

# Engenharia de Software II

## 5º Período – Sistemas de Informação

**Alunos:** Bruno Vinicius Rezende

**MSc.** Vânia de Oliveira Borges

Carlos Jr. Silva Lima

Guilherme Reis Pereira

## Documentação Lógica e Física de um Sistema de controle de Estoque

### Arquitetura Física

Esse sistema encontra-se atualmente com montagem local onde o banco de dados existe apenas na máquina da empresa e o sistema de controle de estoque roda localmente.

Com esta proposta, ofertamos a migração para servidores em Cloud utilizando o sistema Oracle onde o banco de dados seria totalmente em Cloud.

Onde o cliente não precisa ter um espaço nem investir em uma estrutura própria, pois usará os equipamentos fornecidos no pacote. Outro ponto positivo é a possibilidade de escalabilidade, com isso conforme a empresa cresça o pacote de serviço possa ser aumentado.

Ex. Aumento do disco de armazenamento, memória e processamento.

A comunicação entre o servidor e o cliente será via web, onde não será necessário *hardware* específico para acessar o sistema, somente algum dispositivo com *internet*.

Porém por se tratar de um sistema de controle de estoque será necessário utilizar meios de “alimentação” ao estoque.

Ex.: Leitores de códigos de barra, balanças etc.

A implantação do serviço ocorrerá de forma parcial, primeiramente será criado o banco de dados na Oracle e será dado o início à migração dos dados. De forma simultânea será migrado o sistema para web.

Após conclusão de ambos os passos será finalizada a implantação e será dado início aos treinamentos dos colaboradores interessados.

**Obs.: Essa documentação refere-se à um sistema de estoque de uma empresa fictícia.**

**Todos os dados usados foram gerados artificialmente.**

## Arquitetura Lógica: MVC (Model-View-Controller)

### • Camada Model (model)

Contém as classes que representam as entidades da aplicação e os dados manipulados:

1. **Produto.java:** representa um produto no estoque.
2. **Usuario.java:** representa os usuários do sistema.
3. **Log.java:** representa logs de ações do sistema (usados para auditoria ou relatórios).

### • Camada View (view)

Contém a interface gráfica da aplicação, construída com Swing

1. Telas como Login, Cadastro, Edição, Relatório, Início etc.
2. Todas as classes são responsáveis por interações com o usuário, exibindo e recebendo dados.

### • Camada Controller / Repository (repository)

Contém os controladores (apesar do nome Repository, atuam como DAO e controladores de lógica de negócios):

#### 1. **Persistência / Conexão:**

Conexao.java, ConexaoMySQL.java: gerenciam conexão com banco de dados.

#### 2. **Operações CRUD:**

CrudProduto.java, CrudUsuario.java: manipulam dados de produtos e usuários.

#### 3. **Funções específicas:**

BackupMySQL.java, BackupMySQLRepository.java: responsáveis por backup do banco.

ProdutoRepository.java, UsuarioRepository.java: abstraem manipulações de dados.

## Detalhamento de Classes

- **Model**

1. Produto

- **Atributos:** id, nome, quantidade, preço, fornecedor, categoria
- **Responsabilidade:** Representa um item controlado no estoque.
- **Regra associada:** Quantidade não pode ser negativa. Preço deve ser maior que zero.

2. Usuário

- **Atributos:** id, nome, login, senha, nivelAcesso
- **Responsabilidade:** Representa um usuário do sistema.
- **Regra associada:** Usuário com nivelAcesso = admin pode acessar cadastros e backups.

3. Log

- **Atributos:** id, usuário, acao, dataHora
- **Responsabilidade:** Registrar ações importantes no sistema.
- **Regra associada:** Toda modificação relevante (cadastro, edição, exclusão) gera um log.

- **Repository (Controller)**

1. CrudProduto

- **Métodos:** inserirProduto(), editarProduto(), deletarProduto(), listarProdutos()
- **Regra:** Verifica duplicidade de nome, quantidade mínima, atualiza apenas se produto existir.

2. CrudUsuario

- **Métodos:** validarLogin(), inserirUsuario(), listarUsuarios()
- **Regra:** Validação de login deve ser segura (criptografia/sanitização).

### 3. BackupMySQL

- **Método:** executarBackup()
- **Regra:** Qualquer usuário consegue fazer backup

No momento em que o usuário entra o backup se inicia e assim que ele desconecta o backup é salvo.

- **View**

#### 1. Login, Cadastro, Início, Relatório, Edição

- Cada tela interage com o usuário e repassa ações aos controllers correspondentes.
- Uso de eventos de interface para capturar ações de botões.

## Regras de Negócio

#### 1. Autenticação e Acesso

- Usuários devem se autenticar para acessar o sistema.
- Telas administrativas (cadastro de usuário, backup) são restritas ao nívelAcesso = admin.

#### 2. Cadastro de Produtos

- Não permitir:
  - Produtos com nome duplicado.
  - Quantidade < 0.
  - Preço <= 0.

#### 3. Saída de Estoque

- Deve verificar se há quantidade suficiente antes de baixar estoque.
- Registrar ação no Log.

#### 4. Edição de Produto

- Apenas campos permitidos devem ser alterados.
- Histórico de alteração pode ser registrado.

#### 5. Relatórios

- Gerados a partir de filtros (por categoria, quantidade mínima etc.).
- Exibem dados organizados e exportáveis.

#### 6. **Backup**

- Deve gerar um arquivo .sql ou .zip.
- Pode ser manual ou acionado em eventos críticos (como logout de admin).
- Deverá ser gerado automaticamente por serviços da plataforma onde está o banco de dados