Modelo de usuario y rol

Implementa la seguridad de tu API Rest con Spring Boot

Modelo de usuario

- Representará a una persona que utilice nuestro sistema.
- Información básica: nombre de usuario, contraseña y avatar.
- Además, el rol o roles que tiene dicho usuario.
- Lo implementamos como una entidad de JPA para almacenarlo fácilmente en una base de datos.

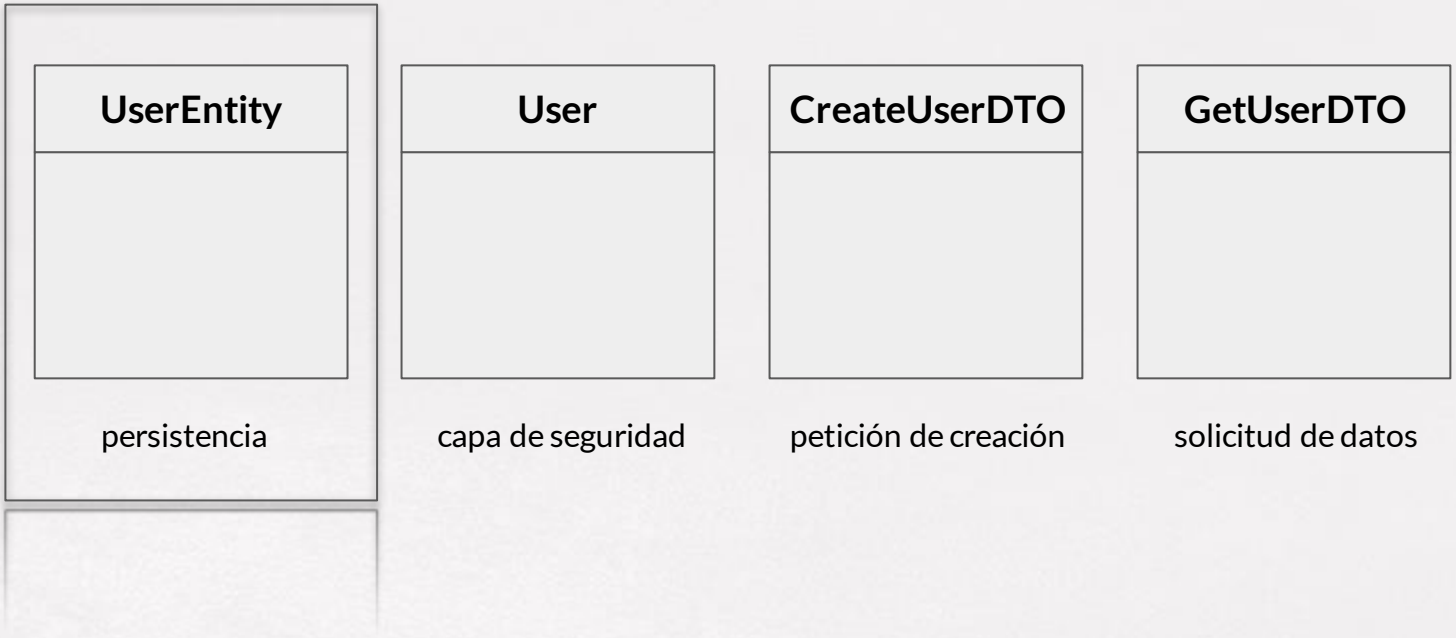
Alternativas

- Si nuestro modelo de usuario implementa *UserDetails*
  - ***Desventaja***: más acoplado a Spring Security (y sus posibles cambios).
  - ***Ventaja***: más integrado con Spring Security (Authentication = nuestro modelo de usuario).

Alternativas

- Si nuestro modelo de usuario no implementa *UserDetails*
  - ***Desventaja***: más integrado con Spring Security (necesitamos transformar, en algún punto, nuestra entidad usuario en algo que implemente a *UserDetails*).
  - ***Ventaja***: menos acoplado con Spring Security (nos afectarán menos los cambios).
- **Escogemos la primera alternativa.**

Múltiples clases para gestionar los usuarios



¡A por el código!

- Clase entidad para *UserEntity*
- Enumeración *UserRole*
- Anotación en una clase de configuración (por ahora nos vale la clase principal).

Reto

- Plantea algunos campos más que pudiera tener la clase
- Revisa los campos y añade las anotaciones de validación que consideres oportunas (curso de Spring Boot)
- Añade los campos necesarios para gestionar de verdad los atributos *accountNonExpired*, *accountNonLocked*, *credentialNonExpired* o *enabled*. Puedes revisar la documentación de la interfaz UserDetails.