

Laboratorio Modulo 5: Construcción de APIs y Manejo de Autenticación

Julian F. Latorre

Índice

1. Objetivos	2
2. Requerimientos	2
3. Parte 1: Métodos de Petición HTTP	2
3.1. Descripción	2
3.2. Actividad	2
4. Parte 2: Construcción de APIs con Express.js	3
4.1. Descripción	3
4.2. Actividad	3
5. Parte 3: Integración con Bases de Datos	3
5.1. Descripción	3
5.2. Actividad	3
6. Parte 4: CRUD API en Postman	4
6.1. Descripción	4
6.2. Actividad	4
7. Parte 5: Autenticación y Manejo de Sesiones	4
7.1. Descripción	4
7.2. Actividad	4
8. Entrega y Evaluación: 25 de Septiembre 00:00	4

1. Objetivos

- Evaluar los conocimientos sobre los métodos de petición HTTP: GET, POST, PUT, PATCH, DELETE.
- Desarrollar APIs utilizando `Express.js`.
- Integrar e interactuar con bases de datos relacionales (MySQL) y no relacionales (MongoDB).
- Realizar operaciones CRUD mediante Postman.
- Implementar autenticación y manejo de sesiones utilizando JWT y OAuth 2.0.

2. Requerimientos

- Node.js y npm instalados.
- MongoDB y MySQL instalados y configurados.
- Postman para probar las APIs.
- Editor de código (Visual Studio Code recomendado).

3. Parte 1: Métodos de Petición HTTP

3.1. Descripción

Los métodos de petición HTTP son fundamentales para la comunicación entre cliente y servidor. Los principales métodos son:

- **GET:** Solicita datos del servidor.
- **POST:** Envía datos al servidor para crear un recurso.
- **PUT:** Actualiza un recurso existente en el servidor.
- **PATCH:** Modifica parcialmente un recurso existente.
- **DELETE:** Elimina un recurso del servidor.

3.2. Actividad

1. Crea un archivo `server.js` usando `Express.js`.
2. Implementa rutas para cada método de petición HTTP (`/api/resources`):
 - GET: Devuelve una lista de recursos.
 - POST: Añade un nuevo recurso.

- PUT: Actualiza un recurso existente.
 - PATCH: Modifica parcialmente un recurso.
 - DELETE: Elimina un recurso.
3. Verifica el funcionamiento de cada ruta usando Postman.

4. Parte 2: Construcción de APIs con Express.js

4.1. Descripción

Express.js es un framework web para Node.js que facilita la creación de APIs robustas y escalables.

4.2. Actividad

1. Inicializa un proyecto Node.js con `npm init`.
2. Instala `Express.js` y configura un servidor básico.
3. Implementa una API RESTful que maneje un recurso de tu elección (por ejemplo, `productos`).
4. Define las rutas necesarias para las operaciones CRUD.
5. Agrega manejo de errores y valida los datos de entrada.

5. Parte 3: Integración con Bases de Datos

5.1. Descripción

Las bases de datos relacionales y no relacionales son esenciales para almacenar y gestionar datos en aplicaciones web.

5.2. Actividad

1. Conecta tu API con una base de datos MySQL y MongoDB.
2. Crea un modelo de datos (por ejemplo, `Usuario`).
3. Implementa operaciones CRUD para interactuar con ambas bases de datos:
 - En MySQL, utiliza `sequelize` o `mysql2`.
 - En MongoDB, utiliza `mongoose`.
4. Realiza pruebas de integración usando Postman para asegurar que tu API maneja correctamente los datos en ambas bases de datos.

6. Parte 4: CRUD API en Postman

6.1. Descripción

Postman es una herramienta útil para probar y documentar APIs.

6.2. Actividad

1. Crea una colección en Postman para tu API.
2. Define las solicitudes para todas las operaciones CRUD.
3. Documenta cada solicitud con ejemplos de cuerpo de petición y respuesta.
4. Verifica que todas las rutas funcionen correctamente.

7. Parte 5: Autenticación y Manejo de Sesiones

7.1. Descripción

La autenticación asegura que solo usuarios autorizados puedan acceder a ciertos recursos. JWT y OAuth 2.0 son dos métodos comunes para manejar la autenticación.

7.2. Actividad

1. Implementa la autenticación con JWT en tu API:
 - Crea una ruta de `/login` que genere un token JWT al autenticar al usuario.
 - Protege rutas específicas mediante middleware que valide el token JWT.
2. Implementa OAuth 2.0:
 - Configura un proveedor de OAuth (como Google o GitHub).
 - Integra el flujo de autorización en tu API.
3. Prueba los flujos de autenticación utilizando Postman.

8. Entrega y Evaluación: 25 de Septiembre 00:00

- Subir el código fuente del proyecto a un repositorio de GitHub.
- Incluir un archivo `README.md` con las instrucciones de instalación y uso.
- Entregar un reporte en PDF documentando cada una de las actividades realizadas, incluyendo capturas de pantalla de las pruebas en Postman.