

# Edital simplificado (Programa de Aquisição e Distribuição de Sementes)

## Propósito

Projetar e validar, em ambiente didático, um sistema web que suporte a gestão do Programa de Aquisição e Distribuição de Sementes, integrando **controle de estoque, logística de distribuição e rastreabilidade de lotes** com transparência pública dos resultados.

## Escopo do MVP (o que a turma deve entregar)

### 1) Módulos e funcionalidades

- **Cadastro:** espécies (ex.: milho, feijão), fornecedores, armazéns, municípios , agricultores, etc.
- **Estoque por armazém e lote:** entrada de lotes (número do lote, espécie, validade, quantidade), movimentações (entrada/saída/transferência) e saldo por armazém.
- **Logística:** criação de ordens de expedição para municípios, com data prevista e comprovante de entrega.
- **Entrega:** registro de entrega por lote (quantidade, destinatário/agricultor, comprovante).
- **Rastreabilidade:** QR Code por lote; ao ler o QR, o sistema mostra o histórico do lote (onde entrou, por onde passou, para quem foi entregue).
- **Transparência:** painel público (página sem login) com números agregados: *total distribuído por espécie, por município e por período, etc.*
- **Relatórios:** por período/lote/espécie, divergências (ex.: saldo negativo bloqueado) e produtividade (entregas/dia).
- **LGPD (transparência):** Política de Privacidade e avisos claros no sistema.

*Observação: mantém-se o foco do edital em estoque, logística, rastreabilidade e transparência, mas sem integrações governamentais ou tecnologia avançada opcional (ex.: blockchain).*

### 2) Perfis de usuários (podem existir outros)

- **Gestor (admin):** configura cadastros e acompanha indicadores.
- **Operador de armazém:** registra entradas e expedições.
- **Agente de distribuição:** registra entregas.
- **Cidadão:** acessa o painel público (somente leitura).
- **Associações, cooperativas, empreendedores familiares rurais, agricultores de base familiar, organizações não governamentais:** quem solicita.

## Requisitos

- Web responsivo (desktop e celular).
- API REST para operações (lotes, expedições, entregas, etc.).
- Front-end em React/Next.js consumindo a API; back-end em Java (Spring Boot).
- Banco de dados relacional com integridade referencial, transações para expedições/entregas, views, procedures e triggers (ex.: impedir saldo negativo; auditar movimentações).

*Segurança mínima esperada: autenticação própria com senhas hasheadas e perfis de acesso (sem integrações externas).*

## Entregáveis (mínimos)

1. Backlog com histórias de usuário e protótipo de telas.
2. Modelo de dados (Diagrama Entidade-Relacionamento) e scripts DDL/DML; views, procedures, triggers e uso de transações.
3. API REST com documentação Swagger e coleção Postman/Insomnia.
4. Aplicação web (front-end e back-end) contemplando os requisitos definidos nesse documento.

## Stack Tecnológica

- **Front-end:** HTML, CSS, Typescript e React/Next.js.
- **Back-end:** Spring Boot (Java).
- **Banco de Dados:** MySQL.
- **Ferramentas:** VS Code/Eclipse, Git/GitHub, Postman/Insomnia, MySQL Workbench, Git.