**Trabajo Práctico Nro. 8 Autenticación JWT**

**Ejercicio Nº 1: Plataforma educativa**

Una universidad está desarrollando una **plataforma educativa en línea** para administrar usuarios, cursos y recursos educativos. Como parte del equipo de desarrollo, se te ha asignado la tarea de implementar la funcionalidad de gestión de usuarios para la API REST de la plataforma.

**Requerimiento:** Desarrollar una API RESTful utilizando Spring Boot que permita a la universidad administrar usuarios en su plataforma educativa, garantizando la seguridad de los datos mediante Spring Security y JSON Web Tokens (JWT).

**Se deberá considerar:**

1. Implementar un **modelo de datos** que incluya entidades como **User, Role, y Permission** donde los usuarios pueden tener los roles: **Estudiante, Profesor o Administrador**.
2. Configurar **Spring Security** para proporcionar autenticación y autorización basadas en **JWT**.
3. Implementar un servicio de autenticación que emita JWTs para usuarios válidos.
4. Proteger las rutas de la API para que solo los usuarios autenticados y autorizados puedan acceder a ellas (salvo la ruta de autenticación que debe ser de acceso libre).
5. Implementar controladores para realizar operaciones CRUD en las entidades de usuario, rol y permiso.  Establecer que SOLO los usuarios con rol ADMINISTRADOR podrán crear nuevos usuarios o editarlos.
6. Probar exhaustivamente la API utilizando herramientas como Postman o cURL, asegurándote de que las operaciones de autenticación y autorización funcionen correctamente.

**Ejercicio Nº 2: Incremento a la plataforma educativa**

La implementación de la funcionalidad de la gestión de usuarios del Ejercicio Nº 1 para la universidad fue todo un éxito por lo que, a partir de ese proyecto base y sus configuraciones de seguridad se te ha asignado llevar a cabo los siguientes requerimientos:

1. Modelado completo de estudiantes, profesores y cursos
2. CRUD completo de estudiantes
3. CRUD completo de profesores
4. CRUD completo de cursos

**Para esto tener en cuenta:**

1. Crear los correspondientes controllers, services y repositories para llevar a cabo cada operación de CRUD implementando de igual manera los endpoints correspondientes.
2. Establecer los siguientes permisos:
   * Los usuarios **administradores** pueden realizar todas las operaciones CRUD sobre todas las entidades.
   * Los usuarios **estudiantes** solo podrán leer/consultar los cursos y estudiantes disponibles.
   * Los usuarios **profesores** podrán leer tanto cursos, como profesores, como estudiantes.
   * **Extra (opcional):** Los profesores tendrán permiso de EDICIÓN sobre los cursos en los que ellos son profesores.
3. Para llevar a cabo el modelado tener en cuenta:
   * Un curso tiene una lista de alumnos y solo un profesor asignado.
   * Un profesor puede dar más de un curso.
   * Un alumno puede tener uno o varios cursos asignados.