

Atividade - Testes automatizados com JEST

Lucas Daniel Batista, RA: 2159171

- Primeiro momento, configurações:
 - Configurações importantes para limpeza das variáveis a cada nova execução dos testes;
 - Configurações para modificações nos testes não reiniciarem o servidor;
- Segundo momento, passo a passo para criação dos testes unitários e de integração:
 - Primeiro passo, descrever o que o teste deve fazer, de maneira clara e objetiva, dentro das funções `it()`, que podem estar dentro da função `describe()`;
 - Depois, escrever o processo do teste até chegar ao valor esperado;
 - Então, é testado se o valor obtido é o valor esperado com a função `expect()`;
 - Utilizado o comando “`yarn test`” no terminal para executar os testes.
- Desenvolvimento do teste de integração
 - É feito um método para testar a criação de usuários e suas autenticações;
 - Criada uma rotina para deleção dos dados da base a cada teste;
 - Criada um novo teste no mesmo método para testar um cenário onde o usuário não deve ser criado, o cenário inverso do primeiro teste;
 - Criado novo teste para garantir que o método de login está retornando um token para autenticação do usuário;
 - Criado um teste para garantir que o usuário pode acessar rotas privadas quando autenticado;
 - Criado um teste contrário ao anterior;
- Desenvolvimento do teste unitário
 - É desenvolvido um método para testar se o hash da senha do usuário está sendo gerado corretamente;
- Recomendações:
 - Utilizar uma base de dados à parte para rodar os testes, pode ser uma SQLite, com gravação em arquivo. Necessário realizar configuração das variáveis de ambiente;
 - Criar os testes primeiro, depois criar as funcionalidades para atender aos testes. Com base no objetivo da funcionalidade;
 - Como boa prática, cada teste deve desfazer todas as alterações que fez, como por exemplo deletar todos os registros criados, para que não interfira nos testes futuros;
 - Criar Factoryes para execução de rotinas repetitivas que podem aparecer em vários testes;
 - Utilizar bibliotecas para gerar dados fictícios e aleatórios, como o faker, por exemplo;