

Proyecto Final: Desarrollo de un CRUD de Usuarios usando Lit y Web Components con API REST

1. Título de la Actividad

Construcción de un Módulo de Gestión de Usuarios mediante Web Components con Lit y API REST conectada a MySQL o PostgreSQL

2. Descripción General

En esta actividad, usted desarrollará un módulo completo de gestión de usuarios utilizando Web Components creados con la librería Lit. El frontend deberá conectarse a un API REST que gestionará las operaciones del CRUD (Crear, Leer, Actualizar y Eliminar) sobre una base de datos MySQL o PostgreSQL.

El objetivo es integrar contenido de desarrollo frontend moderno (Web Components) con desarrollo backend (API REST) y persistencia de datos en un motor relacional.

3. Objetivo de Aprendizaje

Al finalizar la actividad, el estudiante será capaz de:

- Crear Web Components utilizando la librería Lit.
- Implementar componentes reutilizables basados en Shadow DOM.
- Consumir servicios API REST usando fetch().
- Implementar un backend con rutas REST para CRUD.
- Configurar y utilizar MySQL o PostgreSQL para almacenar usuarios.
- Integrar todos los componentes para formar un módulo funcional.

4. Requerimientos de la Actividad

4.1. Frontend (Lit + Web Components)

Debe implementar al menos 3 Web Components, pero usted puede implementar más Web Components:

1. user-form-component

- Formulario para crear y editar usuarios.
- Campos: nombre, correo, rol, estado.

2. user-list-component

- Lista de usuarios obtenidos desde el API REST.
- Debe permitir seleccionar un usuario para editarlo.

- Debe permitir eliminar usuarios.

3. **user-app-component**

- Componente principal que integra los anteriores.
- Controlador central de eventos (crear, actualizar, eliminar).

Cada componente debe:

- Usar **Lit** (lit-element).
- Usar propiedades reactivas.
- Aplicar estilos encapsulados con Shadow DOM.
- Manejar eventos personalizados entre componentes.

4.2. Backend (API REST)

Debe implementar un backend con las siguientes rutas:

Método	Ruta	Descripción
GET	/users	Obtiene todos los usuarios
GET	/users/:id	Obtiene un usuario específico
POST	/users	Crea un nuevo usuario
PUT	/users/:id	Actualiza un usuario
DELETE	/users/:id	Elimina un usuario

El backend puede ser desarrollado con:

- **Node.js + Express**, o
- **PHP + Laravel**, o
- **Python + Flask**, o
- La tecnología que el usted domine o afición.

4.3. Base de Datos

El estudiante debe crear una base de datos llamada:

gestion_usuarios

Con la tabla:

Usuarios:

- id

- nombre
- correo
- rol
- estado

5. Actividades Específicas

1. Diseñar los Web Components usando Lit.
2. Crear la estructura del backend y configurar la conexión con la BD.
3. Implementar el CRUD completo en la API REST.
4. Consumir el API REST desde Lit usando fetch().
5. Diseñar la interacción de los componentes, permitiendo:
 - Registrar usuarios
 - Mostrar listado actualizado en tiempo real
 - Editar usuarios
 - Eliminar usuarios

6. Probar el módulo completo

6. Producto Final

El estudiante debe entregar:

6.1. Código fuente

- Carpeta del backend.
- Carpeta del frontend con Web Components.
- Script SQL de la base de datos.

6.2. Documento PDF (máx. 1 páginas)

Debe contener 2 capturas de pantalla del CRUD funcionando código + Ejecución.

6.3. Evidencia de ejecución

- Subir a repositorio github y dejar el enlace en un comentario de la actividad/tarea/proyecto

7. Rúbrica de Evaluación (sobre 20 puntos)

Criterio	Puntos
Web Components con Lit funcionales	5

Consumo de API REST desde los componentes	4
CRUD backend completo y funcional	4
Integración con MySQL/PostgreSQL	3
Organización del proyecto y buenas prácticas	2
Documento PDF (claridad + capturas + arquitectura)	1
Repositorio github, enlace en comentario moodle	1
TOTAL	20 pts