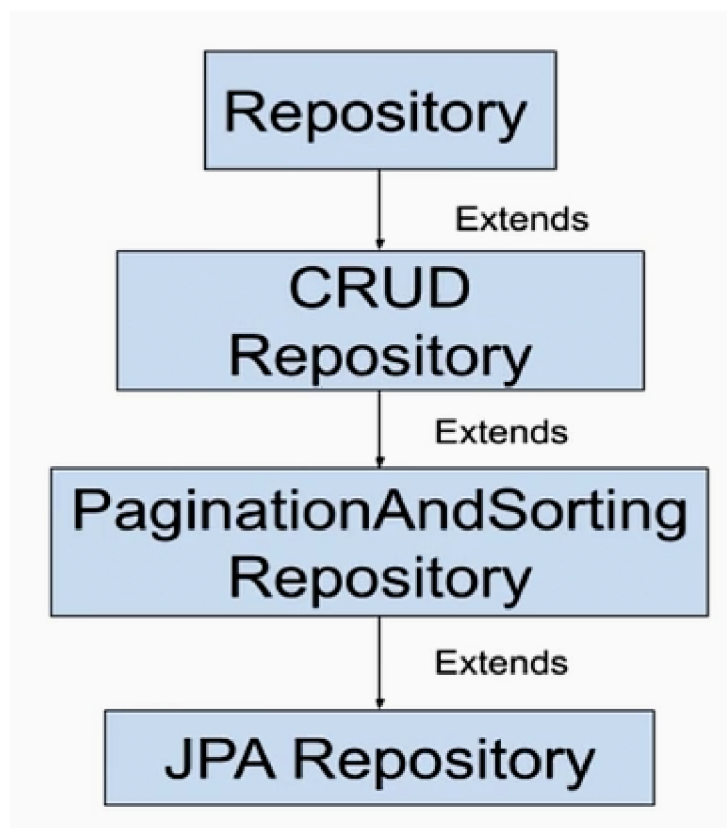




## Usando CrudRepository

### Transcrição

[00:00] Olá. Bem-vindos a mais uma aula. Agora que nós já temos uma tabela no banco de dados, nós precisamos popular essa tabela. Como podemos fazer isso? Através de Repositorys do Spring Data. Conforme podemos ver aqui na tela, existem vários Repositorys. O CRUD Repository, PaginationAndSortingRepository. Cada Repository do Spring Data tem um determinado propósito.



[00:24] Mas fica tranquilo que no futuro nós vamos ter uma aula só para visualizar o que são cada um desses Repositorys. Só estou mostrando agora para que saibamos que existe esse Repository dentro do Spring Data e que nós precisamos utilizar ele para manipular o banco de dados. Vamos à

implementação. Abrindo o Eclipse. Vou vir no projeto, dentro da pasta do "Source Java" pasta principal.

[00:59] Vou criar uma Package, clicando em "New > Package" e vou dar o nome dessa Package de "Repository". Agora que eu já tenho a Package de Repository vou clicar com o botão direito e clicar em "New > Interface" e vou chamar esse Repository de "CargoRepository". Agora que eu já tenho esse Repository, vou fazer o `implements` e como eu disse, existem vários Repositors.

[01:40] Como aqui eu só quero salvar no banco de dados, então eu quero fazer funções de Crud. Então vou utilizar o `extends CrudRepository`, porque o `CrudRepository` também é uma interface. Então tem que ser "extends".

[02:15] Agora eu queria mostrar um comparativo para vocês entre o Spring Data e o JPA. Para quem já desenvolveu um JPA para vermos a facilidade que esse modelo de implementação me dá. Antes de eu fazer isso, só vou adicionar aqui o `@repository` porque como eu havia dito na aula de criação do projeto, que esse `@SpringBootApplication` era para ele varrer a aplicação e ler todas as anotações que temos dentro da nossa aplicação, anotações Spring.

[02:52] E essa é uma anotação Spring que ele vai ler. Deixa eu criar uma classe. Você não precisa criar essa classe. Essa classe é só para mostrar para vocês o comparativo com o JPA puro se fossemos fazer com o JPA. Vou clicar aqui em "repository" com o botão direito, "New > Class > JpaExemplo". E eu já tenho o comando, vou copiar e colar. Deixa eu só fazer o *Import*. Se fossemos utilizar uma classe em JPA, teríamos algo parecido com isso.

[03:40] Teríamos um `ManagerFactory`, teríamos o `EndityManager`. O `EndityManager` que seria criado a partir do `EndityManagerFactory`, e aí nós teríamos que criar esses métodos aqui. O método `save`, o método `findAll`, um `findById`, `deleteById` e aí você teria o quê? Teria que pegar o `EndityManager`, criar uma transação, persistir a transação, comitar a transação e fechar a transação.

[04:05] Isso você tem que criar na mão. É função de o desenvolvedor fazer isso quando se utiliza o JPA. Isso acaba sendo um pouco custoso porque você tem que criar isso na mão e acaba levando um certo tempo. E é por isso que utilizar o Spring Data é tão rápido. Porque basicamente tudo isso que eu já fiz aqui, eu só coloquei algumas funções, o Repository do Spring Data já traz para nós. Vamos entrar aqui no CrudRepository.

[04:38] Eu já tenho um `save`, um `saveAll`, um `findById`, um `existsById`, o `findAll` e o outro é `findById`, tenho um `findAllById`, um `count`, um `delete`. Então eu já tenho todos esses métodos implementados que eu só preciso usar, não preciso desenvolver. Por isso acaba sendo muito rápido partir para o Spring Data que já cria essa casca com o JPA por baixo do que desenvolver o JPA na mão. Você ganha em velocidade. Então agradeço por verem esse vídeo. Espero vocês na próxima aula.