

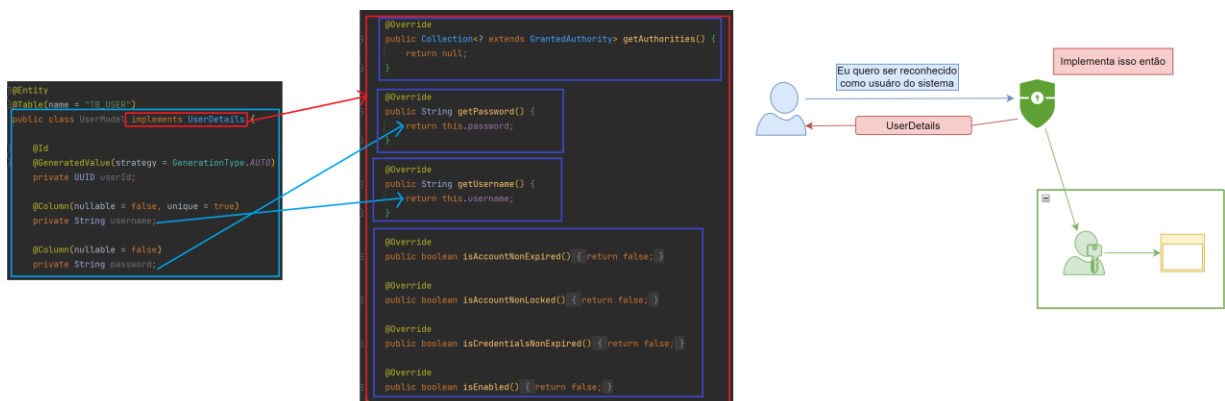


Autenticação em banco de dados - Spring Security

Obviamente uma autenticação em memória não seria útil pra um cenário web, e é por isso que podemos fazer autenticação de usuários que são persistidos em **banco de dados**.

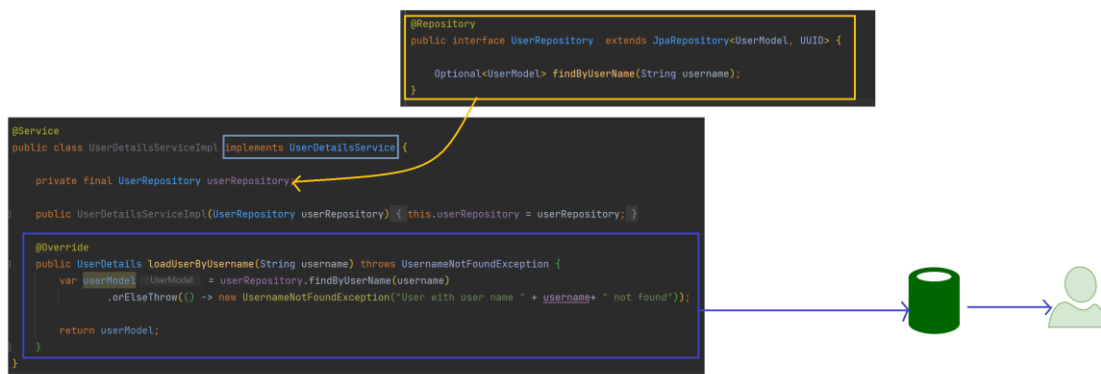
A aplicação provavelmente vai ter um **modelo de usuário** que corresponde ao domínio, até aí tudo bem, mas é necessário que **esse modelo de usuário** esteja disponível para o **Spring Security** enxerga-lo e utiliza-lo para fazer autenticação e autorização internamente.

Fazemos isso **implementando a Interface de UserDetails** no **modelo de usuário**, essa interface **traz uma série de assinaturas que representam informações do usuário no sistema** para que o **Spring Security** garanta que pode fazer o que é necessário, basicamente **ele diz**: *"Quer usar meus recursos, então me fala o que no seu **modelo de usuário** representa essas características de segurança que eu preciso"*:



Agora que o **modelo de usuário** é conhecido pelo **Spring** precisamos fazer a **implementação do serviço que vai autenticar esse usuário**. Fazemos isso através da **UserDetailsService**, é como se o **Spring Security** falasse: *"Beleza, se esse **modelo** é o usuário no sistema, então **implementa o serviço** que garante isso mesmo"*.

Essa **Interface** tem apenas uma **única assinatura**, que é a de *"loadByUsername"*, onde precisamos simplesmente **buscar no nosso banco de dados o usuário com o nome de acesso informado** e retorna-lo:



E por último, mas não menos importante é necessário **alterar a configuração do Adapter de segurança para que ele faça a autenticação** através da nossa **classe de implementação da UserDetailsService**:

