

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
GOIANO CAMPUS TRINDADE
LINGUAGEM DE PROGRAMAÇÃO II

RELATÓRIO DE TESTES E RESULTADOS OBTIDOS NO SISTEMA
GERENCIADOR DE BIBLIOTECA EM JAVA

GUILHERME BARBOSA DE OLIVEIRA SOUSA

MATHEUS GUERRA MARTINS

Prof. Me. ADSON SILVA ROCHA
Trindade, 05/06/2024

1_ INTRODUÇÃO

Visando garantir o desempenho adequado de todas as funcionalidades do sistema gerenciador de biblioteca e a interação apropriada entre as camadas de apresentação, negócios e dados da aplicação desenvolvida.

Este relatório evidencia as funcionalidades do sistema, além disso, revisita os resultados obtidos a partir dos testes executados no mesmo, validando assim a capacidade desse, de criação de cadastros novos, listagens, buscas e exclusões, conforme as demandas dos usuários.

2_ CONCEITOS DE PROGRAMAÇÃO APLICADOS AO PROJETO

- **Java**: linguagem de programação orientada a objetos, robusta e amplamente utilizada para desenvolvimento de aplicações diversas.
- **Spring**: framework que simplifica o desenvolvimento Java, oferecendo recursos para injeção de dependências, gerenciamento de transações e desenvolvimento web. Implementa o padrão de programação em camada **MVC (Model-View-Controller)**, que separa a lógica de negócios (Model), a interface do usuário (View) e o fluxo de controle (Controller), tornando o código mais organizado e fácil de manter.
 - Arquivo **pom.xml** armazena todas as dependências, propriedades e configurações para um bom funcionamento no Spring.
- **Maven**: gerenciador de projetos e pacotes que automatiza a construção, testes e implantação de aplicações Java, facilitando o controle de dependências.
- **JPA (Java Persistence API)**: especificação que define uma interface padrão para mapeamento objeto-relacional, permitindo a persistência de dados em bancos de dados relacionais.
- **Hibernate**: implementação popular da JPA, que facilita o mapeamento objeto-relacional e oferece recursos adicionais para otimização de consultas e cache.
- **PostgreSQL**: sistema de gerenciamento de banco de dados relacional de código aberto, robusto e escalável, com recursos avançados como transações e consultas complexas.

- **UML (Unified Modeling Language):** linguagem de modelagem visual utilizada para representar a estrutura, o comportamento e as interações de um sistema. No contexto deste projeto, UML é usado para criar diagramas de classes, que ilustram as classes, seus atributos, métodos e relacionamentos, facilitando a compreensão da arquitetura do sistema.

3_ UML DO SISTEMA

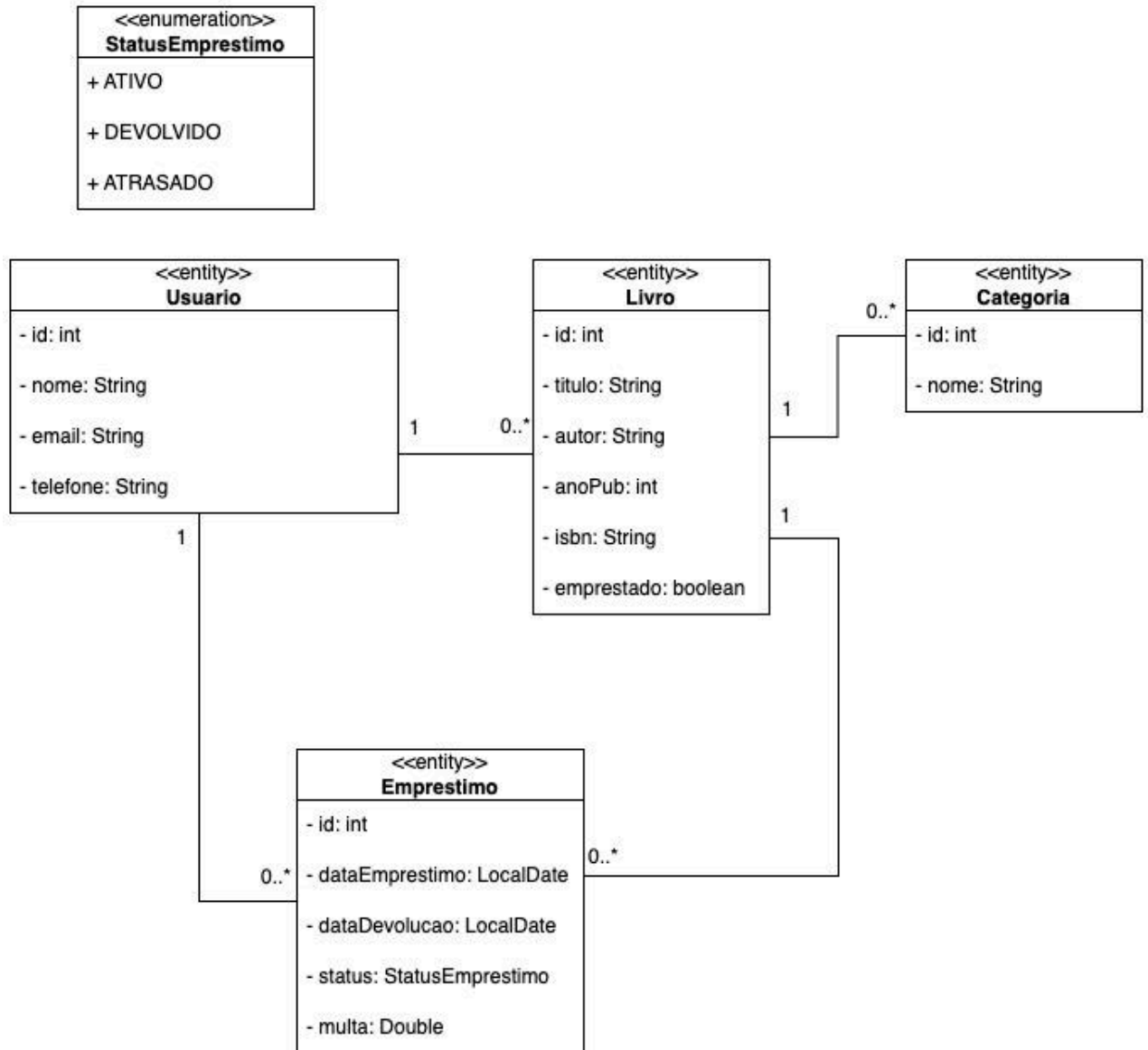
Esta seção tem o propósito de descrever de forma clara as partes que compõem o sistema, para isso descrevendo e evidenciando o diagrama de classes da aplicação desenvolvida.

3.1 – Descrição de Classes e Seus Atributos Presentes no Sistema:

- **Usuario**
 - id: int
 - nome: String
 - email: String
 - telefone: String
- **Emprestimo**
 - id: int
 - dataEmprestimo: LocalDate
 - dataDevolucao: LocalDate
 - status: Enumeration (ATIVO, DEVOLVIDO, ATRASADO)
 - multa: Double
- **Livro**
 - id: int
 - titulo: String
 - autor: String
 - anoPub: int
 - isbn: String
 - emprestado: Boolean
- **Categoria**
 - id: int

- nome: String

3.2 – Diagrama de Classes - UML:



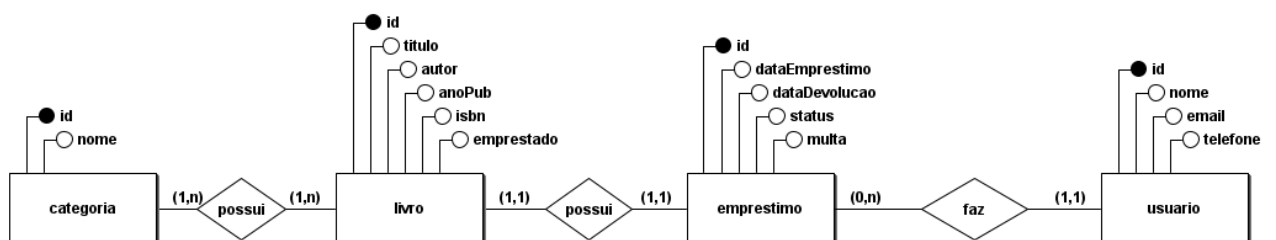
4_ Modelo Entidade-Relacionamento

Este tópico trará o foco para o MER (Modelo Entidade-Relacionamento), onde, será descrito a estruturação de entidades e atributos compostos no banco de dados do Sistema Gerenciador de Biblioteca desenvolvido.

4.1 – Descrição de Classes e Seus Atributos Presentes no MER:

- **Categoria**
 - id: Int - Serial primary key
 - nome: Varchar(255)
- **Livro**
 - id: Int - Serial primary key
 - titulo: Varchar(255)
 - autor: Varchar(255)
 - anoPub: Int
 - isbn: Varchar(13) - Unique
 - emprestado: Boolean
- **Emprestimo**
 - id: Int - Serial primary key
 - dataEmprestimo: Date
 - dataDevolucao: Date
 - status: Varchar(255) - Enun (ATIVO, DEVOLVIDO, ATRASADO)
 - multa: Double precision
- **Usuario**
 - id: Int - Serial primary key
 - nome: Varchar(255)
 - email: Varchar(255)
 - telefone: Varchar(255)

4.2 – Modelo Entidade-Relacionamento do Sistema Gerenciador de Biblioteca:



5_ TESTES REALIZADOS

Essa seção do relatório é dedicada para a listagem dos tipos de testes realizados na aplicação desenvolvida, assim como o ambiente de testagem:

5.1 – Seções do Sistema Testadas e Testes para a Validação do Sistema:

- Para as seções de usuários, livros e categorias:
 - o Fazer a listagem de todos os itens cadastrados;
 - o Criação de novos cadastros;
 - o Fazer a atualização das informações de um item cadastrado;
 - o Fazer a exclusão de um item cadastrado.
- Em especial para os livros:
 - o Fazer a buscar através do ID e apresentar as informações encontradas;
 - o Fazer a buscar através do nome completo ou parcial do livro e apresentar as informações encontradas.
- Para empréstimos:
 - o Registrar empréstimos;
 - o Registrar devoluções;
 - o Listar todos os empréstimos;

5.2 – Pré-requisitos de Ambiente para Testes do Sistema

- Windows 10 Pro;
- Java Development Kit 20.0.2;
- Framework Spring Boot 3.2.5;
- Gerenciador de Dependências Maven 4.0.0;
- Banco de Dados PostgreSQL 16.0;
- Hibernate 6.4.4;
- Java Persistence API (JPA) 3.2.5;
- Neon Serverless PostgreSQL ([Neon Tech DB](#)).

6_ TESTAGENS

6.1 – Listagem de Itens Cadastrados:

Objetivo: O intuito é verificar se o sistema pode se comunicar com o banco de dados e recuperar todas as informações armazenadas relacionadas a cada seção. Para as seções de usuário, livros, empréstimos e categorias, o sistema exibe uma lista das informações encontradas para o usuário.

Requisitos: É necessário haver cadastros realizados previamente, caso contrário, o sistema exibirá um aviso dizendo que não foram encontrados cadastros relacionados com a seção que se busca.

Passos para Executar a Função: Ao iniciar o programa o usuário deve selecionar a opção que busca (usuários, livros, empréstimos, ou, categorias), para isso, indicar o número associado à seção, em seguida selecionar a opção listar da respectiva seção, para isso, deve selecionar o número associado a essa função.

Resultados Obtidos: O sistema passou no teste, e exibiu a lista completa relacionada à respectiva seção, de todos os itens cadastrados.

6.2 – Criação de Novos Cadastros:

Objetivo: Esse teste é válido para as seções de usuário, livros, empréstimos e categorias, onde, o mesmo visa identificar se o sistema consegue adicionar novos cadastros no banco de dados, onde estes poderão ser acessados e exibidos posteriormente.

Requisitos: Para esta funcionalidade do sistema, não existem pré-requisitos.

Passos para executar a função: Iniciando o sistema o usuário deve inserir o número que representa a opção buscada (usuários, livros, empréstimos, ou categorias), em seguida selecionar a opção de adicionar ou criar para a respectiva seção. Em especial para a seção de empréstimos, é possível fazer novos cadastros com registrar, podendo registrar empréstimos ou devoluções.

Resultados obtidos: O sistema passou no teste, e adicionou o novo cadastro para a respectiva seção.

6.3 – Fazer a Atualização das Informações de Item Cadastrado:

Objetivo: A validade deste teste está associada às seções de usuário, livros e categorias, onde, o mesmo visa averiguar se o sistema consegue fazer a atualização de cadastros no banco de dados, permitindo corrigir possíveis erros no cadastro inicial, por exemplo.

Requisitos: Essa funcionalidade exige que haja cadastros na seção do sistema que se tem em vista fazer a atualização da informação, além disso, o usuário deve conhecer o ID do cadastro que quer atualizar.

Passos para Executar a Função: Com o sistema inicializado, o usuário deverá selecionar o número que indica a opção buscada (usuários, livros, ou, categorias), em seguida selecionar a opção de atualizar para a respectiva seção.

Resultados Obtidos: O teste de atualização de informações já previamente cadastradas foi executado com sucesso, portanto, o sistema passou no teste.

6.4 – Fazer a Exclusão de Item Cadastrado:

Objetivo: Esse teste visa averiguar se o sistema consegue fazer a exclusão de cadastros no banco de dados, sendo esse, validado para às seções de usuário, livros e categorias.

Requisitos: Para a execução desse teste é necessário que haja cadastros na seção do sistema que se visa fazer a exclusão da informação, mas além, o usuário deve saber o ID do cadastro que deseja excluir.

Passos para executar a função: Já previamente iniciado o sistema, o usuário terá que inserir o número indicativo da opção desejada (usuários, livros, ou, categorias), selecionar posteriormente a opção de excluir ou deletar para essa respectiva seção.

Resultados Obtidos: O teste de exclusões de informações já previamente cadastradas no sistema, foi executado com maestria, sendo assim, o sistema passou no teste.

6.5 – Fazer a Busca Através do ID e Apresentar as Informações Encontradas:

Objetivo: O sistema deve permitir que o usuário busque um livro pelo seu ID. Se o livro for encontrado, o sistema deve apresentar os resultados da busca. Se o livro não

for encontrado, o sistema deve exibir uma mensagem informando que o ID informado não corresponde a nenhum livro cadastrado. Essa funcionalidade está disponível apenas para a seção de livros.

Requisitos: Para que o teste seja realizado, é essencial que haja cadastros na seção de livros previamente, mas além, o usuário deve saber o ID do livro buscado.

Passos para Executar a Função: Iniciado o sistema, o usuário terá que selecionar o número que indica a opção de gerenciar livros, indicar posteriormente a opção de buscar livros por ID, e informar o ID do livro buscado.

Resultados Obtidos: As testagens mostraram que o sistema consegue receber o ID informado pelo usuário, buscar no banco de dados, e em seguida exibir as informações encontradas, sendo assim, o sistema passou no teste com maestria.

6.6 – Fazer a Busca Através do Nome Completo ou Parcial do Livro e Apresentar as Informações Encontradas:

Objetivo: Testar a capacidade do sistema de fazer buscas no banco de dados e apresentar os resultados encontrados, onde o usuário deve informar o nome completo ou parcial de um livro. E caso não seja encontrado nenhum resultado, o sistema apresente uma mensagem dizendo que não foram encontrados livros com o nome informado, essa funcionalidade está disponível somente para a seção de livros.

Requisitos: Para a execução desse teste é necessário ter livros cadastrados previamente no sistema.

Passos para Executar a Função: Ao iniciar o sistema, diante as opções apresentadas, o usuário terá que selecionar o número que indica a opção de gerenciar livros, em seguida, inserir número associado a opção de buscar livro por nome, e informar o nome completo ou parcial do livro buscado.

Resultados Obtidos: Com os testes da funcionalidade, foi observado que o sistema consegue receber o nome completo ou parcial do livro informado pelo usuário, e buscar no banco de dados por correspondências. Onde ao encontrar, todas as informações relacionadas são exibidas pelo sistema. Posto isso, o sistema conseguiu passar com maestria no teste.

7_ RESULTADOS

TESTAGENS	RESULTADOS
Listagem de Itens Cadastrados	Passou
Criação de Novos Cadastros	Passou
Fazer a Atualização das Informações de Item Cadastrado	Passou
Fazer a Exclusão de Item Cadastrado	Passou
Fazer a Busca Através do ID e Apresentar as Informações Encontradas	Passou
Fazer a Busca Através do Nome Completo ou Parcial do Livro e Apresentar as Informações Encontradas	Passou

8_ EXECUÇÃO DO PROJETO

Para executar o projeto com sucesso, é necessário seguir os seguintes passos rigorosamente:

- **Acessar o diretório raiz do projeto:**
 - Execute o **Prompt de Comando** do Windows ou o **Terminal** do MacOS.
 - Navegue até o diretório raiz do projeto usando o comando **cd**.
- **Executar o JAR:**
 - Execute o seguinte comando no Prompt de Comando ou no Terminal:

java -jar target/biblioteca-0.0.1-SNAPSHOT.jar

- Como resultado da ação acima, os comandos de configuração específicos para a conexão com o banco de dados, detalhados no arquivo *application.properties*, serão usados para se comunicar com o PostgreSQL NeonDB Serverless.
- **Acompanhar a Execução pelo Console Neon Tech:**
 - Enquanto o aplicativo estiver em execução, é possível acompanhar o banco de dados e as tabelas criadas acessando o link:

Serverless Postgres Neon DB

- Este link abrirá o Console do Neon Tech, no qual trata-se de uma interface *web* para gerenciar o banco de dados Serverless PostgreSQL NeonDB.
- **Realizar consultas e alterações:**
 - Como se trata de um Serverless PostgreSQL completo, é possível realizar diversas consultas usando comandos SQL ou alterar diretamente os dados nas tabelas criadas: **usuario**, **livro**, **categoria** e **emprestimo**.

9_ CONCLUSÃO

Após uma série de testes realizados, foi possível avaliar o funcionamento das funcionalidades do sistema e a comunicação entre as três camadas da aplicação. Os testes foram realizados para verificar se o usuário consegue interagir com a camada de negócios e se a mesma consegue interagir com a camada de dados do sistema, atendendo às demandas do usuário.

Os resultados dos testes demonstraram que as três camadas do sistema estão interagindo perfeitamente, e que todas as suas funcionalidades estão operacionais, conforme esperado. Isso permite que o usuário tenha uma experiência confiável e fluida ao utilizar o sistema.