UNIVERSIDAD TECNOLÓGICA Y POLITÉCNICA DE COYUCA DE BENITEZ

ASIGNATURA: FRAMEWORKS PARA DESARROLLO WEB.

CUATRIMESTRE: V UNIDAD: III

TEMA: PROCESO DE SEGURIDAD E IMPLEMENTACION DE LA APLICACIÓN WEB

DOCENTE: ING. GEOVAMY PIZA RODRIGUEZ.

ALUMNA: ADRIANA DE LOS SANTOS LAYNA

CARRERA: INGENIERÍA EN TECNOLOGÍAS DE LA NFORMACIÓN E INNOVACIÓN DIGITAL. (LINEA)

FECHA: 20 ABRIL 2026.

Índice

Tema 1: Fundamentos de seguridad web e Integración de la aplicación web

Actividad 1

¿Qué es la Seguridad Web?

La seguridad web es el conjunto de medidas, prácticas, herramientas y protocolos diseñados para proteger sitios web, aplicaciones web y servicios en línea contra amenazas cibernéticas. Su objetivo es garantizar tres pilares fundamentales:

- **Confidencialidad:** que solo los usuarios autorizados puedan acceder a la información.
- **Integridad:** que los datos no sean alterados sin autorización.
- **Disponibilidad:** que los servicios estén accesibles cuando se necesiten.

En el contexto de nuestra aplicación de Gestión de Eventos Universitarios, la seguridad web implica proteger los datos de los usuarios registrados (estudiantes, profesores, administradores), las credenciales de acceso, la información de los eventos y las inscripciones.

Tipos de Amenazas en Seguridad Web

- **Inyección SQL:** El atacante introduce código SQL malicioso en formularios de entrada para acceder o manipular la base de datos MySQL.
- **Cross-Site Scripting (XSS):** Inyección de scripts JavaScript en páginas web que son ejecutados por otros usuarios. Puede robar cookies de sesión.
- **Cross-Site Request Forgery (CSRF):** Engaña al navegador del usuario para enviar solicitudes no autorizadas, como inscribir o eliminar eventos sin consentimiento.
- **Ataques de fuerza bruta:** Intentos repetitivos de adivinar contraseñas probando miles de combinaciones.
- **Man-in-the-Middle (MITM):** Interceptación de las comunicaciones entre el navegador y el servidor para capturar datos sensibles.
- **DDoS:** Saturación del servidor con tráfico masivo para dejarlo inaccesible.

Herramientas y Mecanismos de Control de Seguridad

Herramienta / Mecanismo	Descripción	Aplicación en el Proyecto
Helmet.js	Middleware de Express que configura headers HTTP de seguridad	Protege contra XSS, clickjacking y sniffing
CORS	Control de acceso de origen cruzado	Restringe qué dominios pueden acceder a la API
Rate Limiting	Limita solicitudes por IP en un período	Previene ataques de fuerza bruta en login
Input Validation	Validación y sanitización de datos de entrada	Previene inyección SQL y XSS
Prepared Statements	Consultas parametrizadas a la BD	Previene inyección SQL en consultas MySQL
CSRF Tokens	Tokens únicos por sesión	Protege acciones como inscripción y creación
WAF (Firewall)	Filtra tráfico malicioso	Protección a nivel de servidor/hosting

Tema 2: Mecanismos de encriptación y control de acceso

Actividad 2

Herramientas y Mecanismos de Control de Seguridad en la Aplicación

En la aplicación de Gestión de Eventos Universitarios se implementan los siguientes mecanismos:

- **bcrypt:** Librería para el hashing de contraseñas. Aplica un salt aleatorio y múltiples rondas de cifrado, haciendo computacionalmente costoso descifrar las contraseñas.
- **express-validator:** Middleware para validar y sanitizar datos de entrada en el servidor, previniendo datos malformados o inyecciones.
- **jsonwebtoken (JWT):** Genera tokens firmados digitalmente para manejar sesiones sin estado (stateless) entre el frontend y el backend.
- **express-rate-limit:** Middleware que limita intentos de login a 5 por cada 15 minutos por IP.

Tipos de Encriptación de Datos

Tipo	Algoritmo	Uso en el Proyecto	Características
Hash (unidireccional)	bcrypt	Almacenamiento de contraseñas	No se puede revertir. Salt aleatorio.
Simétrica	AES-256	Cifrado de datos sensibles en BD	Misma clave para cifrar y descifrar.
Asimétrica	RSA / ECDSA	Certificados SSL/TLS (HTTPS)	Par de claves pública/privada.
Firma digital	HMAC-SHA256	Firma de tokens JWT	Verifica integridad y autenticidad.

Métodos de Identificación y Autenticación

1. **Autenticación basada en contraseña:** El usuario proporciona email + contraseña. La contraseña se compara contra el hash almacenado en MySQL usando bcrypt.
2. **Autenticación basada en tokens (JWT):** El usuario se autentica con credenciales. El servidor genera un JWT firmado con la clave secreta. El cliente almacena el token y lo envía en cada solicitud (header Authorization: Bearer <token>). El servidor verifica la firma del token antes de procesar la solicitud.
3. **Autorización por roles (RBAC):** admin (crear, editar y eliminar eventos, gestionar usuarios); profesor (crear y editar sus propios eventos); estudiante (inscribirse y cancelar inscripciones).

Tema 3: Certificados de seguridad web

Actividad 3

¿Qué es un Certificado de Seguridad?

Un certificado de seguridad SSL/TLS es un archivo digital que:

- Vincula una clave criptográfica con la identidad de una organización o dominio.
- Habilita el protocolo HTTPS (HTTP sobre TLS).
- Cifra la comunicación entre el navegador del usuario y el servidor.
- Muestra el candado de seguridad en la barra de direcciones del navegador.

Tipos de Certificados SSL/TLS

Tipo	Validación	Costo	Uso Recomendado
DV (Domain Validation)	Solo verifica propiedad del dominio	Gratis (Let's Encrypt)	Blogs, proyectos académicos
OV (Organization)	Verifica dominio + organización	$50-$200 USD/año	Sitios corporativos
EV (Extended)	Verificación exhaustiva	$100-$500+ USD/año	Bancos, e-commerce
Wildcard	Cubre dominio + subdominios	$50-$300 USD/año	Múltiples subdominios
Multi-Domain (SAN)	Múltiples dominios diferentes	$100-$400 USD/año	Varios dominios

Certificado utilizado en el proyecto

Let's Encrypt - Certificado DV gratuito

- **Justificación:** Al ser un proyecto académico, un certificado DV es suficiente para cifrar las comunicaciones y proteger los datos.
- **Renovación:** Automática cada 90 días mediante Certbot o el panel de Hostinger.
- **Protocolo:** TLS 1.3 con cipher suites modernas.
- **Compatibilidad:** Compatible con todos los navegadores modernos.

Tema 4: Despliegue de aplicaciones en el servidor

Actividad 4

Servicios, Accesos y Permisos del Servidor en Producción

Servicio	Descripción	Puerto
Apache/Nginx	Servidor web (HTML, CSS, JS)	80 (HTTP) / 443 (HTTPS)
Node.js	Runtime del backend (Express API)	3000 (interno)
MySQL	Base de datos relacional	3306 (solo local)
SSH	Acceso remoto seguro al servidor	65002 (personalizado)
SFTP	Transferencia segura de archivos	65002 (sobre SSH)

Permisos del servidor

- **Archivos:** 644 (lectura para todos, escritura solo el propietario)
- **Directorios:** 755 (lectura y ejecución para todos, escritura solo el propietario)
- **Archivos .env:** 600 (solo lectura/escritura para el propietario)
- **MySQL:** Acceso solo desde localhost (127.0.0.1), sin acceso remoto

Proceso de Despliegue al Servidor de Producción

1. **Preparación del entorno local:** Verificar que todas las dependencias estén instaladas y la app funcione localmente. Configurar variables de entorno para producción.
2. **Configuración del servidor:** Acceder via SSH al servidor. Instalar Node.js, MySQL y configurar el virtual host en Apache/Nginx.
3. **Transferencia de archivos:** Subir los archivos del proyecto mediante SFTP (puerto 65002). Verificar permisos (644 para archivos, 755 para directorios).
4. **Configuración de la base de datos:** Crear la base de datos MySQL y el usuario. Importar el esquema y datos iniciales. Configurar las credenciales en el archivo .env.
5. **Instalación de SSL:** Configurar el certificado Let's Encrypt desde el panel de Hostinger. Verificar que HTTPS funcione correctamente.
6. **Verificación final:** Probar todas las funcionalidades en producción. Verificar login, registro, creación de eventos e inscripciones.

Justificación del protocolo SFTP

Se seleccionó SFTP (SSH File Transfer Protocol) operando sobre el puerto 65002 por las siguientes razones:

- Toda la comunicación se cifra mediante SSH.
- Las credenciales y archivos nunca viajan en texto plano.
- Se utiliza la misma autenticación que SSH.
- Opera sobre un único puerto (65002), simplificando reglas de firewall.
- Herramienta utilizada: FileZilla con protocolo SFTP.

Servicios de Autenticación y Autorización implementados

1. El usuario accede al formulario de registro/login.
2. En registro: los datos se validan, la contraseña se hashea con bcrypt y se almacena en MySQL.
3. En login: se verifica email + contraseña hasheada. Si es válido, se genera un JWT.
4. El JWT se almacena en el cliente (localStorage) y se envía en cada request.
5. El middleware verifica el JWT y extrae el rol del usuario.
6. El middleware de autorización verifica si el rol tiene permiso para la acción.
7. Se registra cada acceso en logs para auditoría.

Glosario

Términos clave de Seguridad Web e Implementación de Aplicaciones - Unidad III

Fundamentos de Seguridad Web

Seguridad Web

Conjunto de medidas, protocolos y herramientas diseñadas para proteger sitios y aplicaciones web contra amenazas, accesos no autorizados, robo de datos y ataques cibernéticos.

OWASP

Organización sin fines de lucro dedicada a mejorar la seguridad del software. Publica el OWASP Top 10, una lista de las 10 vulnerabilidades más críticas en aplicaciones web.

Vulnerabilidad

Debilidad o fallo en un sistema de software que puede ser explotada por un atacante para comprometer la seguridad, integridad o disponibilidad de la aplicación.

Inyección SQL

Ataque que consiste en insertar código SQL malicioso en campos de entrada de una aplicación web para manipular o acceder a la base de datos sin autorización.

XSS (Cross-Site Scripting)

Vulnerabilidad que permite a un atacante inyectar scripts maliciosos en páginas web vistas por otros usuarios, pudiendo robar cookies, sesiones o datos sensibles.

CSRF

Ataque que engaña al navegador del usuario para que envíe solicitudes no autorizadas a un sitio en el que está autenticado.

WAF

Sistema de seguridad que filtra y monitorea el tráfico HTTP entre una aplicación web e Internet, protegiendo contra ataques como XSS, SQL Injection y DDoS.

DDoS

Ataque distribuido de denegación de servicio que busca hacer que un servidor no esté disponible saturándolo con tráfico masivo.

Encriptación y Control de Acceso

Encriptación

Proceso de convertir datos legibles en un formato ilegible mediante un algoritmo y una clave, para proteger la información.

Encriptación Simétrica

Método de cifrado que usa la misma clave tanto para encriptar como para desencriptar. Ejemplo: AES.

Encriptación Asimétrica

Método que utiliza un par de claves: pública (para encriptar) y privada (para desencriptar). Ejemplo: RSA.

Hash

Función que convierte datos de cualquier tamaño en una cadena de longitud fija. Es unidireccional. Ejemplo: bcrypt, SHA-256.

Autenticación

Proceso de verificar la identidad de un usuario, dispositivo o sistema.

Autorización

Proceso de determinar qué recursos o acciones tiene permitido un usuario autenticado.

JWT

Estándar abierto (RFC 7519) para crear tokens de acceso que permiten la transmisión segura de información entre partes.

OAuth 2.0

Protocolo de autorización que permite a aplicaciones de terceros acceder a recursos sin compartir credenciales.

MFA

Método de autenticación que requiere dos o más factores de verificación independientes.

RBAC

Modelo de control de acceso basado en roles, donde los permisos se asignan a roles y los usuarios reciben los permisos del rol asignado.

Certificados de Seguridad Web

SSL

Protocolo criptográfico (ahora reemplazado por TLS) que establece un canal cifrado entre servidor y navegador.

TLS

Versión actualizada y más segura de SSL. TLS 1.3 es la versión más reciente.

HTTPS

Protocolo HTTP cifrado mediante TLS/SSL. Se identifica por el candado en el navegador.

Certificado SSL/TLS

Archivo digital que vincula una clave criptográfica con los datos de una organización. Tipos: DV, OV, EV.

Let's Encrypt

Autoridad de certificación gratuita y automatizada que emite certificados SSL/TLS DV.

CA (Certificate Authority)

Entidad de confianza encargada de emitir, renovar y revocar certificados digitales.

Despliegue de Aplicaciones

Deploy

Proceso de poner una aplicación web en un servidor de producción para que sea accesible a los usuarios finales.

Servidor de Producción

Servidor donde se ejecuta la versión final de una aplicación, accesible por usuarios reales.

CI/CD

Prácticas de desarrollo que automatizan la integración, pruebas y despliegue de código.

FTP / SFTP

FTP transfiere archivos entre cliente y servidor. SFTP es su versión segura cifrada mediante SSH.

SSH

Protocolo de red que proporciona un canal seguro y cifrado para acceder a servidores de forma remota.

DNS

Sistema de Nombres de Dominio que traduce nombres de dominio legibles a direcciones IP.

Hosting

Servicio que proporciona infraestructura de servidor para alojar sitios y aplicaciones web.

Docker

Plataforma de contenedores que empaqueta aplicaciones con todas sus dependencias.

Variables de Entorno

Variables del sistema operativo que almacenan configuraciones sensibles fuera del código fuente.

Bibliografía

Libros

Stallings, W. (2017). "Cryptography and Network Security: Principles and Practice". 7th Edition. Pearson.
Texto fundamental sobre criptografía, protocolos de seguridad de red, cifrado simétrico y asimétrico.

Stuttard, D. & Pinto, M. (2011). "The Web Application Hacker's Handbook". 2nd Edition. Wiley.
Guía completa sobre vulnerabilidades en aplicaciones web, incluyendo inyección SQL, XSS, CSRF.

Fielding, R. T. (2000). "Architectural Styles and the Design of Network-based Software Architectures". UC Irvine.
Tesis que define el estilo arquitectónico REST, base de las APIs web modernas.

Tanenbaum, A. S. & Wetherall, D. J. (2011). "Computer Networks". 5th Edition. Pearson.
Referencia sobre redes, protocolos HTTP, HTTPS y seguridad en la capa de transporte.

Flanagan, D. (2020). "JavaScript: The Definitive Guide". 7th Edition. O'Reilly Media.
Referencia completa del lenguaje JavaScript, desarrollo frontend y seguridad del lado del cliente.

Recursos en Línea

OWASP Foundation. OWASP Top Ten.
https://owasp.org/www-project-top-ten/

Mozilla Developer Network (MDN). Web Security.
https://developer.mozilla.org/en-US/docs/Web/Security

Let's Encrypt. Documentation.
https://letsencrypt.org/docs/

Node.js Documentation. Security Best Practices.
https://nodejs.org/en/docs/guides/security/

Express.js. Production Best Practices: Security.
https://expressjs.com/en/advanced/best-practice-security.html

MySQL. Security in MySQL.
https://dev.mysql.com/doc/refman/8.0/en/security.html

Estándares y RFCs

- RFC 7519 - JSON Web Token (JWT). IETF, 2015.
- RFC 8446 - The Transport Layer Security (TLS) Protocol Version 1.3. IETF, 2018.
- RFC 6749 - The OAuth 2.0 Authorization Framework. IETF, 2012.
- NIST SP 800-63B - Digital Identity Guidelines: Authentication and Lifecycle Management.