

Spring

Execute o seguinte comando no PostgreSQL (ou crie tabela similar no SGBD de sua preferência):

```
1 create table tra_trabalho (  
2     tra_id bigint generated always as identity,  
3     tra_titulo varchar(100) not null unique,  
4     tra_data_hora_entrega timestamp not null,  
5     tra_descricao varchar(200),  
6     tra_grupo varchar(20) not null,  
7     tra_nota int,  
8     tra_justificativa varchar(100)  
9 );  
10  
11 insert into tra_trabalho (tra_titulo, tra_data_hora_entrega, tra_grupo, tra_nota, tra_justificativa)  
12 values ('Teste 1', current_timestamp, 'Alpha', 6, 'Bom, mas falta conteúdo'),  
13        ('Teste 2', current_timestamp, 'Beta', null, 'Incompleto');  
14
```

Atividade:

- 1) Crie uma classe que mapeie a tabela criada;
- 2) Crie um repositório para a classe criada no item 1 contendo uma consulta capaz de buscar todos os trabalhos que contenham uma palavra dentro de seu título com nota superior a um número inteiro. Tanto a palavra quanto o número devem ser recebidos como parâmetro do método;
- 3) Crie um serviço com 3 métodos: um para cadastro, um para listar todos os registros da tabela e outro para buscar usando a consulta criada no item 2. No cadastro verifique se o título e o grupo estejam preenchidos, lançando um erro em caso contrário. Ainda no cadastro, use a data/hora atual se a data/hora de entrega não estiver preenchida;
- 4) Crie um controller que atenda na URL “/trabalho” e contenha 3 rotas: uma para cadastro, outra para listar todos os registros e uma última para executar a busca por título e nota criada no item 2.

Importante: Utilize `java.time.LocalDateTime` para mapear timestamp.