

# Testes de qualidade

## 1 Teste unitário – Verificar se a busca de livros retorna os resultados corretos

Objetivo: Garantir que a funcionalidade de busca retorne apenas os livros que correspondem aos critérios fornecidos pelo usuário.

Entradas de teste:

Termo de busca: "Gestão"

Termo de busca: "Machado de Assis"

Termo de busca: vazio ("")

Resultados esperados:

Apenas livros que contêm o termo “Gestão” no título ou descrição são listados.

Livros do autor “Machado de Assis” são exibidos corretamente.

Quando o campo de busca estiver vazio, deve retornar todos os livros disponíveis.

Resultado do teste: Aprovado

## Teste 2 – Testar cadastro de um novo livro no sistema

Objetivo: Validar que o sistema consegue cadastrar corretamente um novo livro, salvando todas as informações no banco de dados.

Entradas de teste:

Título: "Design Thinking"

Autor: "Tim Brown"

Resultados esperados:

O livro é salvo corretamente no banco.

Aparece na listagem de livros após o cadastro.

Resultado do teste: Aprovado

### **Teste 3 – Verificar o cálculo de multas por atraso**

Objetivo: Testar a lógica que calcula a multa diária aplicada ao usuário por atraso na devolução de um livro.

Entradas de teste:

Livro devolvido com 3, 5 e 10 dias de atraso

Valor da multa por dia: R\$ 1,50

Resultados esperados:

3 dias: multa de R\$ 4,50

5 dias: multa de R\$ 7,50

10 dias: multa de R\$ 15,00

Resultado do teste: Aprovado

### **Teste 4 – Checar se os dados de pagamento estão corretos, sem correr risco de erro no banco**

Objetivo: Garantir que os dados enviados para o pagamento via API estejam formatados corretamente e livres de inconsistências.

Entradas de teste:

Dados de pagamento simulados: nome, CPF, valor, código do livro, data de vencimento

Resultados esperados:

Os dados são validados antes do envio.

Campos obrigatórios estão preenchidos corretamente.

Nenhum dado sensível é transmitido em formato não criptografado.

Resultado do teste: Aprovado

## Testes de Integração

### Teste 1 – Testar se o app consegue registrar o pagamento/aluguel dos livros na API do banco

Objetivo: Validar a comunicação entre o aplicativo e a API do Banco Neon para registrar pagamentos ou aluguéis de livros.

Fluxo testado:

Usuário escolhe livro > Confirma pagamento > Dados enviados à API Neon

Resultados esperados:

A API retorna status 200 (sucesso).

O registro do pagamento é salvo no sistema.

Erros de rede ou falhas da API são tratados corretamente com mensagens amigáveis.

Resultado do teste: Aprovado

### Teste 2 – Validar o fluxo de empréstimo de livro com registro no banco de dados

Objetivo: Testar a integração entre a camada de UI, lógica de negócio e persistência de dados ao emprestar um livro.

Fluxo testado:

Seleção do livro > Clique em "Emprestar" > Registro de empréstimo no banco

Resultados esperados:

O livro é marcado como “emprestado” no banco de dados.

A data de devolução é corretamente gerada.

O empréstimo aparece na lista de livros ativos do usuário.

Resultado do teste: Aprovado

## GRUPO 5

**vinicius kingo 24026141**

**rodrigo cruz 24026578**

**murilo lopes 24026183**

**lucas gomes 24026562**

**lucca brandão 23024740**