

---

## Taller 4 – Desarrollo Web

---

**202200007 – Natalia Calderon Echeverria**  
**202106651 – Luis Carlos Corleto Marroquín**  
**202200131 – Carlos Samuel Aguilar Acosta**  
**202200033 – Isai Dardón Mayén**  
**202002907 – Moises Ivan D. Alvarado García**  
**201909035 – Josue Javier Aguilar Lopez**

### Resumen

Como estudiantes de la carrera de Ciencias y Sistemas uno de los ámbitos donde más campo para aplicar que existe en la actualidad es el desarrollo de aplicaciones web y anteriormente los estudiantes ya han tenido un acercamiento con este tipo de aplicaciones de una manera mucho más general. Es necesario que los estudiantes se empiecen a familiarizar con herramientas que van a utilizar a lo largo de la carrera y que les puede servir en un ambiente laboral, estas herramientas son mayormente conocidas como Frameworks.

### Palabras clave

- MySQL
- Consulta SQL
- Respuesta JSON
- Conexión a la base de datos
- Cursos y Publicaciones

### Introducción

Al inicio de cada periodo estudiantil los estudiantes de la facultad de Ingeniería suelen tener dudas como las siguientes: “¿Qué catedrático es bueno para X curso?”, “¿Qué tan recomendable es llevar un curso?”, “¿Que tan recomendable es un catedrático en general?”, etc. Actualmente contamos con un grupo en Facebook donde los estudiantes pueden solventar estas dudas, pero se sabe que no es la manera más viable para manejar esta información, ya que el grupo no solo se encarga de calificar catedráticos, sino que se comparte información relevante para la facultad y noticias que suelen ser de interés para un grupo de estudiantes. Se quiere implementar una Aplicación Web para que los estudiantes tengan acceso a un récord de los catedráticos que han impartido uno o más cursos en la facultad. El fin de la aplicación es que los estudiantes sean los responsables de darle un récord al catedrático o auxiliar por medio de publicaciones y que los mismos estudiantes puedan darle una retroalimentación a cada publicación para que los estudiantes que tienen duda sobre un catedrático, un auxiliar o un curso tengan más opiniones en una misma publicación

## **Backend**

### **App**

#### **Config:**

- **Conection:** este módulo proporciona una interfaz para interactuar con una base de datos MySQL utilizando Node.js, gestionando las conexiones de forma adecuada, ya sea cerrándolas después de cada consulta (función `closeableQuery`) o manteniéndolas abiertas para ejecutar múltiples consultas en la misma conexión (función `uncloseableQuery`). Además, se carga la información de configuración de la base de datos desde un archivo `.env` para mantener las credenciales fuera del código fuente.

#### **Controllers:**

- **addComentario:** se encarga de recibir datos de comentario a través de una solicitud HTTP, los introduce en una base de datos MySQL y proporciona una respuesta JSON que indica si la operación se completó exitosamente o si se encontró un error en el servidor.
- **addCursoAprobado:** se encarga de registrar cursos aprobados por usuarios en una base de datos y proporciona respuestas para indicar el estado de la operación, ya sea exitosa o con errores.
- **create:** tiene la función de configurar una base de datos, definir su estructura y llenar ciertas tablas con datos iniciales a partir de archivos CSV, lo que facilita la preparación de la base de datos para su uso.
- **createUser:** esta función se encarga de manejar la creación segura de usuarios en la base de datos, cifrando sus contraseñas y proporcionando respuestas JSON que reflejan el resultado de la operación.
- **getCatedraticos:** se utiliza para obtener nombres de catedráticos desde una base de datos y devolverlos como una respuesta JSON. Esta

función maneja tanto los casos exitosos como los errores en la consulta a la base de datos.

- **getComentarios:** se utiliza para obtener y entregar comentarios relacionados con una publicación específica. La función maneja situaciones en las que no hay comentarios y también gestiona errores durante la consulta a la base de datos.
- **getCursos:** en este código se encarga de recuperar una lista de nombres de cursos desde una base de datos MySQL y devolverlos como una respuesta en formato JSON. La función maneja situaciones en las que no hay cursos disponibles y también gestiona errores durante la consulta a la base de datos.
- **getCursosAprobados:** se utiliza para obtener nombres de cursos aprobados por un usuario. La función maneja casos de éxito y errores durante la consulta a la base de datos.
- **getCursosExistentes:** se utiliza para obtener y proporcionar la lista de cursos existentes junto con sus catedráticos desde la base de datos en formato JSON. La función maneja tanto los casos exitosos como los errores durante la consulta a la base de datos.
- **getCursosPendientes:** se utiliza para obtener y proporcionar la lista de cursos pendientes por un usuario desde la base de datos. La función maneja tanto los casos exitosos como los errores durante la consulta a la base de datos.
- **getPublicación:** tiene como finalidad proporcionar detalles específicos de una publicación almacenada en una base de datos MySQL en formato JSON. Esta función se encarga de manejar situaciones en las que la publicación no existe y también gestiona posibles errores que puedan surgir durante la consulta a la base de datos, asegurando una respuesta adecuada en cada caso.
- **getPublicaciones:** Se utiliza para consultar la base de datos y obtener la lista de publicaciones, luego se almacenan en la variable `publicaciones`

para después la función `getPublicaciones` de exporte a otros módulos.

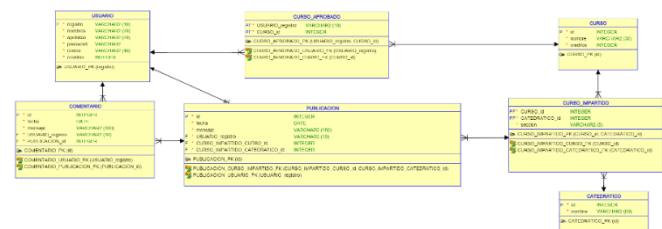
- **getUser:** Se encarga de consultar la base de datos y obtener información de un usuario en función de un número proporcionado en la solicitud HTTP. Se almacena en la variable “results” y si esta contiene información de responder con un estado HTTP 200 y se envían los resultados en formato JSON, por último, se exporta la función.
- **LoginUser:** Esta función se usa para autenticar a un usuario, con su número de registro y contraseña. La autenticación es exitosa si se encuentra un registro de usuario con el número de registro proporcionado y si la contraseña coincide con la almacenada de manera segura en la base de datos.
- **Publicar:** Se encarga de agregar nueva información a la base de datos, luego verifica que el curso y el catedrático existan, de ser así, devuelve un mensaje en formato json para indicar si ocurrió un error o fue exitoso.
- **recoverUser:** Se encarga de recuperar la contraseña del usuario con su número de registro, correo electrónico y la nueva contraseña. De la misma manera devuelve mensajes de error o si la operación fue exitosa.
- **searchPublicacionCatedratico:** Se utiliza para buscar publicaciones con el nombre del catedrático en la base de datos, en la búsqueda se tomo en cuenta la manera en la que se solicita la búsqueda por lo que se puede utilizar mayúscula y minúscula, además, maneja errores y mensajes de éxito en formato json.
- **searchPublicacionCurso:** Busca publicaciones con el nombre del curso proporcionado en la base de datos, de igual manera se tomó en cuenta la manera en hacer la consulta y maneja los errores y mensajes en formato json
- **searchUser:** Esta función se utiliza para buscar a los usuarios en la base de datos con la

información proporcionado de número de registro, la búsqueda devuelve un formato json incluyendo el número de registro y el nombre completo de los usuarios que coincidan.

### Controllers:

- **Routes:** Este código configura las rutas y las funciones controladoras para una aplicación web en Express.js, cada ruta está asociada a una función controladora específica que maneja la lógica de la aplicación para esa acción particular.
- **App:** Configura la aplicación web utilizando Express.js, define middleware para procesar solicitudes y habilita CORS, también importa las rutas del archivo `routes.js` para organizar el enrutamiento de la aplicación.
- **Index:** Inicia el servidor web utilizando la aplicación Express previamente configurada.

### Diagrama de Clases:



### Conclusiones

Se familiarizó con frameworks de desarrollo web, utilizaron repositorios para el control de versiones y administraron información en una base de datos.

Se implementó con éxito una aplicación web que permitió a los estudiantes evaluar catedráticos y cursos de manera eficiente. La aplicación se desarrolló como una versión de prueba, centrándose en el área de sistemas y utilizando Angular en el

cliente, NodeJS en el servidor y una base de datos (MySQL).

Se hizo un uso efectivo del control de versiones utilizando GitHub. El repositorio en línea facilitó la colaboración entre los miembros del equipo de desarrollo y permitió un seguimiento detallado del progreso del proyecto.

La aplicación incluyó funciones esenciales como inicio de sesión, registro de usuarios, recuperación de contraseña, visualización de publicaciones, filtros de búsqueda por curso o catedrático, creación de publicaciones y comentarios, y visualización de perfiles de usuarios. Además, se implementó la gestión de cursos aprobados, proporcionando a los usuarios la capacidad de administrar esta información.

La aplicación abordó la necesidad de los estudiantes de obtener información sobre catedráticos y cursos de manera más eficaz. Los comentarios en las publicaciones permitieron que los usuarios proporcionaran retroalimentación adicional, mejorando la utilidad de la plataforma.

## **Referencias bibliográficas**

Ing. Herman Véliz., Taller 4, “Desarrollo Web”.  
2023