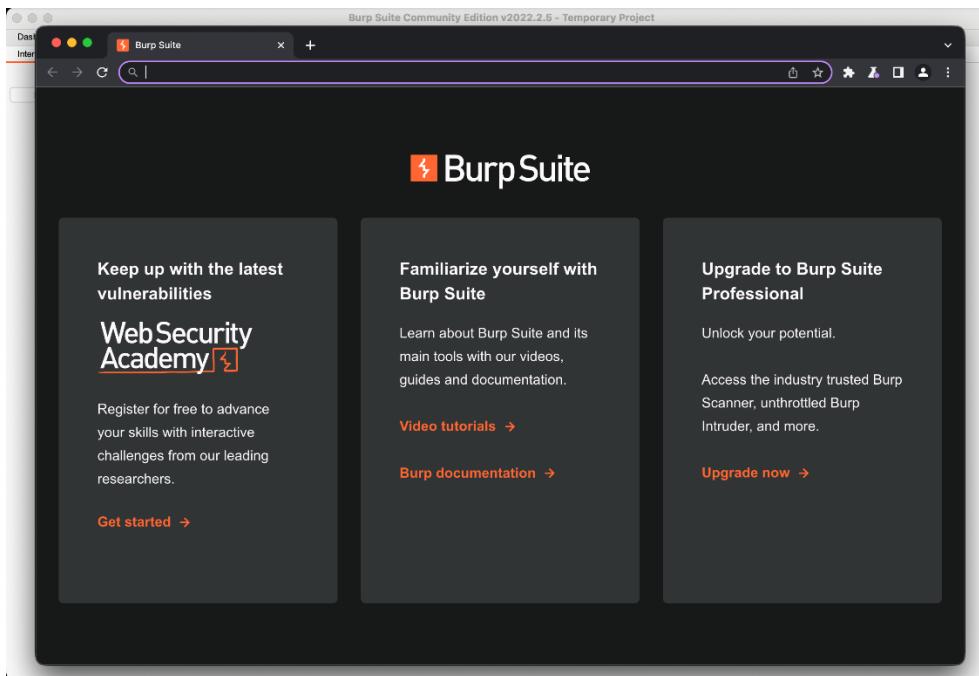
Pantalla para abrir el navegador de pruebas



Nota. En este apartado de proxy abrí el navegador de pruebas.

Figura 17

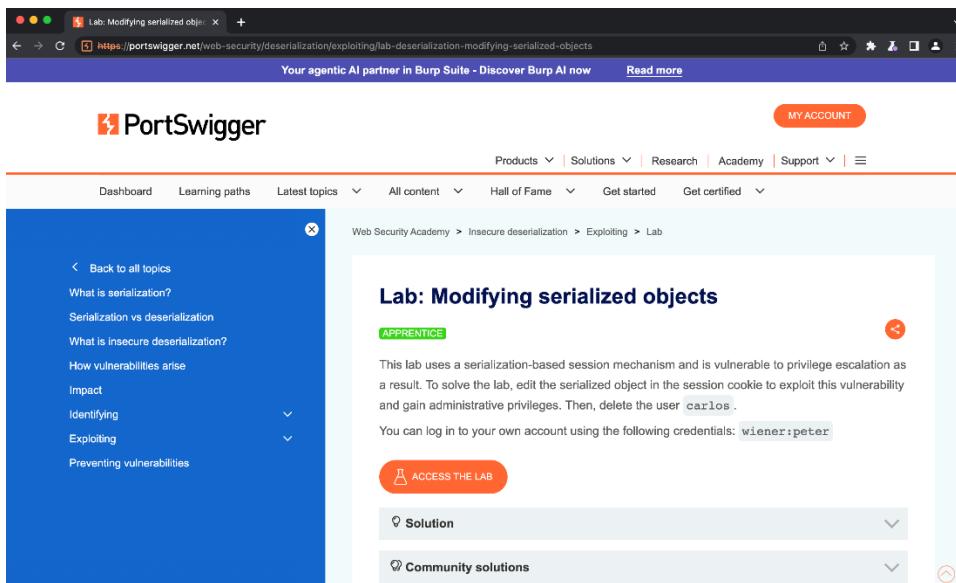
Navegador de Burp Suite



Nota. Es el navegador propio de la herramienta.

Figura 18

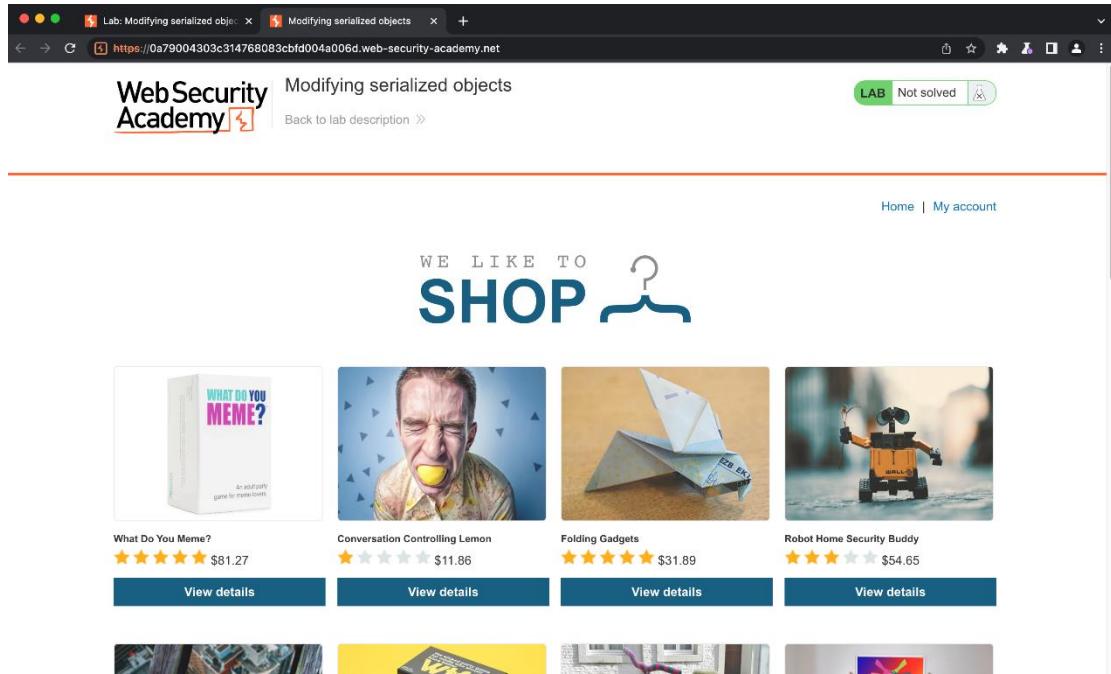
Página del laboratorio



Nota. Entre a la liga del laboratorio e inicie sesión con mi cuenta de la aplicación.

Figura 19

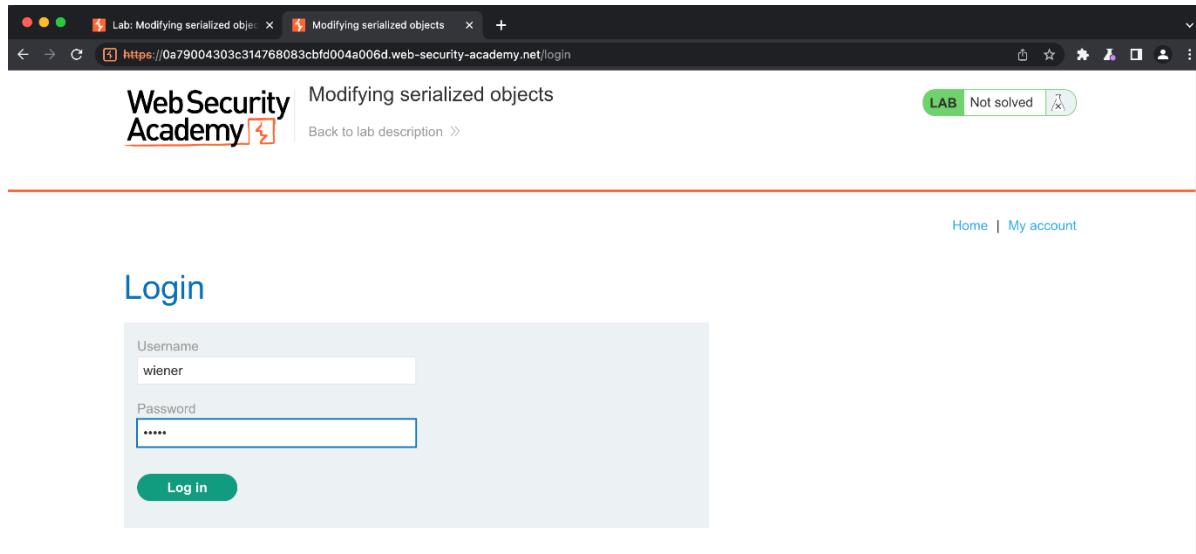
Entrar a la practica del laboratorio



Nota. Esta es la ventana principal de la practica.

Figura 20

Login con las credenciales



Nota. Use el login con las credenciales que nos da la actividad.

Figura 21*Pantalla de inicio*

Chromium Archivo Editar Ver Historial Favoritos Perfiles Pestaña Ventana Ayuda

Lab: Modifying serialized objects Modifying serialized objects

https://0a79004303c314768083cbfd004a006d.web-security-academy.net/my-account?id=wiener

Web Security Academy

Modifying serialized objects

Back to lab description

LAB Not solved

Home | My account | Log out

My Account

Your username is: wiener

Email

Update email

Nota. Nos manda a esta ventana y tenemos que pasarnos a nuestra herramienta.

Figura 22*Interceptamos la petición*

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
1	http://clients2.google.com	GET	/time/1/current?cup2key=5jBBlMFvK...	✓	200	1103	JSON						192.178.52.238
2	https://portswigger.net	GET	/web-security/deserialization/exploiting...		200	37480	HTML						✓ 3.174.207.62
6	https://portswigger.net	GET	/bundles/static-content/public/scripts/...	✓	200	26712	script	js					✓ 3.174.207.62
7	https://portswigger.net	GET	/content/images/logos/portswigger-logo...		200	5483	XML	svg					✓ 3.174.207.62 AW!
8	https://portswigger.net	GET	/content/images/logos/burp-suite-icon...		200	2608	XML	svg					✓ 3.174.207.62 AW!
12	https://portswigger.net	GET	/content/images/svg/icons/community...		200	2740	XML	svg					✓ 3.174.207.62 AW!
13	https://portswigger.net	GET	/content/images/svg/icons/professional...		200	2644	XML	svg					✓ 3.174.207.62 AW!
14	https://portswigger.net	GET	/content/images/svg/icons/enterprise.s...		200	2740	XML	svg					✓ 3.174.207.62 AW!
15	https://portswigger.net	GET	/bundles/static-content/public/scripts/...	✓	200	4272	script	js					✓ 3.174.207.62
16	https://portswigger.net	POST	/api/widgets?	✓	200	12817	JSON						✓ 3.174.207.62 Ses
17	https://pa.containers.piwik.pro	GET	/287552c2-4917-42e0-8982-ba894a2a7...		200	270484	script	js					✓ 20.79.214.157
19	https://pa.containers.piwik.pro	GET	/ppms.js		200	67525	script	js					✓ 20.79.214.157
20	https://www.googletagmanager....	GET	/gtm/g?&id=AW-11422135271	✓	200	384745	script						✓ 192.178.52.200
21	https://www.youtube.com	GET	/iframe_.api		200	4363	script						✓ 192.178.56.78 YSC

Nota. Buscamos la petición de login por el método de post.

Figura 23

Mandar al decoder

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'HTTP history' tab is active, displaying a list of requests. A specific POST request to '/login' is highlighted in orange and selected. The 'Request' tab shows the raw HTTP message:

```

1 POST /login HTTP/2
Host: 0a79004303c314768083cbfd004a006d.web-security-academy.net
Cookie: session=Tzo0OijVc2VyljoyOntz0jg6InVzZXJuYW1lIjt01Ghpibl7YjowO3d
Content-Length: 30
Cache-Control: max-age=0
Sec-Ch-Ua: "(Not(A BRAND);v="8", "Chromium";v="100"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://0a79004303c314768083cbfd004a006d.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a79004303c314768083cbfd004a006d.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: es-419,es;q=0.9

```

The 'Response' tab shows the raw HTTP response message:

```

1 HTTP/2 302 Found
Location: /my-account?id=wiener
Set-Cookie: session=Tzo0OijVc2VyljoyOntz0jg6InVzZXJuYW1lIjt01Ghpibl7YjowO3d; Secure; HttpOnly; S
X-Frame-Options: SAMEORIGIN
Content-Length: 0

```

The 'Inspector' panel on the right is open, showing the selected cookie 'session' with its value: Tzo0OijVc2VyljoyOntz0jg6InVzZXJuYW1lIjt01Ghpibl7YjowO3d. The 'Send to Decoder' option is highlighted.

Nota. Encontramos la petición, seleccionamos la cookie que es lo rojo y lo mandamos a decoder.

Figura 24

Pasamos a URL

The screenshot shows the 'Decoder' tool in Burp Suite. The top bar shows 'Decoder' is selected. The main area displays the cookie value: Tzo0OijVc2VyljoyOntz0jg6InVzZXJuYW1lIjt01Ghpibl7YjowO3d. Below this, two separate decoding panels are shown for the 'Text' and 'Hex' representations of the same value.

Nota. Decodreamos a url..

Figura 25

Decodeamos a base 64 y cambiamos nuestros permisos

Nota. Decodeamos a base 64 y cambiamos permisos de 0 a 1 que son permisos de administrador.

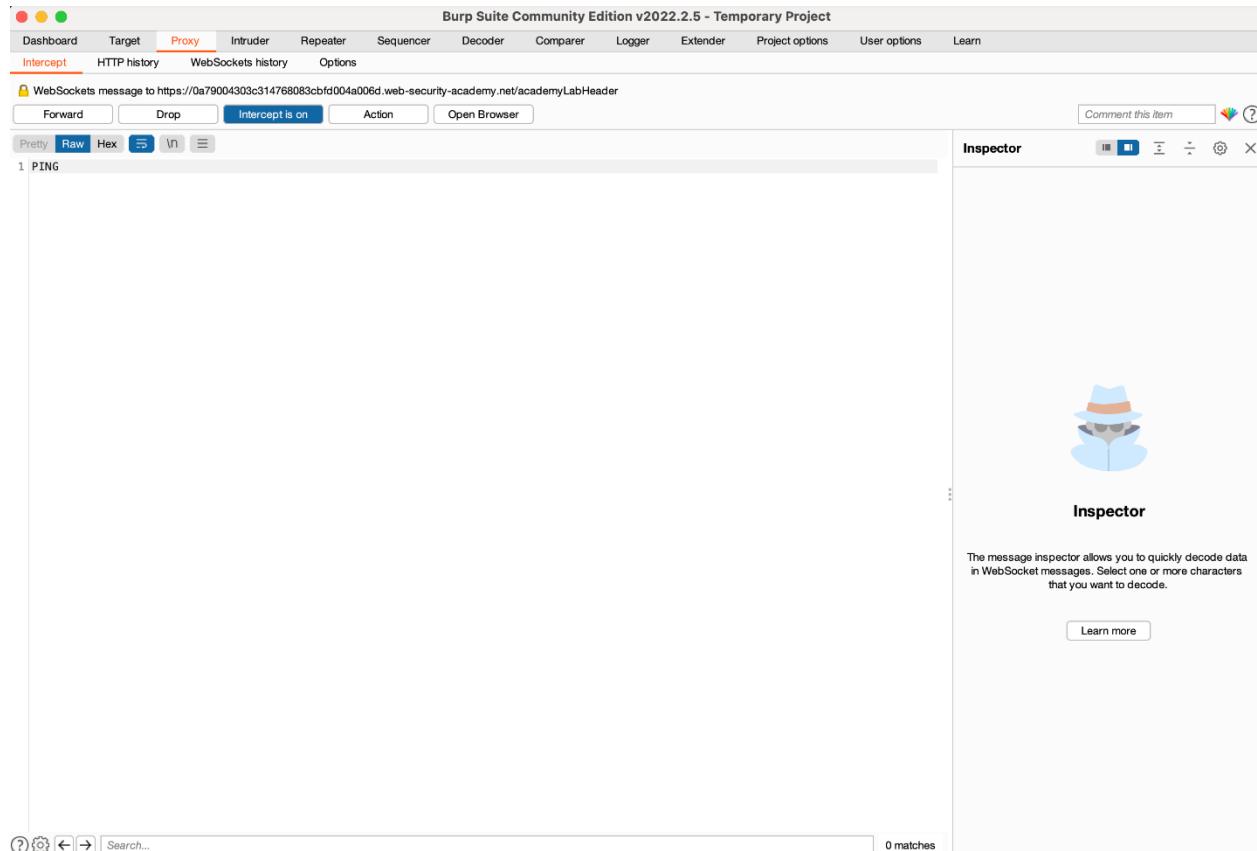
Figura 26

Inversa a nuestra respuesta

Nota. Volvemos a construir nuestra respuesta pasándola a base 64 después url.

Figura 27

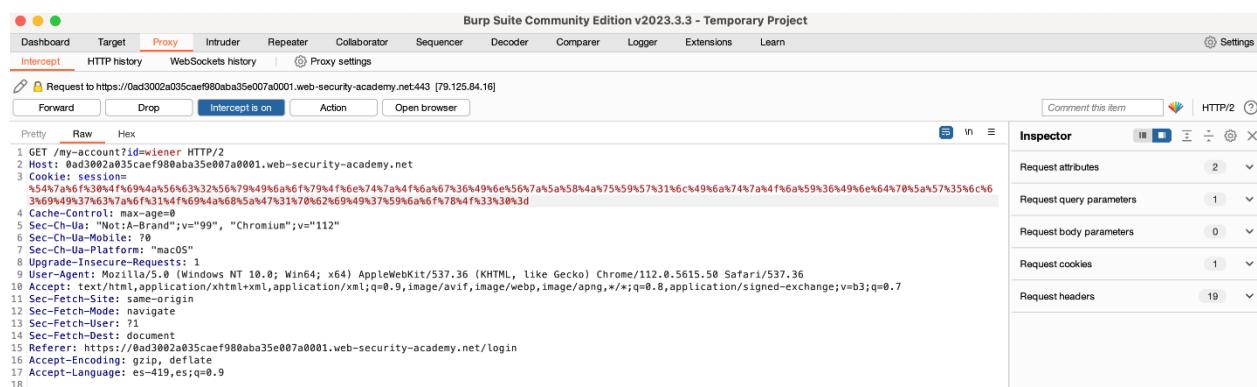
Interceptamos la respuesta



Nota. Nos regresamos a proxy, interceptamos la respuesta recargando el navegador.

Figura 28

Cambiamos la cooki



Nota. Cambiamos la cooki de la sesión por la que construimos nosotros así podremos avanzar.

Figura 29

Cambiamos la cooki

```

1 GET /academy/labheader HTTP/2
2 Host: 0ad3002a035cae980aba35e007a0001.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.5615.50 Safari/537.36
7 Upgrade: websocket
8 Origin: https://0ad3002a035cae980aba35e007a0001.web-security-academy.net
9 Sec-WebSocket-Version: 13
10 Accept-Encoding: gzip, deflate
11 Accept-Language: es-419,es;q=0.9
12 Cookie: session=$547a6f4304df694a456b632%56%79%49%6a%6f%79%47%6e%74%7a%4f%6a%67%36%49%6e%56%7a%5a%58%49%75%59%57%31%6c%49%6a%74%7a%4f%6a%59%36%49%6e%64%70%5a%57%35%6c%36%69%49%37%63%7a%6f%31%4f%66%9a%68%5a%4%73%70%62%69%49%37%59%6a%6f%78%4f%33%30%3d|
13 Sec-WebSocket-Key: YSHYRlpjURImZ2U0P9R6w==
14 
15

```

Nota. Esta intercepción es lo mismo por ejemplo cuando vemos la lista de usuarios y eliminamos a Carlos.

Figura 30

Cambiamos la cooki

WebSecurity Academy Modifying serialized objects

Modifying serialized objects

Back to lab description >

Home | Admin panel | My account

Users

wiener - [Delete](#)

carlos - [Delete](#)

Nota. En esta parte es donde vemos los usuarios y eliminamos a Carlos claro con nuestra cooki cambiada.

Figura 31

Usuario eliminado y laboratorio completo

The screenshot shows a browser window for the 'Web Security Academy' lab titled 'Modifying serialized objects'. The status bar indicates 'LAB Solved'. Below it, a message says 'Congratulations, you solved the lab!'. A link 'Share your skills!' with social media icons is present. The main content area shows a success message: 'User deleted successfully!'. The page title is 'Users' and there is a link 'wiener - Delete'.

Nota. Una vez eliminado se completa el laboratorio y cambia el status a solvet ya que se cumple.

Figura 32

Pantalla completa para que se aprecie que si es mi PC

The screenshot shows a full desktop environment. On the left is a code editor (VS Code) with a sidebar showing project files like 'gestor-libros', 'app', 'Http / Controllers', 'Models', 'Providers', 'bootstrap', 'cache', 'config', and 'OUTLINE'. In the center is a browser window for the 'Web Security Academy' lab titled 'Modifying serialized objects', showing the 'Solved' status and the 'Congratulations, you solved the lab!' message. The browser's address bar shows the URL 'https://0ad3002a035caeef980aba35e007a0001.web-security-academy.net/admin'. The desktop background is dark, and the taskbar at the bottom shows various application icons.

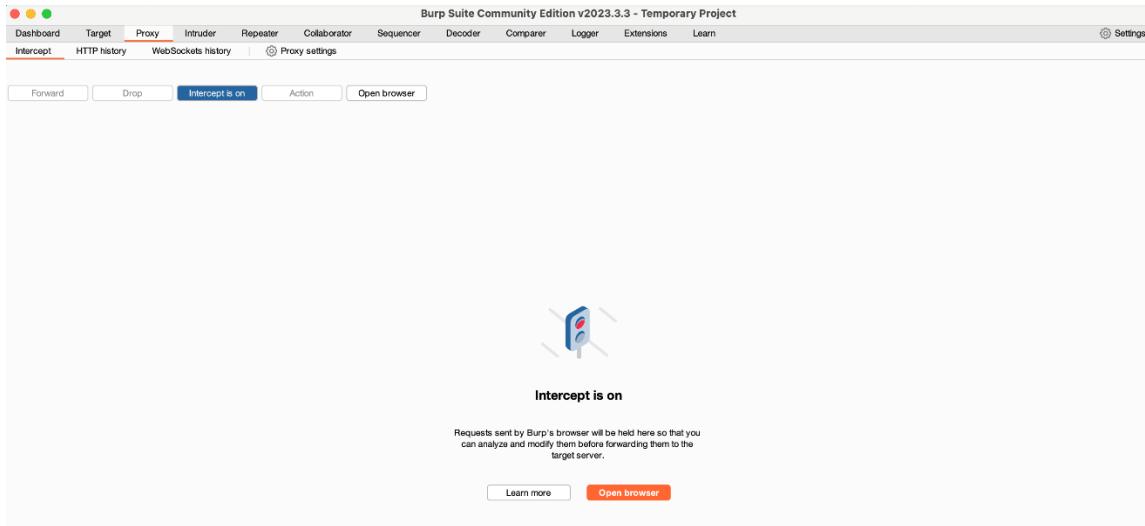
Nota. Una captura completa para que se aprecie que si se resolvió el laboratorio.

Desarrollo III

Ataque al sitio

Figura 33

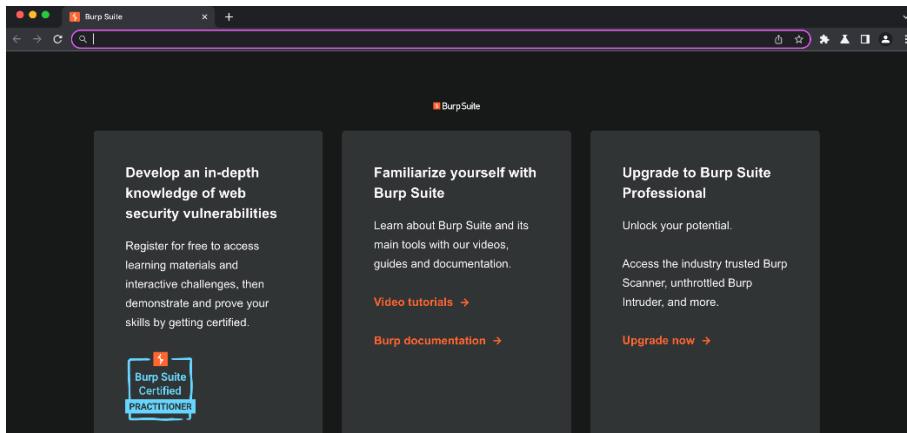
Pantalla de inicio



Nota. Aquí se ve como abrimos el programa y lo ponemos a escuchar.

Figura 34

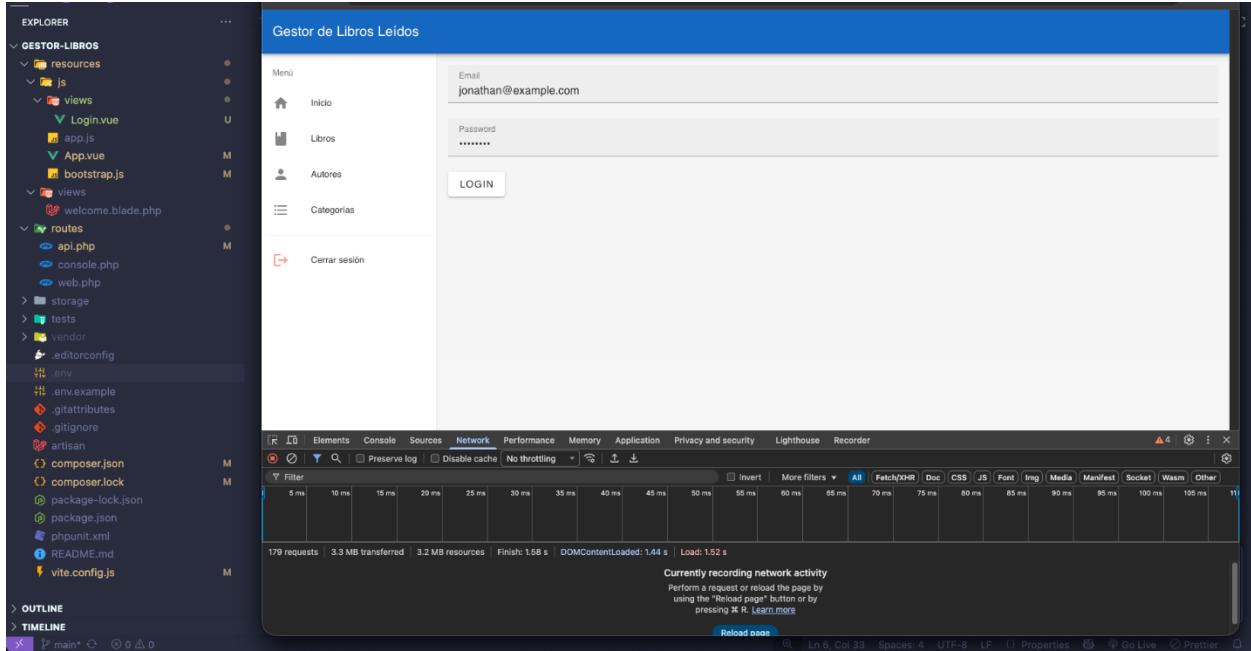
Pantalla del navegador



Nota. Originalmente usaría el buscador de la aplicación, pero me dio problemas, termine usando chrome pero lo levante en el puerto que ocupa el programa para escuchar los eventos y ver todo el tráfico.

Figura 35

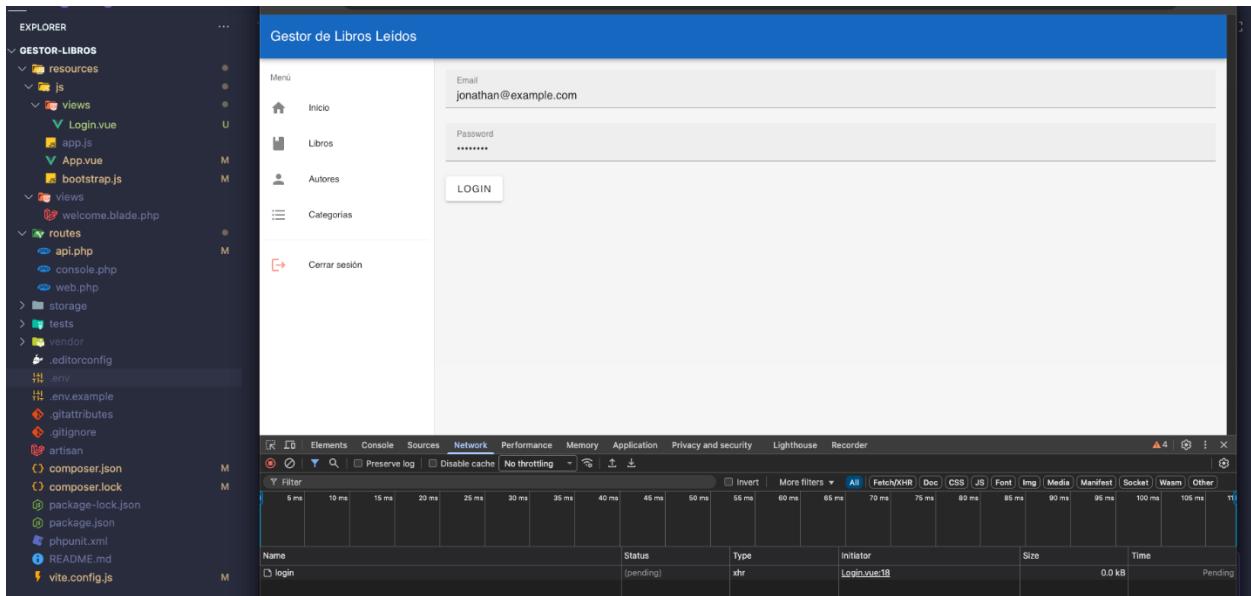
Inicio de la aplicación donde ponemos las credenciales



Nota. Es el login que ya conocemos de la actividad 1, ponemos las credenciales de la actividad 1.

Figura 36

Se ve que la petición se espera porque estamos interviniendo el tráfico



Nota. La petición se espera ya que estamos deteniendo el tráfico, volvemos a la aplicación.

Figura 37

petición del login

```

POST /api/login HTTP/1.1
Host: [REDACTED] 8000
Content-Length: 54
X-XSRF-TOKEN: eyJpdiI6ImZVRnVOTlooxSWvMEIwRD6cHB3YVE9PSIsInzbhHVljoir210d01BV2hBc0NIb2R4Y25EcncITzgzzjd5bmIxSJBlR1V0T2j6cnZyeDzxUe1raWNQ04vZC9FUytzTevrZUNUN2drclJRN3dLdT
EwV0VtCERczXcwVhuUzhdjJNnsBshy2dBeGVGdlnhSTzpxWlqVhzeXydeHQUjgjlCjYMM10iJMDU4NTyWjE3MThlnMyYy2R1YmUzYTvm0TAyZDVkNjNNDzjY2fIn2uWntu4Nzk3NT14N2uWnmfk
YjczZWYxliwidGfnjoiN03d; laravel_session=eyJpdiI61RLQ3pKam04We01QXNgV3RKTnA5vnc9PSisInzbhHVljoizBcmF20WVs0E96WlNNMxpWcfBqaUjU109hStGymp3Z0dHzEpjaGrZGpYQWVm1hus2JrNEN4Qzhxc2JUngV15Ub8xF1xdULxU
FZL2wycWY1tm10TNxQmxR1gxODQ0cKZNnRmE12clobwdvOGVjS2xLY2hWmhJ4V0w1lcjtYMM10iJjMGM5ZjN1zmEyotUwZmVHTf1ZTdjMTNm0DJlyj13MD00zmu1M)MwYtU00w0yMtQ5YzBkZTE3mI2
MgQ0xje51wiidGfnjoiN03d
Connection: close
{
    "email": "jonathan@example.com",
    "password": "password"
}

```

Nota. Se ve la petición del login con la información de las credenciales proporcionadas.

Figura 38

Modificación de las credenciales

```

POST /api/login HTTP/1.1
Host: [REDACTED] 8000
Content-Length: 54
X-XSRF-TOKEN: eyJpdiI6ImZVRnVOTlooxSWvMEIwRD6cHB3YVE9PSIsInzbhHVljoir210d01BV2hBc0NIb2R4Y25EcncITzgzzjd5bmIxSJBlR1V0T2j6cnZyeDzxUe1raWNQ04vZC9FUytzTevrZUNUN2drclJRN3dLdT
EwV0VtCERczXcwVhuUzhdjJNnsBshy2dBeGVGdlnhSTzpxWlqVhzeXydeHQUjgjlCjYMM10iJMDU4NTyWjE3MThlnMyYy2R1YmUzYTvm0TAyZDVkNjNNDzjY2fIn2uWntu4Nzk3NT14N2uWnmfk
YjczZWYxliwidGfnjoiN03d; laravel_session=eyJpdiI61RLQ3pKam04We01QXNgV3RKTnA5vnc9PSisInzbhHVljoizBcmF20WVs0E96WlNNMxpWcfBqaUjU109hStGymp3Z0dHzEpjaGrZGpYQWVm1hus2JrNEN4Qzhxc2JUngV15Ub8xF1xdULxU
FZL2wycWY1tm10TNxQmxR1gxODQ0cKZNnRmE12clobwdvOGVjS2xLY2hWmhJ4V0w1lcjtYMM10iJjMGM5ZjN1zmEyotUwZmVHTf1ZTdjMTNm0DJlyj13MD00zmu1M)MwYtU00w0yMtQ5YzBkZTE3mI2
MgQ0xje51wiidGfnjoiN03d
Connection: close
{
    "email": "demodemo1@example.com",
    "password": "123456"
}

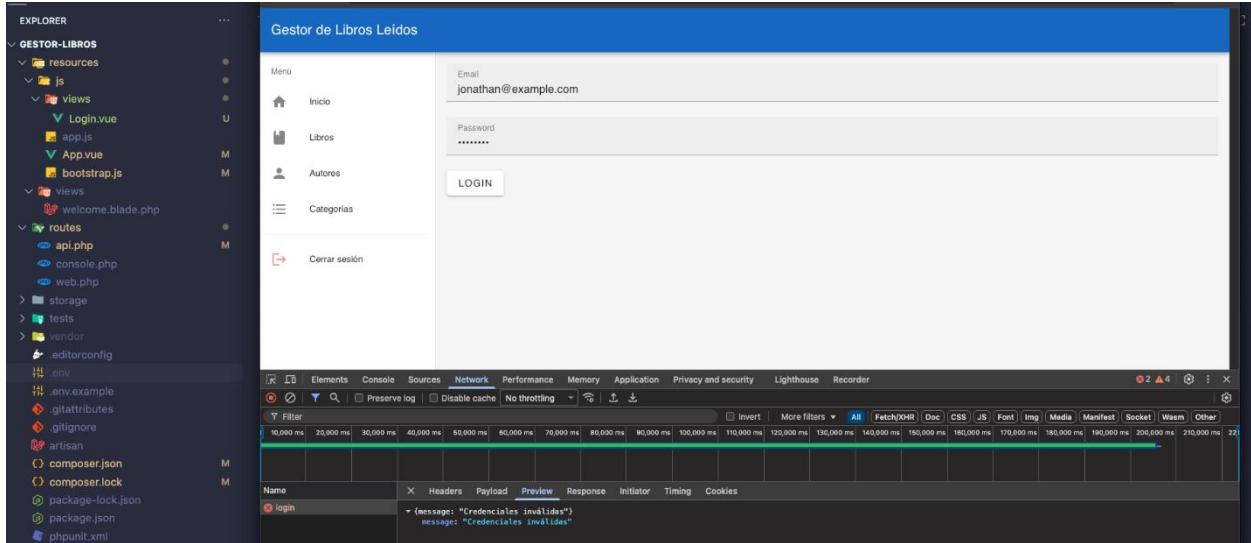
```

Nota. Modificamos las credenciales tanto correo como contraseña para enviarlas, claro ambas

incorrectas así que se espera una respuesta a estas.

Figura 39

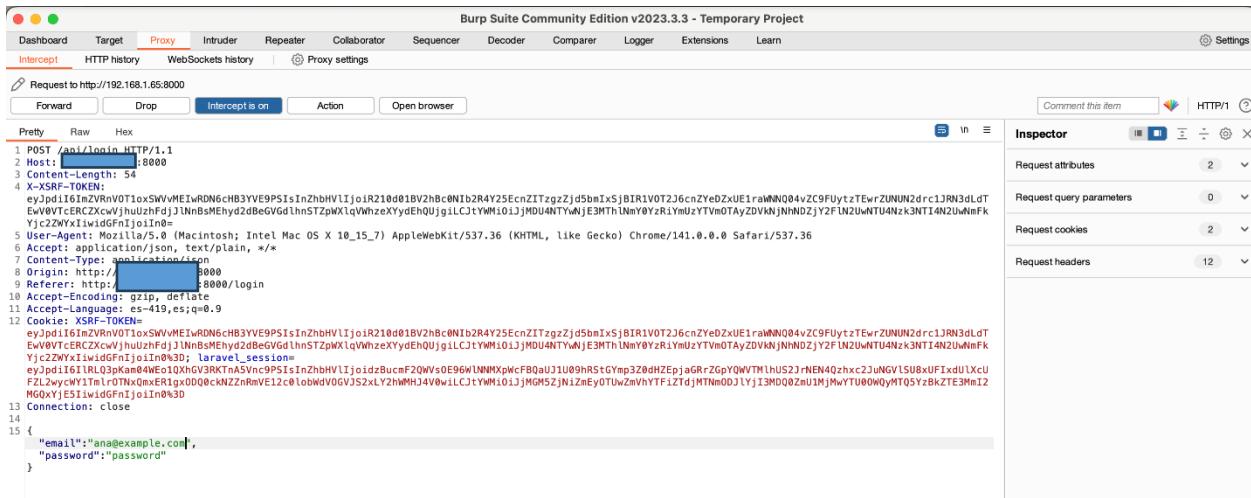
Respuesta de credenciales incorrectas



Nota. Se esperaba esta respuesta ya que las credenciales son incorrectas.

Figura 40

Otras credenciales correctas



Nota. Enviamos otras credenciales, que si bien no es la misma esta si se encuentra en la base de

datos así que nos tiene que dar acceso.

Figura 41

Respuesta de acceso

The screenshot shows a browser developer tools interface with the Network tab selected. The request 'login' is highlighted, showing a successful response. The response payload contains user information and a token:

```

{
  "user": {
    "id": 2,
    "name": "Ana Torres",
    "email": "ana@example.com",
    "email_verified_at": null,
    ...
  },
  "token": "14/fpxLSSr8eOCRakJxExZAK0PH10gICeacsawBpgb6d6o433"
}
  
```

Nota. La respuesta es exitosa ya que cambiamos las credenciales, pero estas si están registradas.

Figura 42

Desglose de la respuesta y redirección

The screenshot shows a browser developer tools interface with the Network tab selected. A response from 'Book.vue' is shown, which includes a 'Location' header indicating a redirect:

```

HTTP/2 302
Location: /libros
Content-Type: application/json; charset=UTF-8
Content-Length: 11
  
```

The response payload contains user information and a token:

```

{
  "user": {
    "id": 2,
    "name": "Ana Torres",
    "email": "ana@example.com",
    "email_verified_at": null,
    ...
  },
  "token": "14/fpxLSSr8eOCRakJxExZAK0PH10gICeacsawBpgb6d6o433"
}
  
```

Nota. Se ve el desglose en el retorno de todo el usuario que, aunque no fue el que se envió y lo cambiamos este también es correcto y se aprovechó para enviar la redirección desde el programa.

Conclusión

Estas tres actividades son de las que me han parecido más interesantes de hacer casi casi de la carrera, aunque fueron cosas digamos que se pueden ver como sencillas se tenia que tener mas conocimientos previos, o mínimo entender que se estaba haciendo en todo momento, más yo que corrí en local las cosas y tenía que entender el funcionamiento de las herramientas, los puertos y todo lo que se ocupaba para lograr el objetivo de las actividades, de verdad fue entretenido en un buen sentido, se siente raro como si uno entendiera mas del como funcionan las cosas y hasta cierto punto se pueden romper para que actúen como tú quieras, como decía un maestro de otra carrera para romper las reglas primero tienes que conocerlas, ese fue mi pensamiento a lo largo de las actividades, el entender el que, porque y como se hacían las cosas, me da pie a seguir aprendiendo mas cosas vinculadas a estos temas y también a repasar fundamentos que me di cuenta que tenía un tanto olvidados.

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.