

UNIVERSIDAD PRIVADA DE TACNA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS



Informe de Laboratorio

Examen de Unidad I

Que se presenta para el curso:

“Auditoría de sistemas”

Integrante(s):

- Cano Sucso Anthony Alexander

2020067573

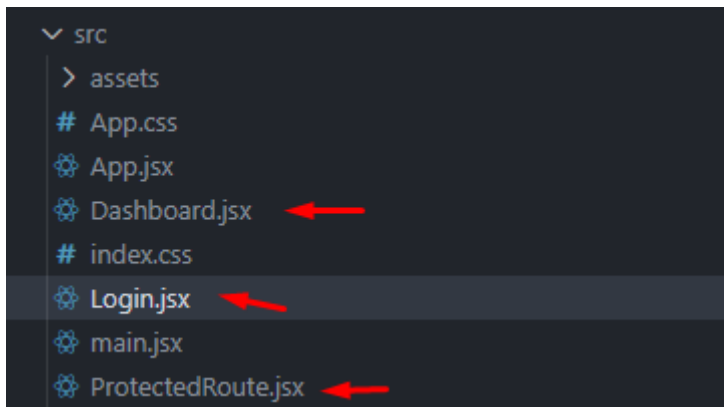
Docente:

Dr. Oscar Juan Jimenez Flores

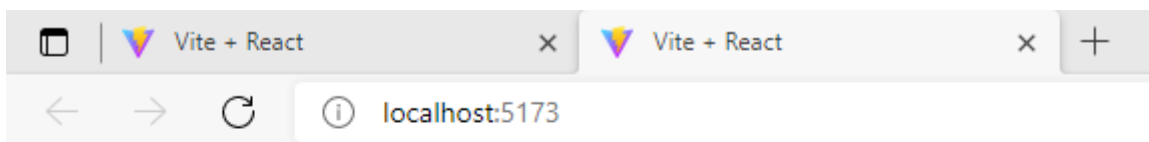
TACNA – PERÚ
2025

Link GitHub: https://github.com/anthonycs4/ExaUI_Auditoria.git

1.- Primero creamos el login en el front



```
src > Login.jsx > ...
1  import { useState } from 'react';
2  import { useNavigate } from 'react-router-dom';
3
4  const Login = () => {
5    const [user, setUser] = useState('');
6    const [pass, setPass] = useState('');
7    const [error, setError] = useState('');
8    const navigate = useNavigate();
9
10   const handleLogin = (e) => {
11     e.preventDefault();
12
13     // Usuarios ficticios
14     const validUsers = {
15       admin: '1234',
16       auditor: 'password',
17     };
18
19     if (validUsers[user] && validUsers[user] === pass) {
20       localStorage.setItem('auth', 'true');
21       navigate('/dashboard');
22     } else {
23       setError('Credenciales incorrectas');
24     }
25   };
26
27   return (
28     <div className="login">
29       <h2>Iniciar Sesión</h2>
30       <form onSubmit={handleLogin}>
31         <input type="text" placeholder="Usuario" value={user} onChange={(e) => setUser(e.target.value)} />
32         <input type="password" placeholder="Contraseña" value={pass} onChange={(e) => setPass(e.target.value)} />
33         <button type="submit">Entrar</button>
34         {error && <p style={{color: 'red'}}>{error}</p>}
35       </form>
36     </div>
37   );
38 };
39
40 export default Login;
41
```



Iniciar Sesión

<input type="text" value="Usuario"/>	<input type="password" value="Contraseña"/>	<input type="button" value="Entrar"/>
--------------------------------------	---	---------------------------------------

Usuario: admin
Contraseña:1234

Link GitHub: https://github.com/anthonycs4/ExaUI_Auditoria.git

2.-Conectamos y verificamos la conexión entre front y back

```
101   const handleOk = () => {  
102     setIsLoading(true); // Activar el estado de carga al inicio  
103  
104     axios.post('http://localhost:5500/analizar-riesgos', { activo: newData.activo })  
105       .then(response => {  
106         const { activo, riesgos, impactos } = response.data;  
107         addNewRows(activo, riesgos, impactos);  
108         setIsModalVisible(false); // Cerrar el modal
```

3.- Modificamos los cors en el back

```
app.py > ...
1  from openai import OpenAI
2  from flask import Flask, send_from_directory, request, jsonify, Response
3  import re
4  import requests
5  from flask_cors import CORS ←
```

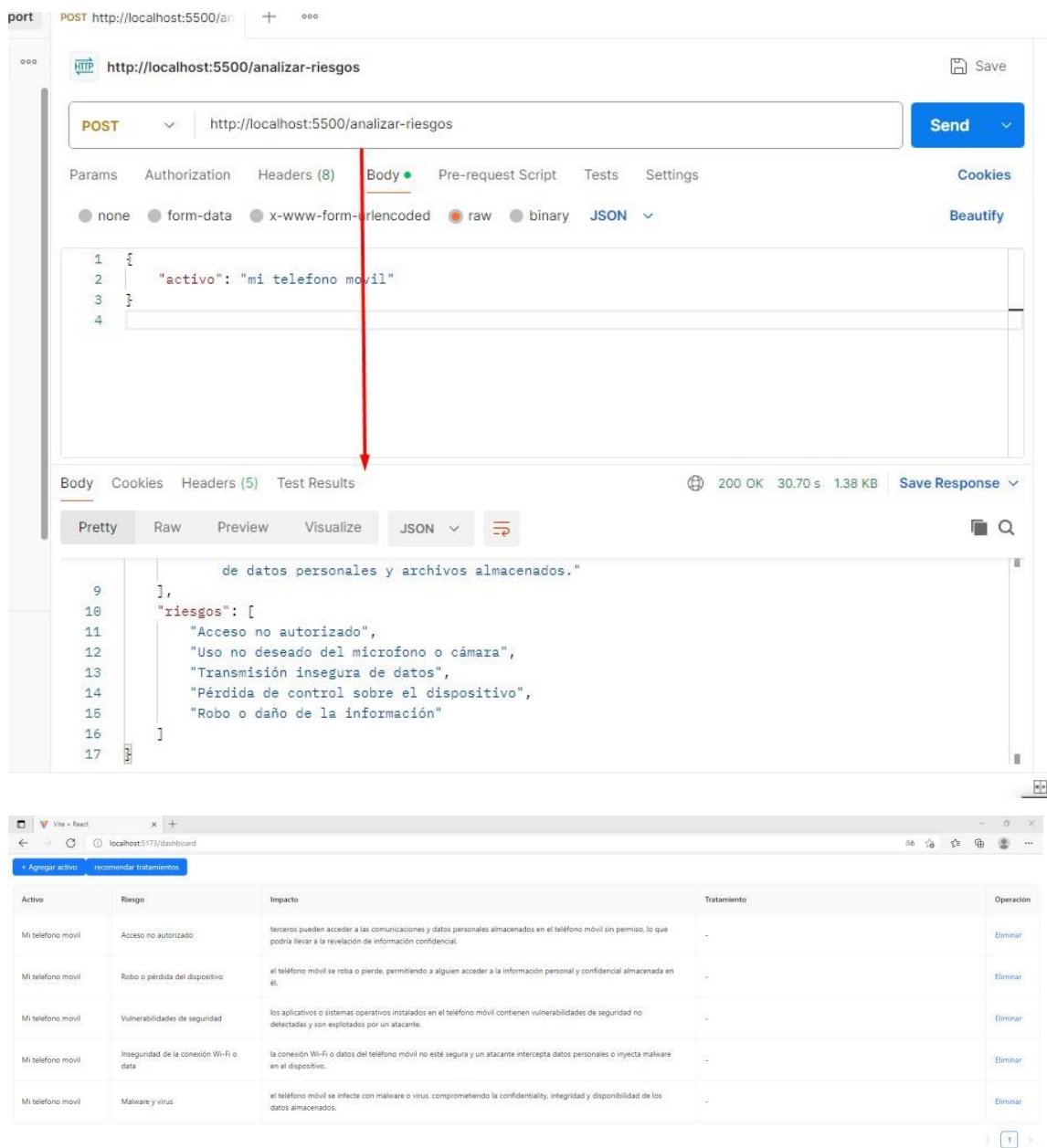
4.-Cambiamos el modelo a usar

```
def obtener_tratamiento( riesgo ):
    response = client.chat.completions.create(
        model="llama3",
        messages=[
            {"role": "system", "content": "Responde en español, eres una herramienta para gestion de riesgos de la i"},
            {"role": "user", "content": "mi telefono movil;Acceso no autorizado;un atacante puede acceder a la infor"},
            {"role": "assistant", "content": "Establecer un bloqueo de la pantalla de inicio que requiera autentica"},
            {"role": "user", "content": riesgo }
        ]
    )
    answer = response.choices[0].message.content
    return answer

def obtener_riesgos(activo):
    print(f"🔌 Llamando a Ollama para obtener riesgos de: {activo}")

    # Armar el cuerpo de la solicitud para Ollama
    data = {
        "model": "llama3", # Cambia este si tienes otro modelo cargado (ej. llama2, mistral, etc.)
        "messages": [
```

5.- Y ejecutamos tanto pruebas en postman como en el front



6.- y culminamos con los 5 activos

- API Transacciones:

Activo	Riesgo	Impacto	Tratamiento	Operación
API Transacciones	Inyección de SQL	un atacante inyecta código SQL malintencionado a través de la API para acceder o modificar datos en la base de datos, lo que podría llevar a una pérdida de confianza en la integridad de los datos.	-	Eliminar
API Transacciones	Ataques DDoS	la API es objeto de ataques Distribuidos (DDoS) lanzados por un atacante para sobrecargar el sistema y causar una interrupción temporal o permanente en la disponibilidad del servicio.	-	Eliminar
API Transacciones	Exfiltración de datos	los datos transmitidos a través de la API se exfilieron ilícitamente, lo que podría llevar a la revelación de información confidencial o sensible.	-	Eliminar
API Transacciones	Autenticación y autorización no segura	el sistema de autenticación y autorización employed by the API es débil o inseguro, permitiendo a un atacante acceder o modificar datos sin permiso.	-	Eliminar
API Transacciones	Vulnerabilidades en la implementación	la implementación de la API contiene vulnerabilidades de seguridad no detectadas y son explotadas por un atacante para acceder, leer, escribir o suprimir datos.	-	Eliminar

- Aplicación Web de Banca

Activo	Riesgo	Impacto	Tratamiento	Operación
Aplicación Web de Banca	Acceso no autorizado	terceros pueden acceder a las cuentas bancadas o información financiera comprometiendo la integridad y confidencialidad de la información.	-	Eliminar
Aplicación Web de Banca	Fraude	el sistema de banca se utiliza para cometer fraudes, como transferencias o consultas fraudulentamente, lo que puede dar lugar a pérdidas económicas importantes.	-	Eliminar
Aplicación Web de Banca	Vulnerabilidades de seguridad	la aplicación web tiene vulnerabilidades de seguridad no detectadas y son explotadas por un atacante, permitiéndoles obtener acceso a información financiera confidencial o inyectar malware en el sistema.	-	Eliminar
Aplicación Web de Banca	Inseguridad del servidor	el servidor que aloja la aplicación web no está segura y es objeto de ataques malintencionados, lo que puede llevar a la compromisión de la información bancaria o la inoperatividad del sistema.	-	Eliminar
Aplicación Web de Banca	Ataques DDoS	el sitio web de banca es víctima de ataques DDoS (ataque por dedicación de recursos), lo que puede hacer imposible acceso a la aplicación para los clientes, generando pérdidas y afectaciones a la reputación.	-	Eliminar

- Servidor de base de datos

Activo	Riesgo	Impacto	Tratamiento	Operación
Servidor de base de datos	Pérdida o daño de datos	la base de datos es dañada o eliminada accidentalmente o intencionalmente, lo que puede llevar a la pérdida permanente de información valiosa.	-	Eliminar
Servidor de base de datos	Vulnerabilidades de seguridad	las vulnerabilidades de seguridad en el servidor de bases de datos no detectadas y son explotadas por un atacante, lo que permite acceder a la información sensible almacenada en él.	-	Eliminar
Servidor de base de datos	Acceso no autorizado	terceros pueden acceder a la base de datos sin permiso, lo que podría llevar a la revelación de información confidencial o la modificación de datos críticos.	-	Eliminar
Servidor de base de datos	Problemas de performance	el servidor de bases de datos experimenta problemas de rendimiento, como retrasos en la ejecución, error de respuesta o baja velocidad, lo que puede afectar negativamente los procesos empresariales relacionados con la base de datos.	-	Eliminar
Servidor de base de datos	Brecha de seguridad en la configuración	una configuración insegura del servidor de bases de datos permite a un atacante explotar vulnerabilidades y acceder a la información almacenada sin autorización.	-	Eliminar

- Servidor de correo

Activo	Riesgo	Impacto	Tratamiento	Operación
Servidor de correo	Inyección de malware	un atacante inyecta malware en el servidor de correo electrónico para robar contraseñas, acceder a información confidencial o crear tráfico malintencionado.	-	Eliminar
Servidor de correo	Acceso no autorizado	terceros acceden al servidor de correo electrónico sin permiso, lo que podría llevar a la revelación de datos confidenciales o la alteración de correos electrónicos.	-	Eliminar
Servidor de correo	Pérdida o daño de datos	los correos electrónicos y archivos almacenados en el servidor se pierden o dañan debido a un error en el sistema o un fallo en el hardware.	-	Eliminar
Servidor de correo	Exfiltraciones de información	el servidor de correo electrónico no está configurado adecuadamente para proteger la privacidad de los usuarios, lo que lleva a la filtración inesperada de correos electrónicos y datos personales.	-	Eliminar
Servidor de correo	Dificultades en la recuperación	no hay respaldos adecuados o no se han realizado copias de seguridad periódicas del servidor, lo que hace difícil o imposible la restauración en caso de un fallo.	-	Eliminar

- Autenticación MFA

Activo	Riesgo	Impacto	Tratamiento	Operación
Autenticación MFA	Incorreción en la implementación	la autenticación Multi-Factor (MFA) no se implementa correctamente, lo que permite el acceso no autorizado a recursos y sistemas.	-	Eliminar
Autenticación MFA	Vulnerabilidades de protocolos	los protocolos utilizados para la autenticación MFA contienen vulnerabilidades de seguridad no detectadas y son explotados por un atacante.	-	Eliminar
Autenticación MFA	Riesgo de phishing	los usuarios pueden ser víctimas de phishing y proporcionar sus credenciales, incluyendo información adicional requerida por el MFA, a un atacante malintencionado.	-	Eliminar
Autenticación MFA	Pérdida o daño de la contraseña	la contraseña utilizada para la autenticación MFA se pierde o se daña, lo que impide el acceso autorizado a recursos y sistemas.	-	Eliminar
Autenticación MFA	Incapacidad de respuesta en situaciones de emergencia	en caso de una emergencia, la falta de redundancia o la complejidad de la configuración del MFA puede llevar a una situación crítica en la que no se pueda acceder a los recursos críticos.	-	Eliminar