Instituto Técnico Superior Córdoba
Tecnicatura Superior en Desarrollo de Software
Práctica Profesionalizante I

Trabajo final:
Implementación de API REST para una red social

Descripción:

El objetivo de este trabajo es desarrollar una **API REST** utilizando **Node.js**, **Express**, y **MySQL** como base de datos, que sirva de backend para una red social. El sistema manejará tres tablas principales: **Usuario**, **Post** y **Following**, cada una con sus respectivas funcionalidades. A continuación, se describen los requisitos y endpoints para cada tabla.

Tabla: Usuario

Esta tabla representa el perfil de un usuario en la red social. Los campos que la componen son: **nombre**, **nickname**, **mail**, **password** (encriptada) y **avatar**.

> El prefijo de las rutas será `/api/usuarios`.
> Endpoints requeridos:
> 1. `POST /register` Permite registrar un nuevo usuario.
> 2. `GET /` Lista todos los usuarios registrados.
> 3. `POST /login` Autentica al usuario mediante **mail** y **password**.
> 4. `PUT /me` Permite al usuario editar su perfil (requiere autenticación).

Restricciones:

> No se permiten dos usuarios con el mismo **mail** o **nickname**.

Tabla: Post

Esta tabla representa las publicaciones realizadas por los usuarios. Sus campos son: **id_usuario**, **título**, y **contenido**.

> El prefijo de las rutas será `/api/posts`.
> Endpoints requeridos:
>
> 1. `POST /` Crea una nueva publicación (requiere autenticación).
> 2. `GET /` Lista todas las publicaciones (requiere autenticación).
> 3. `PUT /:id` Modifica una publicación existente (requiere autenticación).

4. `DELETE /:id` Elimina una publicación (requiere autenticación).
5. `GET /:id` Muestra una publicación específica (requiere autenticación; solo accesible si el post es propio o pertenece a un usuario seguido).
6. `GET /user-posts/:id` Muestra los posts de un usuario determinado (requiere autenticación; solo accesible si sigo a dicho usuario).

Tabla: Following

Esta tabla define la relación de seguimiento entre usuarios. Los campos son: **id_usuario** (el que sigue) e **id_usuario_seguido** (el seguido).

El prefijo de las rutas será `/api/following`.
Endpoints requeridos:

1. `POST /` Agrega una nueva relación de seguimiento (requiere autenticación).
2. `DELETE /` Elimina una relación de seguimiento (requiere autenticación).
3. `GET /following` Lista a los usuarios que sigo (requiere autenticación).
4. `GET /followers` Lista a los usuarios que me siguen (requiere autenticación).
5. `GET /mutual` Lista a los usuarios con los que existe un seguimiento mutuo (requiere autenticación).

Restricción:

No se permite que un usuario se siga a sí mismo; debe devolver un mensaje de error.

Herramientas y Consideraciones:

Se utilizará **Sequelize** como ORM para interactuar con la base de datos MySQL.
La API deberá estar **documentada** utilizando **Swagger**.
El trabajo se subirá a un **repositorio en GitHub**, donde se verificará el historial de commits (debe reflejar el progreso continuo del desarrollo, no solo un único commit).

Entrega:

1. **Código en GitHub**: El código de la API deberá estar subido a un repositorio público o privado (si se indica así) en GitHub.
2. **Informe escrito**: Se deberá acompañar el código con un informe que detalle el proceso de desarrollo, las decisiones técnicas y las dificultades encontradas.

Este trabajo será evaluado en función de la correcta implementación de la API, el uso de las tecnologías solicitadas, la calidad del código, la documentación y el informe.