UNICURITIBA – CENTRO UNIVERSITÁRIO

HENRIQUE MAKUCH DA SILVEIRA

**VISÃO E ANÁLISE DE PROJETOS**

**DESIGN E ARQUITETURA**

CURITIBA

2023

HENRIQUE MAKUCH DA SILVEIRA

**VISÃO E ANÁLISE DE PROJETOS**

**DESIGN E ARQUITETURA**

Trabalho prático apresentado ao curso de Análise e Desenvolvimento de Software ADS como requisito parcial à obtenção de nota em Modelos Métodos e Técnicas Engenharia de Software.

Professor: Rubem Koide

CURITIBA

2023

**Testes Unitários de Mock**

O código fornecido é um exemplo de teste de unidade utilizando mocks com o framework Mockito em Java. O teste está focado em um componente chamado BookSearch, que depende de um serviço chamado BookService. O BookService é substituído por um mock para isolar o teste do comportamento real do serviço.

Vamos analisar cada parte do código e sugerir alguns testes unitários adicionais para cobrir casos específicos:

Teste testGetBook (Busca por livro existente):

Este teste verifica se o método getBook do BookSearch retorna o título correto para um livro existente. Você pode adicionar mais casos de teste para diferentes livros existentes.

Teste testGetAnotherBook (Busca por outro livro existente):

Similar ao primeiro teste, mas para um livro diferente.

Teste testGetNullBook (Busca por livro inexistente):

Verifica se o método getBook lida corretamente com a situação em que o serviço retorna um livro nulo.

Teste para Cobrir o Cenário de Mock para Livro Nulo:

Verifica se o BookSearch trata corretamente o caso em que o serviço retorna um livro nulo. Você já cobriu isso no testGetNullBook, mas você pode expandir este teste para diferentes casos de livro nulo.

Teste para Cobrir o Cenário de Mock para Livro Existente:

Verifica se o BookSearch trata corretamente o caso em que o serviço retorna um livro existente.

Teste para Cobrir o Cenário de Mock para Livro Inexistente:

Teste para um livro que não está no mock (635263). Certifique-se de que o getBook lida corretamente com essa situação.

Aqui está um exemplo adicional de teste para cobrir o caso de um livro inexistente:

java

Copy code

@Test

public void testGetNonExistentBook() {

BookSearch bs = new BookSearch(service);

String titulo = bs.getBook(9999).getTitulo();

assertEquals("NULL", titulo);

Lembre-se de que a qualidade dos testes depende da cobertura de casos e situações relevantes ao seu sistema. Certifique-se de que os testes cobrem diferentes cenários, incluindo casos de borda e situações de erro.