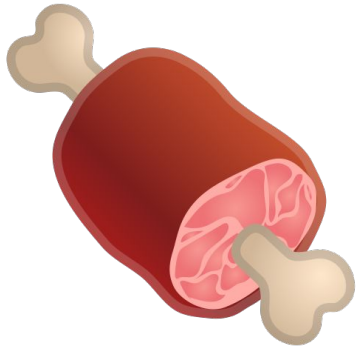
Prática de TDD, Refactoring usando Caso de Ensino

Grupo 2 - Adriana de Arruda, Henrique Kalife, Lucas Lotar do Amaral Figueira, Maximiliano Alves da Cruz e Renan Carvalho de Oliveira Fernandes

UNISINOS

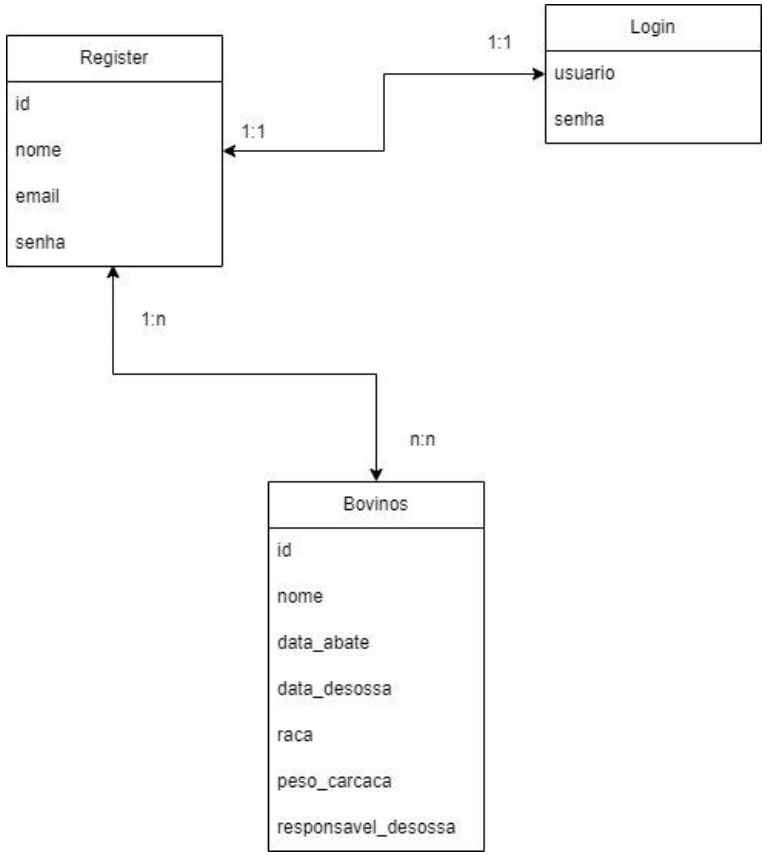
UNISINOS

Projeto api-desossa



Para fazer os testes foi escolhido o projeto de API para o Módulo de Desossa da PEDRAMOURA que foi construída utilizando a stack Typescript, NestJS e Prisma, para a parte de qualidade estamos utilizando o Jest como framework de teste e FakeJS como gerador de massa de dados.

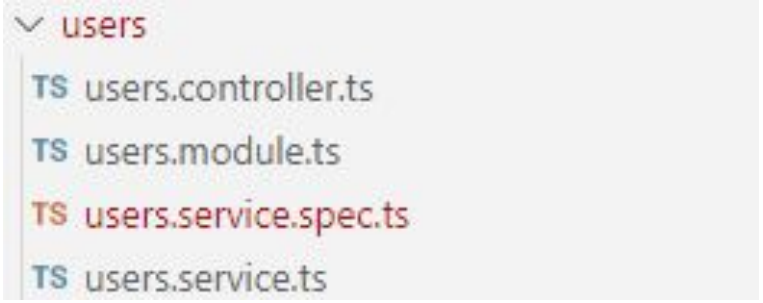
UML - api desossa parcial



Feature: Gerenciamento de usuários

Essa feature é responsável por buscar, adicionar e validar as regras de negócios de novo usuários dentro da plataforma.

-



UNISINOS

Cenários de testes

UsersService - findOneByUsername
Cenário 1: Deve retornar um usuário válido.
Dado que quero buscar um usuário na base
Quando enviar os dados válidos
Então terei encontrado um usuário

Cenário 2: Busca por usuário inexistente.
Dado que quero buscar um usuário inexistente
Quando enviar os dados inexistente
Então NÃO terei encontrado um usuário

UsersService - addUsers
Cenário 3: Deve adicionar um novo usuário com sucesso
Dado que desejo criar um novo usuário
Quando enviar os dados válidos
Então terei criado um novo usuário

Cenário 4: Não deve adicionar um novo usuário com duplicidade
Dado que desejo criar um usuário com o mesmo nome de usuário já existente
Quando eu enviar os dados em duplicidade
Então não terei criado um novo usuário

CreateUserBodyDto
Cenário 5: Deve ser criado o DTO com sucesso
Dado que desejo criar um objeto válido
Quando inserir os dados de usuário
Então terei um objeto de usuário válido

Cenário 6: Não deve ser criado o DTO com sucesso - Nome vazio
Dado que desejo criar um objeto sem nome
Quando inserir os dados sem nome
Então terei um erro informando que o usuário deve ter um nome.

Cenário 7: Não deve ser criado o DTO com sucesso - Email vazio
Dado que desejo criar um objeto sem Email
Quando inserir os dados sem Email
Então terei um erro informando que o usuário deve ter um Email.

Cenário 8: Não deve ser criado o DTO com sucesso - usuario vazio
Dado que desejo criar um objeto sem usuario
Quando inserir os dados sem usuario
Então terei um erro informando que o usuário deve ter um usuario.

Cenário 9: Não deve ser criado o DTO com sucesso - senha vazio
Dado que desejo criar um objeto sem senha
Quando inserir os dados sem senha
Então terei um erro informando que o usuário deve ter um senha.

UNISINOS

Aplicação no código

```
describe('addUsers', () => {
  it('Cenário 3: Deve adicionar um novo usuário com sucesso', async () => {
    const bodyToAddUser = {
      nome: faker.person.fullName(),
      email: faker.internet.email(),
      usuario: faker.internet.userName(),
      senha: faker.internet.password(),
    };
    await usersService.addUsers(bodyToAddUser).then((result: any) => {
      expect(result.user.nome).toBe(bodyToAddUser.nome);
      expect(result.user.email).toBe(bodyToAddUser.email);
      expect(result.user.usuario).toBe(bodyToAddUser.usuario);
      expect(result.user.senha).toBe(bodyToAddUser.senha);
    });
  });
```

UNISINOS

Execução dos testes

UNISINOS

Execução dos testes

```
$ npm run test

> api-desossa@0.0.1 test
> jest

 PASS  src/users/users.service.spec.ts (6.133 s)
 PASS  src/dtos/create-user-body.dto.spec.ts (6.107 s)

Test Suites: 2 passed, 2 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        6.87 s, estimated 8 s
Ran all test suites.
```

UNISINOS

Coverage

```
$ npm run test:cov

> api-desossa@0.0.1 test:cov
> jest --coverage

 PASS  src/dtos/create-user-body.dto.spec.ts (5.659 s)
 PASS  src/users/users.service.spec.ts (5.71 s)
--------------------------|---------|----------|---------|---------|-------------------
File                      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
--------------------------|---------|----------|---------|---------|-------------------
All files                 |      95 |       50 |     100 |   94.44 |
 dtos                     |     100 |      100 |     100 |     100 |
  create-user-body.dto.ts |     100 |      100 |     100 |     100 |
 users                    |   92.85 |       50 |     100 |   91.66 |
  users.service.ts        |   92.85 |       50 |     100 |   91.66 | 32
--------------------------|---------|----------|---------|---------|-------------------

Test Suites: 2 passed, 2 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        6.347 s
Ran all test suites.
```

UNISINOS

Coverage

All files

95% Statements 19/20 **50%** Branches 1/2 **100%** Functions 3/3 **94.44%** Lines 17/18

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File	Statements		Branches		Functions		Lines	
dtos	100%	6/6	100%	0/0	100%	0/0	100%	6/6
users	92.85%	13/14	50%	1/2	100%	3/3	91.66%	11/12

Obrigada.

UNISINOS