

**Pró-Reitoria Acadêmica
Curso de Engenharia de Software
Trabalho de Disciplina**

Sistema de Gestão de Estoque

**Autor(es): Vitor Hugo Sena dos Reis, Pedro Eduardo DA COSTA
NASCIMENTO,**

Professor: Jefferson Salomão Rodrigues

Sumário

Resumo.....	3
Palavras-chave.....	3
1. Introdução.....	3
2. Objetivo.....	3
3. Metodologia.....	3
4. Solução Desenvolvida.....	4
4.1 Descrição do Sistema.....	4
4.2 Funcionalidades Principais.....	4
5. Tecnologias Utilizadas.....	5
6. Modelagem e Banco de Dados.....	5
6.1 Estrutura das Tabelas.....	5
6.2 Mecanismos Implementados.....	5
6.3 Controle de Acesso.....	6
6.4 Entidades e Relacionamentos.....	6
7. Conclusão.....	6
Referências Bibliográficas.....	7

Resumo

Este projeto apresenta um sistema simples de gerenciamento de estoque, desenvolvido em **Java** com **Spring Boot** e integrado a um banco **relacional (MySQL)**. O sistema permite o cadastro e controle de produtos, atualização automática de estoque e registro de auditorias. A solução prioriza clareza, organização e boas práticas.

Palavras-chave

Gestão de estoque, Spring Boot, MySQL, sistema web, auditoria

1. Introdução

A gestão de estoque é essencial para qualquer empresa que precise controlar produtos, entradas e saídas. O sistema proposto busca simplificar esse processo com uma aplicação web intuitiva e uma base de dados estruturada, garantindo integridade das informações e facilidade de consulta.

2. Objetivo

Desenvolver um sistema web funcional para **gerenciamento de produtos e movimentações de estoque**, com autenticação de usuários e integração do MySQL para armazenamento e auditoria.

3. Metodologia

O sistema foi desenvolvido com base na arquitetura **Spring Boot MVC**, utilizando o padrão **Model–View–Controller (MVC)** para manter baixo acoplamento entre as camadas.

O backend foi implementado em **Java**, com **Spring Boot**, **Spring Data JPA** e **Spring Security** para autenticação e persistência de dados.

O frontend foi construído com **HTML**, **Bootstrap** e **Thymeleaf**, gerando páginas dinâmicas integradas diretamente ao backend.

Os testes foram realizados em ambiente local, com servidor **Tomcat embutido** e banco de dados **MySQL**.

4. Solução Desenvolvida

4.1 Descrição do Sistema

O sistema permite:

- Login de usuários com diferentes níveis de acesso.
- Cadastro, edição e exclusão de produtos.
- Registro de movimentações de entrada e saída de estoque.
- Atualização automática do saldo de produtos.

4.2 Funcionalidades Principais

- **CRUD de produtos** (nome, categoria, descrição, preço, estoque).
- **Controle automático de estoque** por triggers.
- **Autenticação e autorização** baseadas em grupos de usuários (Operador, Gerente, Admin).
- **Relatórios simplificados** por meio de *views* SQL.

5. Tecnologias Utilizadas

Camada	Tecnologia	Justificativa
Frontend	HTML + Bootstrap + Thymeleaf	Simples, responsivo e integrado ao Spring.
Backend	Java + Spring Boot + Spring Data JPA	Framework robusto e padronizado para aplicações web.
Banco Relacional	MySQL	Ideal para dados estruturados e relacionamentos.

6. Modelagem e Banco de Dados

6.1 Estrutura das Tabelas

- **grupos_usuarios** → define os papéis de acesso.
- **usuarios** → informações de login e vínculo com grupo.
- **produtos** → nome, categoria, descrição, preço e estoque.
- **movimentos** → registra entradas e saídas de produtos.

6.2 Mecanismos Implementados

- **Índices:** agilizam buscas por nome e produto.
- **Triggers:** controlam atualização automática do estoque.
- **Views:** facilitam visualização e controle de permissões.

- **Procedures e Functions:** centralizam lógica de inserção e geração de IDs inteiros.

6.3 Controle de Acesso

- **Operador:** pode consultar e registrar movimentações.
- **Gerente:** pode editar e inserir produtos.
- **Admin:** tem acesso completo ao sistema e banco.

6.4 Entidades e Relacionamentos

- Cada **usuário** pertence a um **grupo de acesso**, que define suas permissões dentro do sistema.
- Os **produtos** são cadastrados com informações básicas (nome, categoria, preço, estoque atual).
- A tabela **movimentos** registra as **entradas** e **saídas** de produtos, vinculando cada operação ao usuário que a realizou.
- Essa estrutura garante integridade referencial e facilita a auditoria das ações.

7. Conclusão

O sistema cumpre seu objetivo de forma simples e eficaz: oferece controle de estoque confiável, com segurança de dados e histórico de operações. O uso conjunto de **Spring Boot** e **MySQL** resultou em uma solução moderna, modular e escalável.

Referências Bibliográficas

JOEL. **Triggers no SQL Server: teoria e prática aplicada em uma situação real**. DevMedia, 2013. Disponível em: <https://www.devmedia.com.br/triggers-no-sql-server-teoria-e-pratica-aplicada-em-uma-situacao-real/28194>. Acesso em: 13 nov. 2025.

LUCSSSLUCSSS. **Seu primeiro CRUD em Java com Spring**. Medium, 30 jan. 2024. Disponível em: <https://medium.com/@lucssslucsss/seu-primeiro-crud-em-java-com-spring-0c1dfd476a70>. Acesso em: 13 nov. 2025.

MICROSOFT. **CREATE TRIGGER (Transact-SQL)**. Microsoft Learn, 2025. Disponível em: <https://learn.microsoft.com/pt-br/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver17>. Acesso em: 13 nov. 2025.

ORACLE. **MySQL 8.0 Reference Manual**. Redwood Shores: Oracle, 2025. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/>. Acesso em: 13 nov. 2025.

ORACLE. **The Java Tutorials**. Redwood Shores: Oracle, 2025. Disponível em: <https://docs.oracle.com/javase/tutorial/>. Acesso em: 13 nov. 2025.

ROCKETSEAT. **Java: Criando CRUD com Spring Boot**. [S. l.]: Rocketseat, 2025. Disponível em: <https://www.rocketseat.com.br/blog/artigos/post/crud-java-spring-boot-guia-completo>. Acesso em: 13 nov. 2025.

SPRING. **Spring Boot Reference Documentation**. [S. l.]: Spring, 2025. Disponível em: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>. Acesso em: 13 nov. 2025.

THIAGO. **Administrando o controle de acesso no SQL Server**. DevMedia, 2014. Disponível em: <https://www.devmedia.com.br/administrando-o-controle-de-acesso-no-sql-server/31602>. Acesso em: 13 nov. 2025.

