

Atividade

Testes de unidade

Questão 01

1. Realize o clone do repositório template <https://github.com/upe-campus-surubim/disciplina-verificacao-e-validacao-de-sistemas-es>, abra o projeto `sistema-de-agendamentos` e instale as suas dependências.
2. Adicione mais um teste de unidade ao caso de uso `create-appointment`, verificando que **não deve ser possível criar um agendamento em uma data já ocupada**.

Obs.: Utilize erros personalizados para as exceções e verificações.

Questão 02

Refatore o teste `create-appointment.spec` para que haja reaproveitamento de código entre os testes.

Dica: utilize a função `beforeEach`.

Questão 03

Implemente um **novo caso de uso** chamado `edit-appointment.ts`. Em seguida, crie a suíte de testes `update-appointment.spec.ts` com pelo menos 2 verificações, incluindo:

1. Não deve ser possível atualizar um agendamento **inexistente**.
2. Não deve ser possível atualizar um agendamento para uma **data já ocupada**.

Obs.: Utilize erros personalizados para as exceções e verificações.

Questão 04

Implemente um **novo caso de uso** chamado `list-employee-day-appointments.ts`, que lista os agendamentos de um **funcionário** (`employee`) em um **dia específico**. Em seguida, crie a suíte de testes com pelo menos 2 verificações, como:

1. A lista deve conter **apenas** agendamentos do funcionário solicitado, excluindo os de outros funcionários no mesmo dia.
2. A lista deve conter **apenas** agendamentos do dia solicitado, excluindo os do mesmo funcionário em outros dias.

Questão 05

Rode todos os testes de unidade do sistema e certifique-se de que **todos os casos de uso** possuam mais de **80% de cobertura**. Faça essa verificação pelo **console** e pelos **arquivos HTML** de relatório.

Questão 06

Elabore um resumo sobre **Princípios de Testes de Software** e **Identificação de Code Smells em Testes**, seguindo os requisitos abaixo:

1. **Fonte Principal:** Utilize como base a seção **8.3 Princípios e Smells** do livro *Engenharia de Software Moderna*.
 - *Link:* <https://engsoftmoderna.info/cap8.html#princ%C3%ADpios-e-smells>
2. **Pesquisa Adicional:** Enriqueça sua resposta com informações de outras fontes (livros, artigos ou documentações) para aprofundar os conceitos.
3. **Exemplos Práticos:** Para cada princípio e *code smell* abordado, apresente exemplos de código que ilustrem tanto a aplicação correta (bom padrão) quanto a incorreta (anti-padrão).