UNIVERSIDAD TÉCNICA NACIONAL

INGENIERÍA EN TECNOLOGÍAS DE INFORMACIÓN

SEDE DEL PACÍFICO

GUÍA DE ACTIVIDAD PRÁCTICA Y/O LABORATORIO

Curso:	ITI-523 – Tecnologías y Sistemas Web II	**Puntos**	100
Profesor:	Jorge Ruiz (york)	**Valor %**	10
Fecha Entrega:	08/Agosto/2025	**Avatar**	
Tiempo:	Trabajo en clases + 1 semana		

• Objetivos de la actividad.

- Que los estudiantes pongan en práctica los conceptos desarrollados en la clase relacionado con el tema de fundamentos de programación en JavaScript por medio del Framework React JS.

- Implementación de mecanismos de seguridad que protejan la ejecución, entorno y data administrada por el API-Rest.

- Aplicar pruebas o testing sobre el API-Rest, para validar el rendimiento, ejecución y los procesos de seguridad implementados.

• Instrucciones de la práctica.

Práctica 03

Observaciones:

- En el curso dispuesto en el Campus Virtual, existe un segmento denominado “Entorno de Trabajo”, se sugiere que lo revisen y sigan con detenimiento con el objetivo de que sus equipos, aplicaciones puedan trabajar adecuadamente con el repositorio de sub-versionamiento de GitHub.

- Lea cuidadosamente cada uno de los problemas planteados y en caso de duda, puede utilizar el chat dispuesto para tal fin, que se encuentra al inicio del curso en la plataforma Moodle del Campus Virtual.

- Recuerde que el trabajo debe ser cargado en GitHub.

Problemas:

1. Utilizando la estructura de la tabla de Categories de la base de datos NorthWind:

   Categories : {
   CategoryID : texto
   CategoryName : texto
   Description : texto
   Image : Blob
   Mime: texto
   }

   Adapte o cree una nueva API que trabaje con el motor de base de datos de MongoDB, y que esta sea capaz de almacenar la estructura anterior, en otras palabras, que almacene los textos y la data de la imagen ya procesada.

2. Además, esta nueva API, debe aplicar los mecanismos de control vistos en la clase anterior:
   a. Creación de Roles
      i. Root - Rol superior, soporte global.
      ii. Admin - Rol administrativo (acceso parcial actualizaciones/consultas)
      iii. User - Usuario normal (acceso parcial).
      iv. Guest - Invitado (solo ciertas consultas)

   b. Creación del usuario Root y por defecto pertenece al grupo Root, este usuario no se puede borrar, no se puede modificar, solo la contraseña. Este usuario es el único que puede crear usuarios, modificarlos y/o borrarlos

      Además, que mantenga el conjunto actual de controles:
      i. Cross Origin Resource Sharing
      ii. Helmet
      iii. Cookie Session
      iv. Request and Response Timeout
      v. CORS Headers
      vi. Keep alive

3. Además, debe agregar controles adicionales a su nueva API:
   a. Limitar la cantidad de solicitudes entrantes (rae limit).
   b. Limitar el tamaño del Payload o carga útil de solicitud y respuesta.
   c. Validar los datos de entrada (formato, longitud, otros)
   d. Content Security Policies (políticas de seguridad de contenido)

4. Aplique auditoría a su nueva API para identificar y reportar posibles vulnerabilidades de las librerías o dependencias que usa usted en su proyecto, para tal efecto y usando la consola en la raíz de su nueva API, ejecute el siguiente comando: **npm audit**, analice las implicaciones de las

vulnerabilidades encontradas y posteriormente defina pasos a seguir para corregirlas o en su defecto mitigarlas.

5. Usando su IA favorita desarrolle una serie de pruebas o mejor dicho ataques que pueda usted ejecutar sobre su nueva API, para poder validar que tan funcionales, prácticos u oportunos son los elementos mencionados en los puntos 2b y 3 de esta práctica. Documente este proceso:
    a. IA utilizada y en caso de que pueda elegir el LLM también indicar.
    b. Copiar y explicar el prompt ejecutado para lograr definir sus pruebas.
    c. Explicar la prueba:
        i. Objetivo que busca desarrollar
        ii. Que indican los resultados obtenidos
        iii. Plan o pasos por seguir.

Recuerde, además, documentar su trabajo:

- Librerías utilizadas
- Los controles de seguridad aplicados
- El proceso de auditar la salud del API

• Evaluación de la práctica.

Ítem	Descripción	Puntos
1	Mezclar las API-Rest para el soporte de datos y elementos de seguridad.	10
2	Creación del Rol y Usuario Root, con el perfil determiando.	10
3	Adaptación de los métodos para el procesamiento de datos.	10
4	Agregar los nuevos requerimientos de seguridad	10
5	Aplicación de prueba de vulnerabilidad por medio de npm audit	15
6	Creación de pruebas de testing para validar el funcionamiento del API-Rest	30
7	Documentación	15
	Puntos	**100**