**TP Integrador Final**

**Objetivo**

El objetivo de este proyecto integrador final es el de validar los conocimientos prácticos y técnicos referidos al desarrollo seguro en el lenguaje de programación Java mediante Spring Boot y Spring Security.

Al mismo tiempo, este proyecto busca ser desarrollado para formar parte del futuro portfolio del estudiante para que sea una herramienta de presentación/demostración de conocimientos y habilidades ante una propuesta laboral.

**Escenario**

Estas trabajando en un proyecto de una api. Para esta situación, el cliente de la web donde se alojara la aplicación nos especificó como tarea desarrollar un api que se encargará de diferentes aspectos del software:

• **CRUD** de al menos una relación entre 2 entidades de usuario

• **CRUD** de usuarios, roles y permisos

En base a las especificaciones técnicas relevadas en un análisis funcional, se especifican las siguientes medidas de seguridad para el proyecto:

• Para acceder a cualquier endpoint de la aplicación es necesario estar autenticado y la asignación de roles y permisos a tener en cuenta para el correcto acceso son:

**Roles**

• Entidad de Usuario (por ejemplo: Cliente, Producto, Medico, Usuario, Alumno, etc.)

• User

• Admin

**Permisos:**

• READ

• UPDATE

• CREATE

• DELETE

Asignación de Permisos en base a rol:

• ADMIN: **CREATE, READ, UPDATE, DELETE** sobre todos los endpoints (incluidos aquellos para operaciones CRUD de usuarios, roles y permisos).

• USER: **READ** para ambas entidades de usuario

• ENTIDAD DE USUARIO:

• **CREATE**: para una sola entidad de usuario

• **READ**: para ambas entidades de usuario

• **UPDATE**: para una sola entidad de usuario

**Nota**: Tener en cuenta que todos los usuarios creados deberán poder tener acceso al endpoint de autenticación/login a desarrollar.

**Requerimientos del Backend**

*A partir del relevamiento realizado respecto, tener en cuenta los siguientes requerimientos:*

1. Diagramar (dibujar/realizar un gráfico representativo de planteo) el modelo de clases a llevar a cabo, incluyendo el manejo de usuarios, roles y permisos.

2. Llevar a cabo los correspondientes CRUD y operaciones necesarias.

3. Implementar un end-point para autenticación mediante login **(/auth/login).**

4. Implementar la tecnología JWT para el manejo de tokens de autenticación.

5. Implementar correctamente el uso de roles, usuarios y permisos a partir de los datos almacenados en una base de datos.

6. Implementar por lo menos una forma de autenticación mediante OAuth2.

7. Implementar encriptado de contraseñas mediante BCrypt.

8. Establecer los permisos mediante annotations en los diferentes controllers en base a la asignación de permisos especificada.

9. Crear al menos un usuario de cada rol para hacer pruebas.

10. Desplegar la aplicación en Docker o directamente en un hosting (se recomienda **Vercel** para el frontend y **Render** para el backend y base de datos en MySQL). Documentar el proceso de despliegue.

**Importante**: **manejar el versionado del desarrollo de software mediante Git/GitHub**

**Requerimientos del Frontend**

Una vez completado el backend, proceder a desarrollar el frontend consumiendo la API RESTful.

1. Desarrollar la interfaz gráfica utilizando HTML, CSS y Bootstrap/Tailwind para el maquetado y utilización de componentes gráficos o manejar la vista desde el lado del servidor mediante plantillas Thymeleaf.
2. Integrar JavaScript (nativo o mediante librería React) para manejar interacciones y consumo de API utilizando la función nativa Fetch o la librería externa Axios.

**Documentación a entregar:**

Subir en el espacio del Aula Virtual el enlace a github donde se encontrara la aplicación web completa, diagrama de entidad-relación y la colección de Postman/Thunder Client/Curl/etc. para realizar el testing de api.