## Especificación de Mecanismos de Seguridad (Autenticación y Autorización)

La aplicación implementa un mecanismo de seguridad basado en JSON Web Tokens (JWT) para manejar tanto la autenticación como la autorización de los usuarios.

Autenticación: Al iniciar sesión, los usuarios envían sus credenciales (correo y contraseña). Si las credenciales son válidas, el sistema genera un token JWT firmado que contiene información relevante como el ID del usuario y su rol. Este token se retorna al cliente y debe ser incluido en futuras solicitudes mediante el encabezado Authorization: Bearer <token>.

Autorización: El backend analiza el JWT en cada solicitud y verifica que el usuario tenga los permisos adecuados para acceder al recurso solicitado. El rol del usuario es utilizado para aplicar restricciones a nivel de endpoints mediante anotaciones como @PreAuthorize.

## Especificación del Protocolo de Comunicación Interna

La aplicación está construida bajo una arquitectura monolítica modular, en la cual cada módulo representa un dominio funcional específico (por ejemplo, usuarios, roles, equipos, etc.).

Cada módulo contiene su propia estructura interna, que incluye:

* Entidades
* DTOs (Data Transfer Objects)
* Controladores
* Servicios
* Repositorios

Dentro de la aplicación, la comunicación entre módulos se realiza mediante invocaciones directas en memoria, respetando el principio de separación de responsabilidades. No se utilizan mecanismos externos como mensajería o gRPC, ya que todo ocurre dentro del mismo proceso de ejecución.

En cuanto a la comunicación externa (entre cliente y servidor), esta se lleva a cabo mediante una API HTTP REST, siguiendo una arquitectura orientada a recursos. Las solicitudes y respuestas se envían en formato JSON, y cada petición del cliente debe incluir el token JWT para autenticar al usuario.