



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS GUARABIRA DIRETORIA DE DESENVOLVIMENTO DE ENSINO
COORDENAÇÃO DO CURSO DE TECNOLOGIA EM SISTEMAS PARA
INTERNET**

RAFAEL MOURA DE SOUZA

RELATÓRIO TÉCNICO

**Projeto Integrador em Sistemas para Internet (PISI): SGG (Sistema Gerenciador
de Granjas)**

**GUARABIRA– PB
2026**

RAFAEL MOURA DE SOUZA

Projeto Integrador em Sistemas para Internet (PISI): SGG (Sistema Gerenciador de granjas)

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Sistemas para Internet, do Instituto Federal da Paraíba – Campus Guarabira, em cumprimento às exigências parciais para a obtenção do título Tecnólogo em Sistemas para Internet.

Orientador: Prof. Dr. Rodrigo Leone Alves
Coorientador: Prof. Me. Lucas Vieira de Souza

**GUARABIRA– PB
2026**

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IFPB - GUARABIRA

S729r	Souza, Rafael Moura de Relatório técnico: Projeto Integrador em Sistemas para Internet (PISI): SGG (Sistema Gerenciador de Granjas) / Rafael Moura de Souza.- Guarabira, 2026. 36f.; il.; color. Trabalho de Conclusão de Curso (Tecnólogo em Sistemas para Internet). – Instituto Federal da Paraíba, Campus Guarabira. 2026. “Orientação: Prof. Dr. Rodrigo Leone Alves Coorientação: Prof. Me. Lucas Vieira de Souza” Referências. 1.Sistema para internet 2. Gestão de granjas. 3. Sistemas de Informação. 4. Agropecuária. 5.Gestão Rural. I. Título. CDU 004.4(0.067)
-------	---

Elaborada por Ana Carine da Costa Gonçalves - CRB 000676



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS GUARABIRA

ATA 15/2026 - CCSTSI/DDE/DG/GB/REITORIA/IFPB

ATA DE APRESENTAÇÃO E DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO
CCS de Tecnologia em Sistemas para Internet

Aos 06 de agosto de 2026, às 10:00, no Laboratório de Informática, reuniram-se os membros da banca avaliadora, Rodrigo Leone Alves (Orientador), Lucas Vieira de Souza (Coorientador), Tannissa Luanna Cardoso de Araujo (Examinador Interno), Daniel Marques Targino Guimarães (Examinador Interno), para avaliarem a apresentação do *Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet* (Relatório Final do Projeto Integrador em Sistemas para Internet - PISI) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *campus* Guarabira, desenvolvido pela aluno(a) **Rafael Moura de Souza**, matrícula de nº 202323810006, intitulado "*SGG (Sistema Gerenciador de Granjas)*", protocolado para apresentação de acordo com os requisitos expostos no Projeto Pedagógico de Curso de Tecnologia em Sistemas para Internet. Após a apresentação, a banca apresentou, por unanimidade, pareceres a favor da aprovação do trabalho. Desta forma, o Trabalho de Conclusão de Curso foi **aprovado** e definiu-se a **nota final 90 (Noventa)**.

Nada mais havendo a tratar, às 16:45, encerraram-se os trabalhos, determinando a lavratura desta ata, que, após lida e considerada conforme, será assinada pelos presentes. Eu, Rodrigo Leone Alves, lavrei a presente ata.

Guarabira/PB, em 10 de agosto de 2026.

Documento assinado eletronicamente por:

- **Rodrigo Leone Alves, PROFESSOR ENS BASICO TECN TECNOLÓGICO**, em 10/08/2026 11:57:34.
- **Tannissa Luanna Cardoso de Araujo, PEDAGOGO-AREA**, em 10/08/2026 12:47:55.
- **Lucas Vieira de Souza, PROFESSOR ENS BASICO TECN TECNOLÓGICO**, em 10/08/2026 14:21:45.
- **Daniel Marques Targino Guimarães, PROF ENS BAS TEC TECNOLÓGICO-SUBSTITUTO**, em 10/08/2026 22:46:06.

Este documento foi emitido pelo SUAP em 10/08/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 919896
Verificador: 78e23863d9
Código de Autenticação:



AGRADECIMENTOS

Por Rafael Moura de Souza

Agradeço aos meus avós que tanto se esforçaram para me criar, que me apoiam com tanto esmero e me fortalecem para que eu possa alcançar meus objetivos. Aos meus colegas que dividiram estes anos, experiências e desafios.

À instituição e aos professores, por todo conhecimento compartilhado e pela dedicação em nossa formação, fundamentais para a minha trajetória acadêmica e profissional.

*"A ciência é mais que um corpo de
conhecimento, é uma forma de
pensar, uma forma cética de
interrogar o universo, com pleno
conhecimento da falibilidade
humana."*

- Carl Sagan

RESUMO

O Projeto Integrador em Sistemas para Internet tem como objetivo desenvolver a capacidade de aplicação integrada dos conceitos e teorias adquiridos ao longo do curso. O presente Trabalho de Conclusão de Curso apresenta a documentação do sistema **SGG – Sistema Gerenciador de Granjas**, implementado na forma de software, com a finalidade de permitir que pequenas e médias granjas organizem, centralizem e rastreiem seus processos de gerenciamento. A metodologia adotada baseou-se em uma pesquisa aplicada, utilizando práticas de desenvolvimento de software com levantamento de requisitos funcionais e não funcionais e implementação incremental da solução. Para o desenvolvimento do sistema, foram utilizadas as tecnologias Python com o framework Flask no backend, React no frontend e PostgreSQL para o gerenciamento do banco de dados, fornecendo funcionalidades para o controle de lotes de frangos, estoque, produção, vendas, despesas, receitas, funcionários e demais informações relacionadas à administração da granja. Dessa forma, o sistema busca contribuir para a organização das informações, a redução de processos manuais e o apoio à tomada de decisões pelos produtores rurais.

Palavras-chave: Gestão de Granjas; Sistemas de Informação; Agropecuária; Gestão Rural.

ABSTRACT

The Integrated Project in Internet Systems aims to develop the ability to apply, in an integrated manner, the concepts and theories acquired throughout the undergraduate program. This Final Course Project presents the documentation of the **SGG – Farm Management System**, implemented as a software solution designed to enable small and medium-sized poultry farms to organize, centralize, and monitor their management processes. The adopted methodology was based on applied research, using software development practices, including the definition of functional and non-functional requirements and the incremental implementation of the solution. The system was developed using Python with the Flask framework for the backend, React for the frontend, and PostgreSQL as the database management system, providing features for managing broiler flocks, inventory, production, sales, expenses, revenues, employees, and other information related to farm administration. Therefore, the system aims to improve information organization, reduce manual processes, and support decision-making by poultry farmers.

Keywords: Farm Management; Information Systems; Agriculture; Rural Management.

LISTA DE FIGURAS

Figura 1 - Arquitetura do Sistema.....	24
Figura 2 - Modelo lógico da base de dados	26
Figura 3 - Continuação do modelo lógico.....	27
Figura 4 – Diagrama de Caso de Uso	28
Figura 5 – Tela de Login	30
Figura 6 – Tela de Granjas.....	30
Figura 7 – Tela Associados.....	31
Figura 8 - Tela Lotes de Frango	31
Figura 9 - Tela de Mortalidade da Granja.....	32
Figura 10 – Tela de Produtos/Estoque.....	32
Figura 11 – Tela de Ração.....	33
Figura 12 – Tela de Finanças da Granja	33
Figura 13 – Tela de Vendas	34
Figura 14 – Tela de Receitas	34
Figura 15 – Tela de Despesas	35
Figura 16 – Tela de Configurações da Granja.....	35

LISTA DE TABELAS

Tabela 1 - Cronograma de Atividade do TCC	17
Tabela 2 - Resultados dos Testes de Carga	37

LISTA DE ABREVIATURA E SIGLAS

API – *Application Programming Interface* (Interface de Programação de Aplicação)

CSS3 – *Cascading Style Sheets 3*

ERPs – *Enterprise Resource Planning* (Planejamento de Recursos Empresariais)

HTML5 – *HyperText Markup Language 5*

IBGE – Instituto Brasileiro de Geografia e Estatística

IFPB – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

ORM – *Object-Relational Mapping* (Mapeamento Objeto-Relacional)

PISI – Projeto Integrador em Sistemas para Internet

RAM – *Random Access Memory* (Memória de Acesso Aleatório)

REST – *Representational State Transfer* (Transferência de Estado Representacional)

RF – Requisito Funcional

RNF – Requisito Não Funcional

SGBDR – Sistema Gerenciador de Banco de Dados Relacional

SGG – Sistema Gerenciador de Granjas

SI – Sistemas de Informação

TCC – Trabalho de Conclusão de Curso

UN – Unidade (medida de quantidade)

VUs – *Virtual Users* (Usuários Virtuais)

P90 – 90º percentil

P95 – 95º percentil

2026
SUMÁRIO

1 INTRODUÇÃO	13
1.1 Descrição do problema	13
1.2 Descrição do Software a ser Desenvolvido.....	14
1.3 Objetivos Gerais.....	14
2. REFERENCIAL TEÓRICO	15
2.1 A importância da digitalização e dos sistemas de informação na gestão	15
2.2 Os desafios tecnológicos enfrentados por pequenos e médios produtores avícolas	15
2.3 A integração das tecnologias da informação no setor rural.....	15
3 METODOLOGIA	17
3.1 Abordagem Metodológica	17
3.2 Cronograma.....	17
3.3 Procedimentos Técnicos	17
3.4 REQUISITOS FUNCIONAIS	18
3.5 REQUISITOS NÃO FUNCIONAIS	22
4 RESULTADOS E DISCURSÃO.....	24
4.1 Arquitetura do Sistema	24
4.2 Modelagem do Sistema	26
4.3 DIAGRAMAS DE CASO DE USO	28
4.4 Telas do Sistema	30
4.5 Testes de Carga.....	36
5 CONCLUSÃO	38
REFERÊNCIAS	39

1 INTRODUÇÃO

Com o constante crescimento do setor agropecuário e a consequente ampliação do número de granjas na região, observa-se também um aumento na complexidade administrativa desses empreendimentos, especialmente em granjas de médio e grande porte. A gestão eficiente dessas unidades exige controle rigoroso de informações, acompanhamento de processos operacionais e tomada de decisões baseada em dados confiáveis (LAUDON; LAUDON, 2021).

Diante desse cenário, torna-se evidente a necessidade de ferramentas tecnológicas que auxiliem na organização, no controle e na otimização das atividades relacionadas à administração de granjas. Sistemas informatizados podem reduzir falhas humanas, melhorar o gerenciamento de recursos e facilitar o acompanhamento de indicadores relevantes para o funcionamento do negócio (PRESSMAN; MAXIM, 2022).

Assim, propõe-se o desenvolvimento do SGG – Sistema Gerenciador de Granjas, uma solução voltada para centralizar informações e automatizar processos essenciais, contribuindo para uma gestão mais eficiente e estruturada. O sistema busca oferecer suporte ao administrador e à equipe operacional, permitindo maior controle sobre atividades, registros e rotinas administrativas.

O presente projeto tem como objetivo desenvolver uma aplicação de fácil utilização, capaz de atender às demandas de gerenciamento de uma ou múltiplas granjas, garantindo uma interface intuitiva, boa performance e baixa curva de aprendizado. Além disso, o sistema visa contribuir para a modernização do setor, promovendo maior eficiência operacional, melhor organização dos dados e apoio à tomada de decisão (SOMMERVILLE, 2019).

1.1 Descrição do problema

A gestão de granjas, especialmente as de pequeno e médio porte, ainda enfrenta dificuldades devido à falta de soluções tecnológicas específicas para esse tipo de empreendimento. Em muitos casos, o controle das atividades administrativas e operacionais é realizado manualmente ou por meio de ferramentas genéricas, o que pode ocasionar falhas no registro de informações, desorganização de dados, retrabalho e perda de eficiência no acompanhamento das rotinas diárias (EMBRAPA, 2023).

Além disso, a ausência de um sistema centralizado dificulta a visualização geral do funcionamento da granja, comprometendo a análise de dados e tornando a tomada de decisões mais lenta e menos precisa. Esse cenário evidencia a necessidade de uma solução que permita registrar e organizar informações de forma estruturada, garantindo maior controle e confiabilidade (LAUDON; LAUDON, 2021).

Diante desse problema, surge a proposta do SGG – Sistema Gerenciador de Granjas, com a finalidade de centralizar recursos e integrar o acompanhamento das atividades operacionais e administrativas. Dessa forma, busca-se facilitar o gerenciamento diário, melhorar o controle operacional e oferecer suporte à modernização do processo de gestão, mesmo em granjas de menor escala.

1.2 Descrição do Software a ser Desenvolvido

Será desenvolvido um sistema web, disponibilizado por meio de navegador, com o objetivo de integrar e centralizar os dados coletados diariamente nos processos internos de gerenciamento de uma granja. A aplicação permitirá o armazenamento e tratamento dessas informações, oferecendo ao usuário uma visualização organizada e consolidada dos dados registrados.

O sistema será projetado para atender equipes de pequeno e médio porte, possibilitando que diferentes colaboradores realizem o lançamento de informações relacionadas às suas atividades específicas. Dessa forma, será possível manter um histórico rastreável das operações realizadas, garantindo maior controle e transparência no gerenciamento.

Além disso, o software contará com a geração automática de relatórios, reunindo dados agregados e apresentando informações resumidas e detalhadas, de modo a auxiliar a administração na análise de desempenho e na tomada de decisões.

1.3 Objetivos Gerais

Desenvolver um sistema web para gerenciamento de pequenas e médias granjas, permitindo o controle das atividades operacionais, geração de relatórios e apoio à tomada de decisão.

1.4 Objetivos Específicos

- Registrar e gerenciar funcionários;
- Registrar e gerenciar aviários;
- Permitir gerenciar rações e suprimentos;
- Rastrear atividades de vendas e compras;
- Gerar relatórios automatizados;
- Gerar alertas automatizados de quantidade de ração restante e atividades que precisem de atenção;
- Gerar relatórios financeiros básicos;
- Visualização produtos gerados, como: quantidade de ovos, quilos de carne, quilos de esterco etc.

2. REFERENCIAL TEÓRICO

2.1 A importância da digitalização e dos sistemas de informação na gestão

A transformação digital alterou a forma como as organizações operam, tornando a gestão da informação um ativo estratégico. Segundo Laudon e Laudon (2014), os Sistemas de Informação (SI) são fundamentais para o suporte à tomada de decisão, permitindo que gestores analisem dados em tempo real. A implementação desses sistemas minimiza falhas humanas e retrabalhos, pois centraliza o fluxo de dados em uma única base de conhecimento (O'BRIEN; MARAKAS, 2013).

Além da organização dos dados, a digitalização promove ganhos significativos em eficiência operacional e rastreabilidade dos processos. No contexto agropecuário, essa tecnologia permite o monitoramento preciso da produção, convertendo dados brutos em inteligência para o negócio (PRESSMAN, 2016).

2.2 Os desafios tecnológicos enfrentados por pequenos e médios produtores avícolas

A avicultura desempenha um papel vital na economia nacional, com destaque para a força da agricultura familiar e de pequenos empreendimentos (IBGE, 2019). Contudo, observa-se um hiato tecnológico nesses setores. Enquanto grandes empresas utilizam sistemas integrados, o pequeno produtor frequentemente recorre a controles informais, como papéis e planilhas desconexas, o que limita sua capacidade de gestão.

Conforme dados do Sebrae e da Embrapa, a principal barreira para a digitalização desses produtores não é a falta de interesse, mas o alto custo e a complexidade dos ERPs agrícolas convencionais. Existe, portanto, uma carência de ferramentas acessíveis, intuitivas e de baixo custo que atendam especificamente às demandas de escala e operação desses pequenos negócios (SEBRAE, 2023; EMBRAPA, 2023).

2.3 A integração das tecnologias da informação no setor rural

A convergência das tecnologias da informação no campo marca a era da "Agricultura 4.0" (ALCANTARA et al., 2021). A integração de sistemas web com a computação em nuvem permite que o produtor acesse dados críticos de qualquer local, superando as limitações geográficas tradicionais.

Ao consolidar processos em um sistema de gestão integrada, a rastreabilidade deixa de ser um custo burocrático e torna-se um diferencial competitivo, garantindo a qualidade e a segurança do produto final. A automatização de processos manuais, sustentada por bancos de dados robustos, oferece ao pequeno produtor a mesma

clareza de indicadores utilizada pelas grandes agroindústrias, fundamentando decisões mais seguras, menos intuitivas e baseadas em dados concretos.

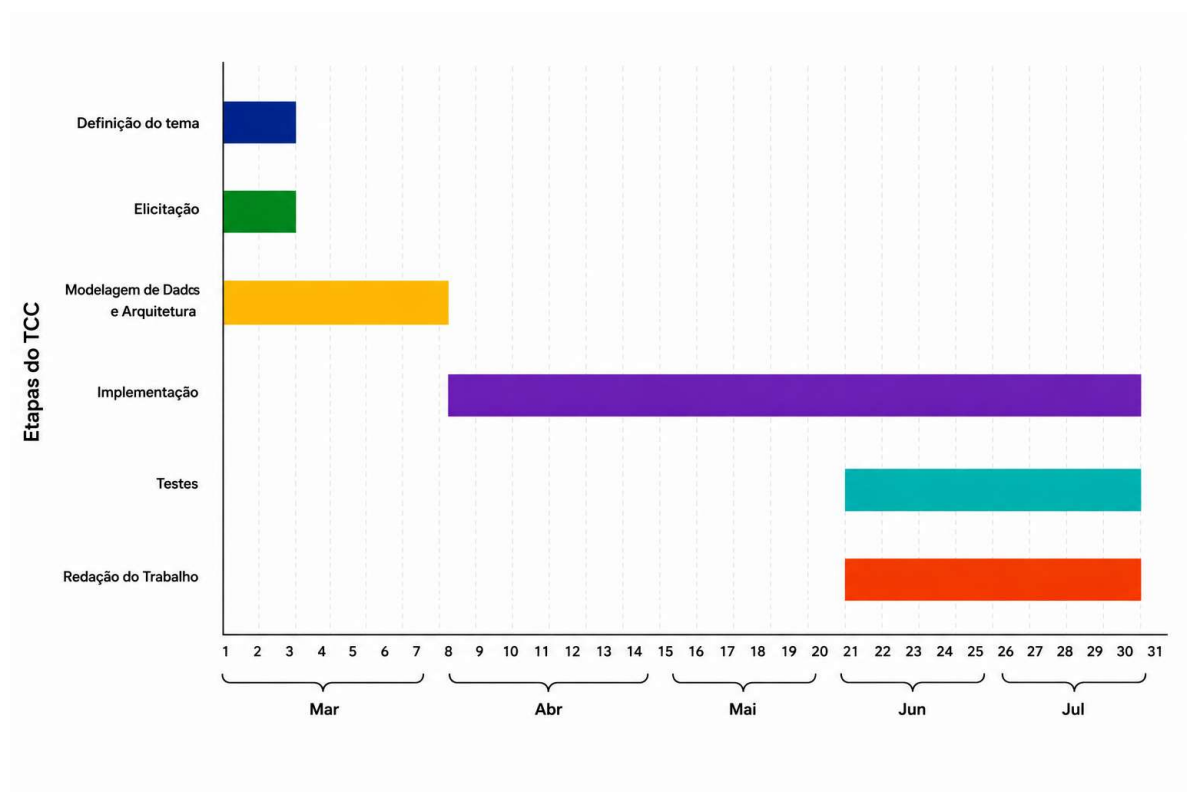
3 METODOLOGIA

3.1 Abordagem Metodológica

Este trabalho caracteriza-se como uma pesquisa aplicada, de abordagem qualitativa, com objetivos exploratórios e descritivos. Quanto aos procedimentos técnicos, fundamenta-se em pesquisa bibliográfica e no desenvolvimento tecnológico de um software para gerenciamento de granjas. O desenvolvimento da solução compreendeu as etapas de levantamento de requisitos, modelagem, implementação e testes, visando atender às necessidades de organização e controle das atividades de pequenas e médias granjas.

3.2 Cronograma

Tabela 1 - Cronograma de Atividade do TCC



Fonte: Elaborado pelo autor

3.3 Procedimentos Técnicos

O ciclo de vida do desenvolvimento do software seguiu as seguintes etapas técnicas:

- **Definição de Tema:** Busca de entender as problemáticas a serem abordadas e criar pensamento crítico ao assunto.

- **Metodologia e Planejamento:** Pesquisa e organização dos recursos tecnológicos a serem usadas no projeto; Elaboração das etapas e tempo necessário para finalizá-las.
- **Modelagem de Dados e Arquitetura:** Definição dos dados a serem tratados no banco de dados e estrutura lógica relacional; Planejamento das disposições necessárias para criação do sistema e elaboração da redação do projeto.
- **Implementação:** Dividido em dois escopos – backend, implementando uma API (Application Programming Interface) em Python no microframework Flask, módulos/bibliotecas para abstrações para banco de dados.
- **Testes:** Verificação das regras de negócio e recursos disponíveis do sistema.

3.4 REQUISITOS FUNCIONAIS

3.4.1 Conta Master

RF001 – Criação de Conta Master

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Ao criar uma conta o usuário pode criar uma granja, tornando-se uma conta master das granjas criadas.

3.4.2 Gestão de Granjas

RF002 – Cadastrar Granja

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir cadastrar granjas associadas a uma conta administradora.

RF003 – Atualizar Granja

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir atualizar os dados das granjas cadastradas.

RF004 – Desativar Granja

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

O sistema deve permitir desativar granjas mantendo seu histórico operacional.

RF005 – Consultar Granja

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

O sistema deve permitir consultar granjas cadastradas.

3.4.3 Gestão de Equipe

RF006 – Convidar Outro Usuário

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir o envio de convite para criar relação entre os usuários.

RF007 – Relação de Granjas

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir realizar associação de granjas do usuário master.

RF008 – Desassociar Funcionário

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

O sistema deve permitir desassociar funcionários da granja retirando a granjas associadas do registro.

RF009 – Associar Cargo ao Funcionário

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir associar cargos aos funcionários cadastrados.

3.4.6 Gestão de Clientes

RF010 – Cadastrar Cliente

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir cadastrar clientes para realização de vendas.

RF011 – Atualizar Cliente

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir atualizar os dados dos clientes cadastrados.

RF012 – Deletar Cliente

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

O sistema deve permitir desativar clientes mantendo seu histórico de compras.

3.4.7 Gestão de Produtos

RF013 – Cadastrar Produto

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir cadastrar produtos produzidos pela granja.

RF014 – Atualizar Produto

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir atualizar os dados dos produtos cadastrados.

RF015 – Deletar Produto

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir deletar produtos.

RF016 – Consultar Produto

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir consultar produtos cadastrados.

3.4.8 Gestão de Aviários e Lotes

RF017 – Cadastrar Aviário

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir cadastrar aviários vinculados a uma granja.

RF018 – Atualizar Aviário

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir atualizar os dados dos aviários cadastrados.

RF019 – Consultar Aviário

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir consultar aviários cadastrados.

RF020– Cadastrar Lote de Frango

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir cadastrar lotes de frango vinculados a um aviário.

RF021– Atualizar Lote de Frango

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir atualizar os dados dos lotes cadastrados.

RF022 – Encerrar Lote de Frango

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir encerrar lotes mantendo seu histórico produtivo.

RF023 – Consultar Lote de Frango

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir consultar lotes cadastrados.

3.4.9 Gestão de Produção

RF024 – Registrar Produção

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir registrar a produção diária de produtos gerados por um lote.

RF025 – Atualizar Estoque por Produção

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve atualizar automaticamente o estoque após o registro de produção.

RF026 – Consultar Histórico de Produção

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir consultar registros de produção realizados.

3.4.10 Gestão de Mortalidade

RF027 – Registrar Mortalidade

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir registrar a quantidade de aves mortas em um lote.

RF028 – Atualizar Quantidade de Aves

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve atualizar automaticamente a quantidade de aves do lote após registros de mortalidade.

RF029 – Consultar Histórico de Mortalidade

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

O sistema deve permitir consultar registros de mortalidade dos lotes.

3.4.11 Gestão de Estoque

RF030 – Consultar Estoque

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir consultar o saldo atual dos produtos armazenados.

RF031 – Registrar Ajuste de Estoque

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir registrar ajustes de estoque quando necessário.

RF032 – Consultar Movimentações de Estoque

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir consultar o histórico de movimentações de estoque.

3.4.12 Gestão de Vendas

RF033 – Registrar Venda

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir registrar vendas para clientes cadastrados.

RF034 – Atualizar Status de Venda

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir alterar o status das vendas durante seu ciclo operacional.

RF035 – Consultar Venda

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

O sistema deve permitir consultar vendas registradas.

RF036 – Consultar Histórico de Vendas

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

O sistema deve permitir consultar o histórico de vendas realizadas.

3.4.13 Gestão Financeira

RF037 – Registrar Despesa

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir registrar despesas relacionadas à operação da granja.

RF038 – Consultar Despesas

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir consultar despesas registradas.

3.5 REQUISITOS NÃO FUNCIONAIS

3.5.1 Segurança

RNF001 – Autenticação de Usuários

O sistema deve exigir autenticação para acesso às funcionalidades protegidas.

RNF002 – Controle de Acesso

O sistema deve restringir o acesso às funcionalidades de acordo com o cargo do usuário.

RNF003 – Armazenamento Seguro de Senhas

O sistema deve armazenar senhas utilizando algoritmos de hash seguros.

3.5.2 Desempenho

RNF004 – Tempo de Resposta

O sistema deve responder às requisições do usuário em até **1 segundo** para operações comuns (cadastros, consultas e edições), considerando condições normais de uso. Esse limite foi adotado por manter o fluxo de interação do usuário sem interrupções perceptíveis, conforme proposto por Nielsen (1993), como parte dos limites clássicos de resposta.

3.5.3 Disponibilidade

RNF005 – Persistência dos Dados

O sistema deve manter os dados armazenados em banco de dados relacional.

3.5.4 Usabilidade

RNF006 – Interface Responsiva

O sistema deve possuir interface adaptável a diferentes tamanhos de tela.

RNF007 – Padronização da Interface

O sistema deve manter consistência visual entre suas funcionalidades.

3.5.5 Confiabilidade

RNF008 – Integridade dos Dados

O sistema deve garantir a integridade dos dados armazenados durante operações de cadastro, atualização e consulta.

RNF009 – Consistência do Estoque

O sistema deve manter a consistência das informações de estoque após registros de produção, vendas e ajustes.

3.5.6 Tecnológicos

RNF010 – API REST

O sistema deve disponibilizar suas funcionalidades por meio de uma API REST.

RNF011 – Compatibilidade com Navegadores Modernos

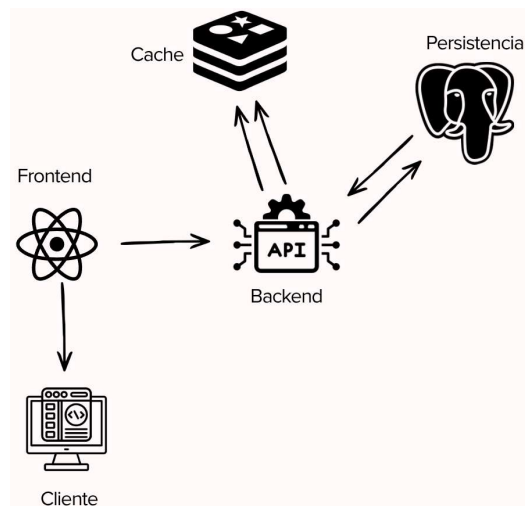
O sistema deve funcionar nos principais navegadores modernos compatíveis com HTML5, CSS3 e JavaScript.

4 RESULTADOS E DISCURSÃO

4.1 Arquitetura do Sistema

O sistema possui uma arquitetura em camadas, seguindo o modelo cliente-servidor, separados em frontend, backend, persistência, cache e busca especializada.

Figura 1 - Arquitetura do Sistema



Fonte: Elaborado pelo autor.

Frontend: Desenvolvido em React (REACT, 2026), utilizando a biblioteca Axios (AXIOS, 2026) para realizar requisições HTTP (Hypertext Transfer Protocol), permitindo que os usuários acessem o sistema por meio de um navegador web. A aplicação é disponibilizada por um servidor web Nginx (NGINX, 2026), executado em um contêiner Docker (DOCKER INC., 2026) isolado.

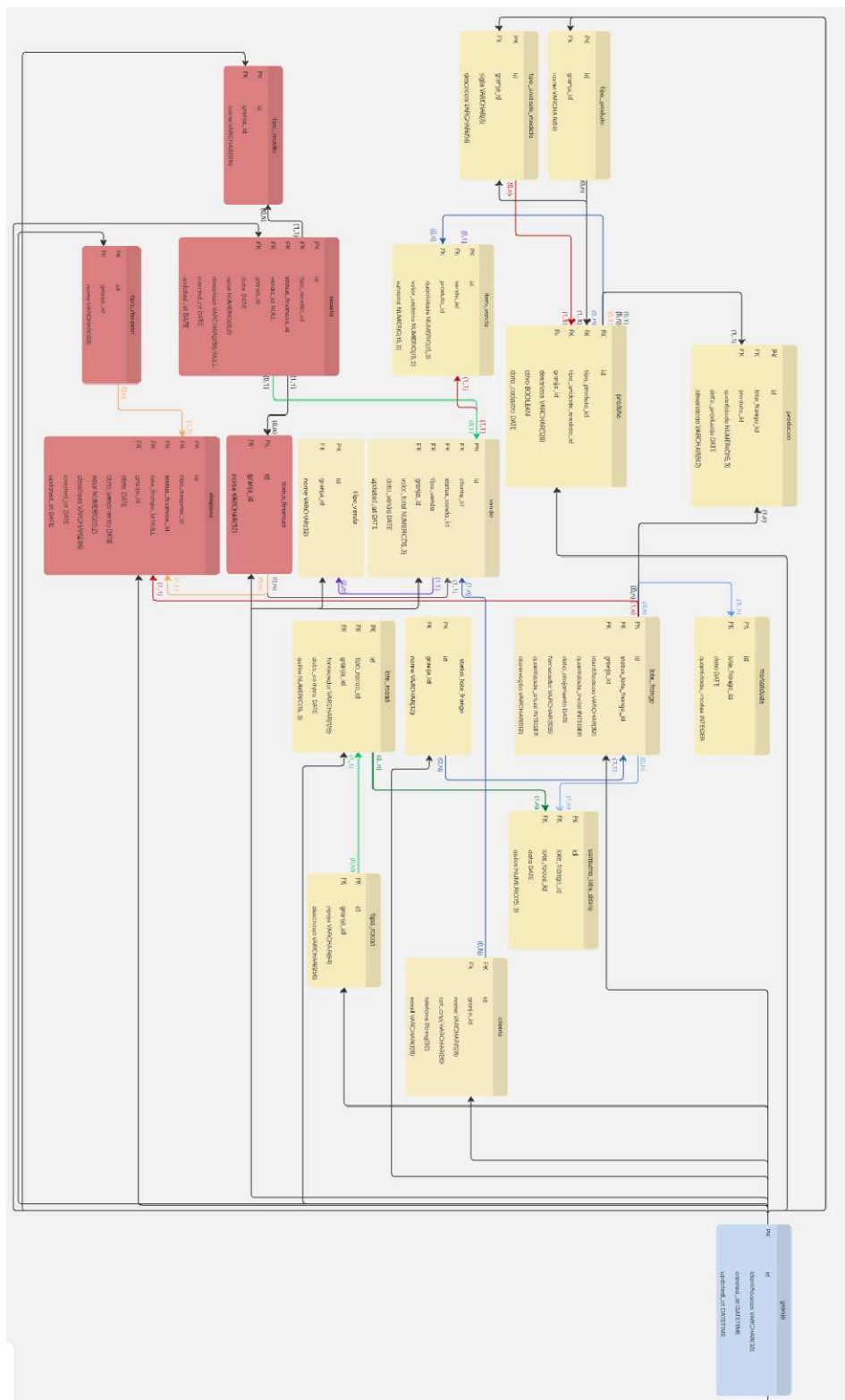
Backend: Desenvolvido em Python utilizando o microframework Flask (PALLETS, 2026), responsável pela implementação das regras de negócio e pela comunicação com o banco de dados. Para o mapeamento objeto-relacional (ORM - Object-Relational Mapping), foi utilizada a extensão Flask-SQLAlchemy (PALLETS, 2026). O controle de versões do esquema do banco de dados é realizado por meio do Flask-Migrate (GRINBERG, 2026). Além disso, emprega-se o Flask-Caching (FLASK-CACHING, 2026) para abstrair a camada de cache, permitindo flexibilidade na utilização de diferentes soluções. Para este projeto, foi adotado o Redis (REDIS, 2026) como sistema de gerenciamento de cache.

PostgreSQL: Sistema gerenciador de banco de dados relacional (SGBDR) escolhido para armazenar os dados persistentes da aplicação, oferecendo confiabilidade, integridade e suporte a transações (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2026).

Redis: Sistema de gerenciamento de banco de dados NoSQL utilizado como solução de cache, proporcionando baixa latência e alto desempenho por armazenar os dados diretamente na memória principal (RAM) (REDIS, 2026).

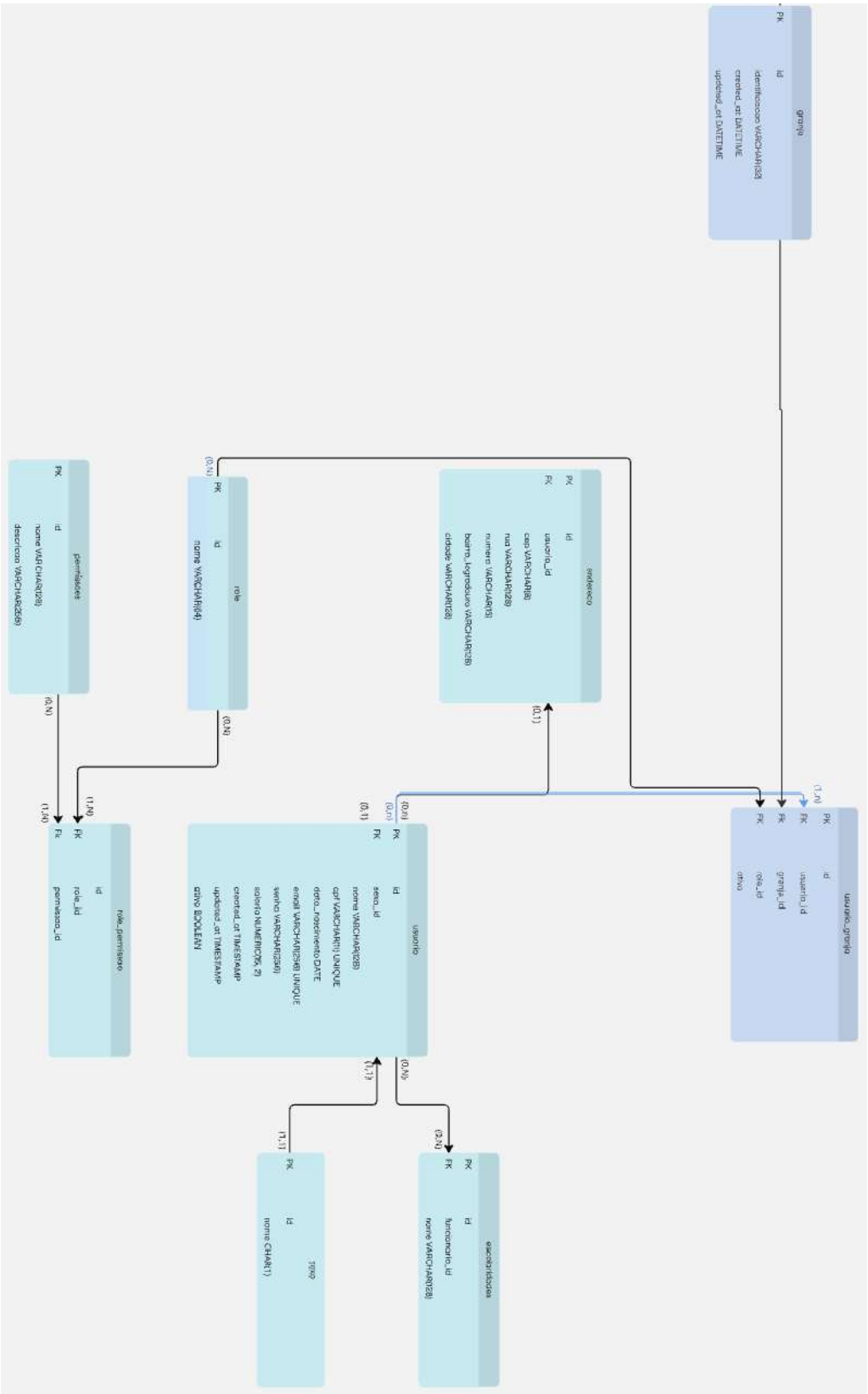
4.2 Modelagem do Sistema

Figura 2 - Modelo lógico da base de dados



Fonte: Elaborado pelo autor.

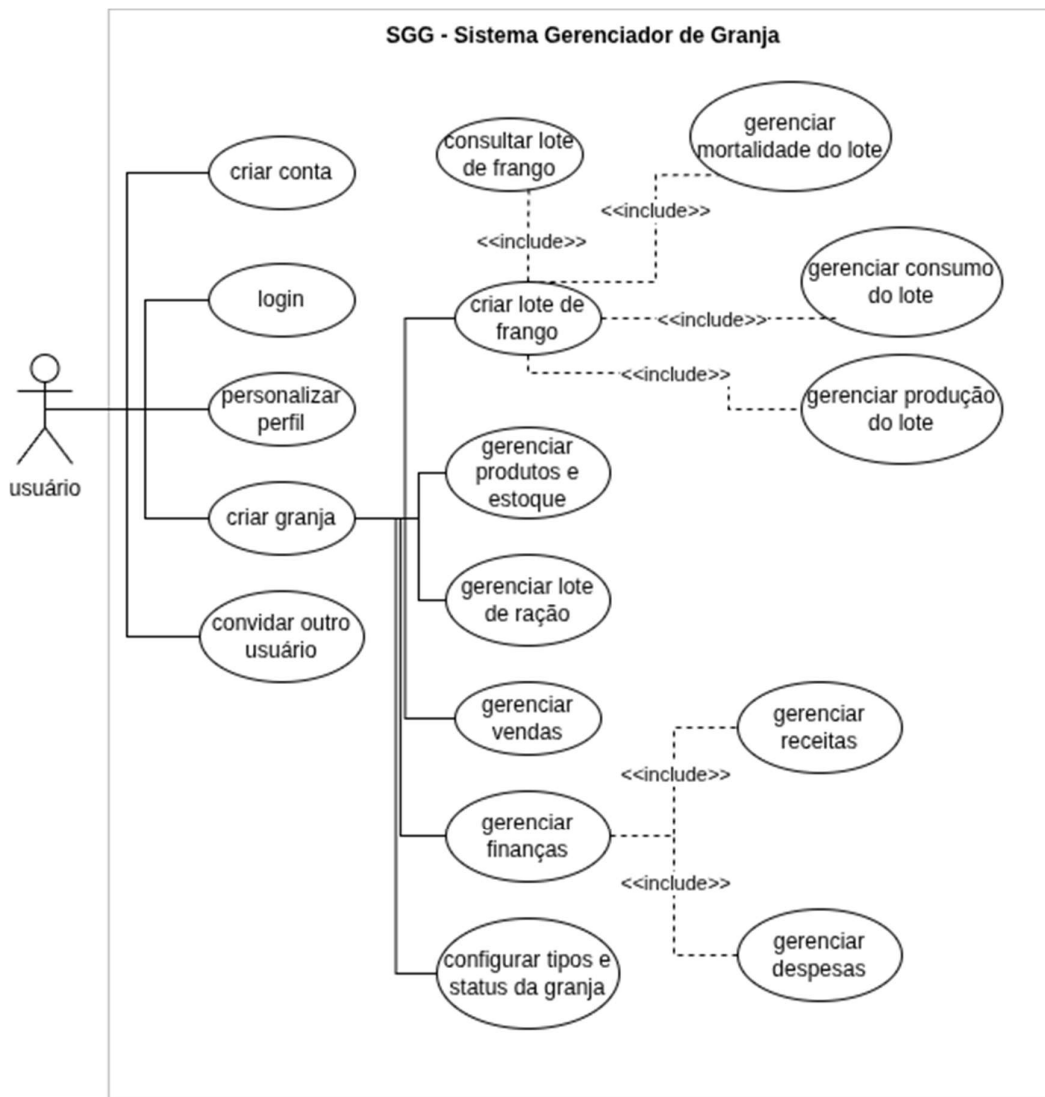
Figura 3 - Continuação do modelo lógico



Fonte: Elaborado pelo autor.

4.3 DIAGRAMAS DE CASO DE USO

Figura 4 – Diagrama de Caso de Uso



Fonte: Elaborado pelo autor

O Diagrama de Casos de Uso do SGG (Sistema Gerenciador de Granja), ilustrado na Figura 4, apresenta as principais funcionalidades disponibilizadas aos usuários do sistema, bem como as relações estruturais entre os casos de uso por meio de relacionamentos de inclusão (<<include>>).

A arquitetura de controle de acesso do sistema é estruturada em três perfis distintos de usuários, delimitando as permissões e responsabilidades operacionais:

- **Master:** Usuário responsável pela criação e propriedade inicial da granja, possuindo autoridade máxima sobre a configuração da propriedade e sobre o convite de novos usuários.

- **Administrador:** Perfil que acumula todas as permissões operacionais e gerenciais, englobando o gerenciamento de lotes de frango, produtos e estoque, ração, além do controle completo de **finanças** (receitas e despesas) e **vendas**.
- **Operador:** Perfil focado na rotina produtiva da granja, com acesso restrito às consultas e registros de lotes de frango, controle de produtos, estoque e gerenciamento de lotes de ração.

Com base nessa estrutura de papéis e nas funcionalidades modeladas, os cenários de uso do sistema dividem-se da seguinte forma:

Cenário 1 – Gestão de Acesso, Perfil e Estrutura da Granja

O fluxo inicial contempla as ações gerais que qualquer usuário autenticado ou em processo de cadastro pode realizar. Após **criar conta** e realizar o **login**, o usuário pode **personalizar perfil** e utilizar a função de **criar granja** (atribuindo-se o papel de Master). A partir da granja estabelecida, o usuário Master ou com permissões gerenciais pode **convidar outro usuário** para integrar a equipe e **configurar tipos e status da granja**, parametrizando o sistema de acordo com as necessidades da propriedade.

Cenário 2 – Operação e Controle de Lotes e Produção

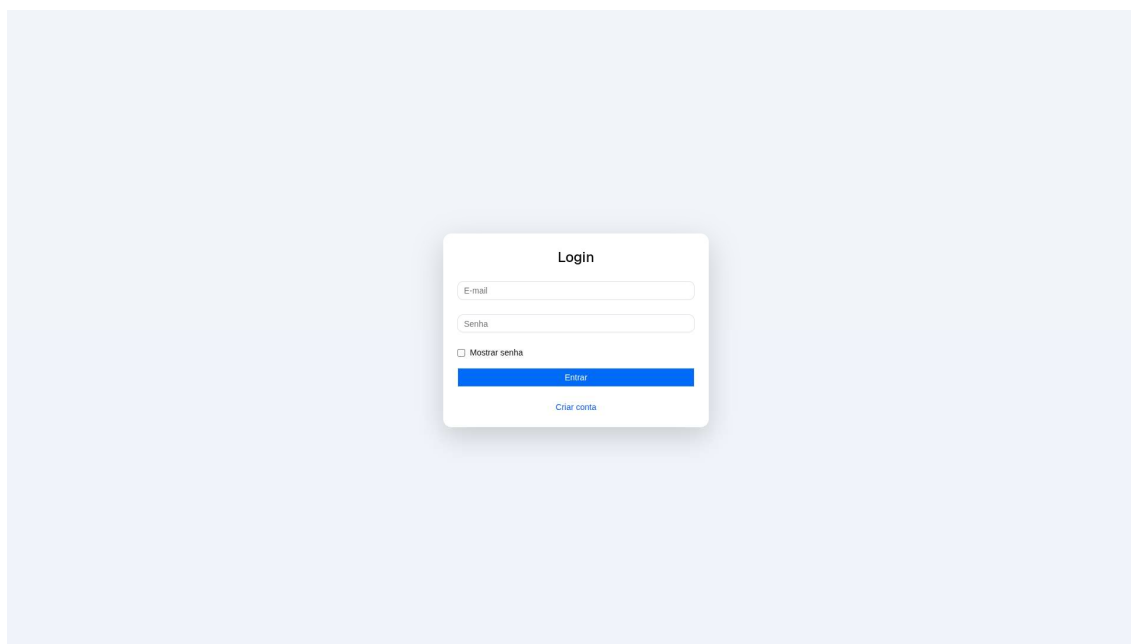
Responsabilidade central dos perfis de **Operador** e **Administrador**. Após o acesso, o usuário pode gerenciar o ecossistema produtivo dos frangos. Como ilustrado pelas relações de inclusão (<<include>>) a partir de *criar lote de frango*, o gerenciamento de um lote obrigatoriamente envolve e subsidia as atividades cotidianas de **consultar lote de frango**, **gerenciar mortalidade do lote**, **gerenciar consumo do lote** e **gerenciar produção do lote**. Paralelamente, os operadores realizam o **gerenciar produtos e estoque** e o **gerenciar lote de ração**, garantindo o suprimento e o acompanhamento logístico da granja.

Cenário 3 – Gestão Financeira e Comercial

Atribuição exclusiva do perfil de **Administrador** (e do Master). Este cenário engloba o **gerenciar vendas** e o **gerenciar finanças**. Conforme especificado pelas relações estruturais do diagrama, o caso de uso de gerenciar finanças utiliza obrigatoriamente as inclusões (<<include>>) de **gerenciar receitas** e **gerenciar despesas**, permitindo o controle rigoroso do fluxo de caixa e a sustentabilidade econômica da granja.

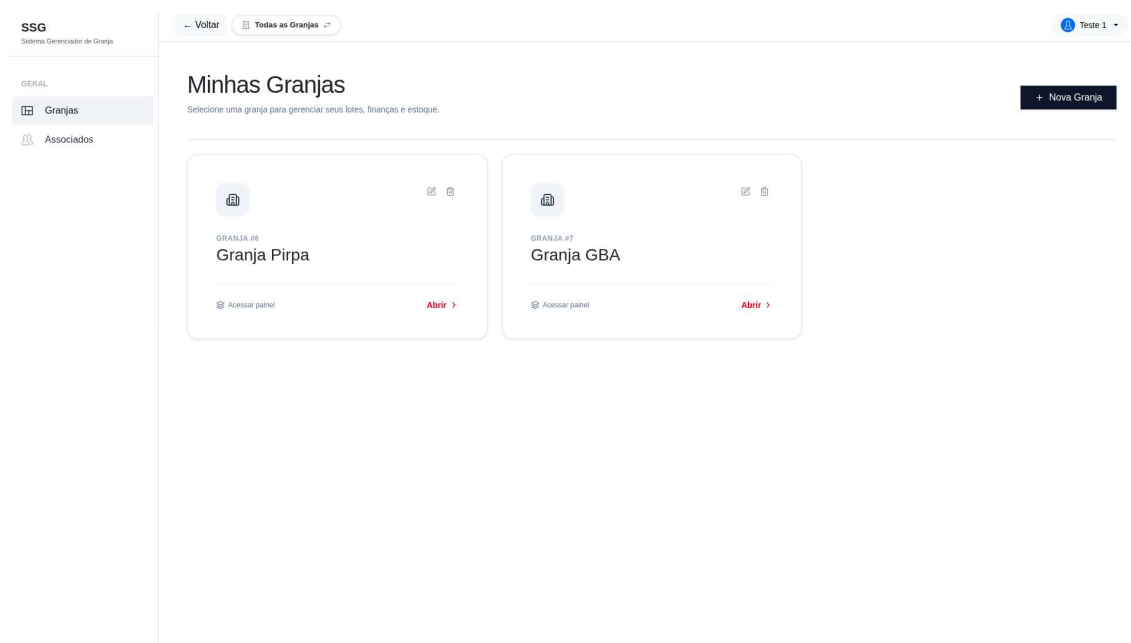
4.4 Telas do Sistema

Figura 5 – Tela de Login



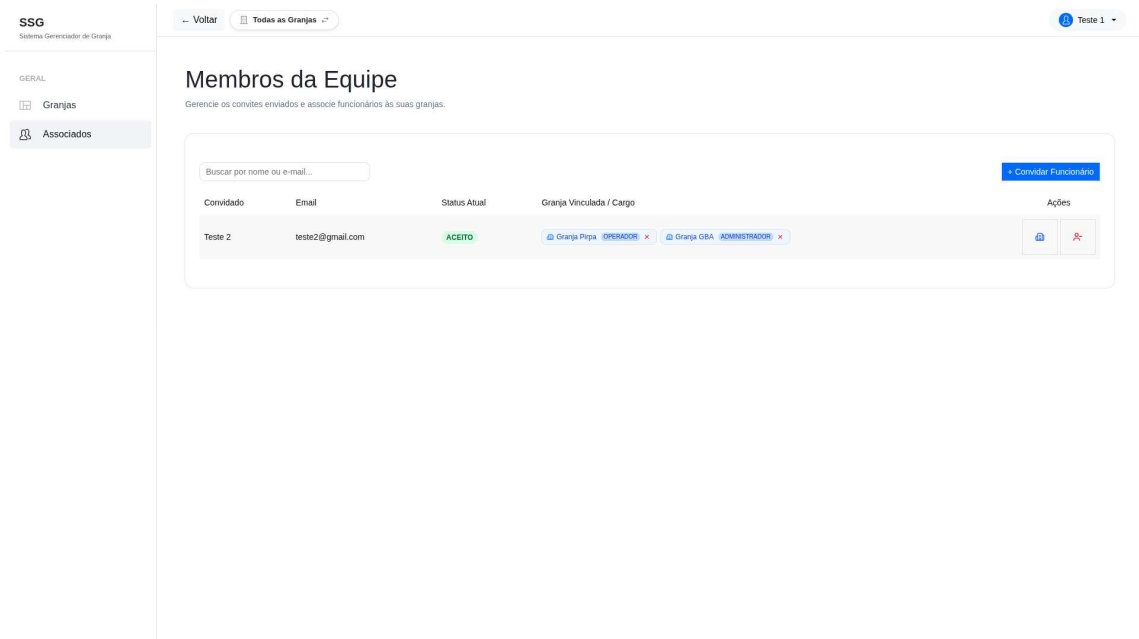
Fonte: Elaborado pelo autor.

Figura 6 – Tela de Granjas



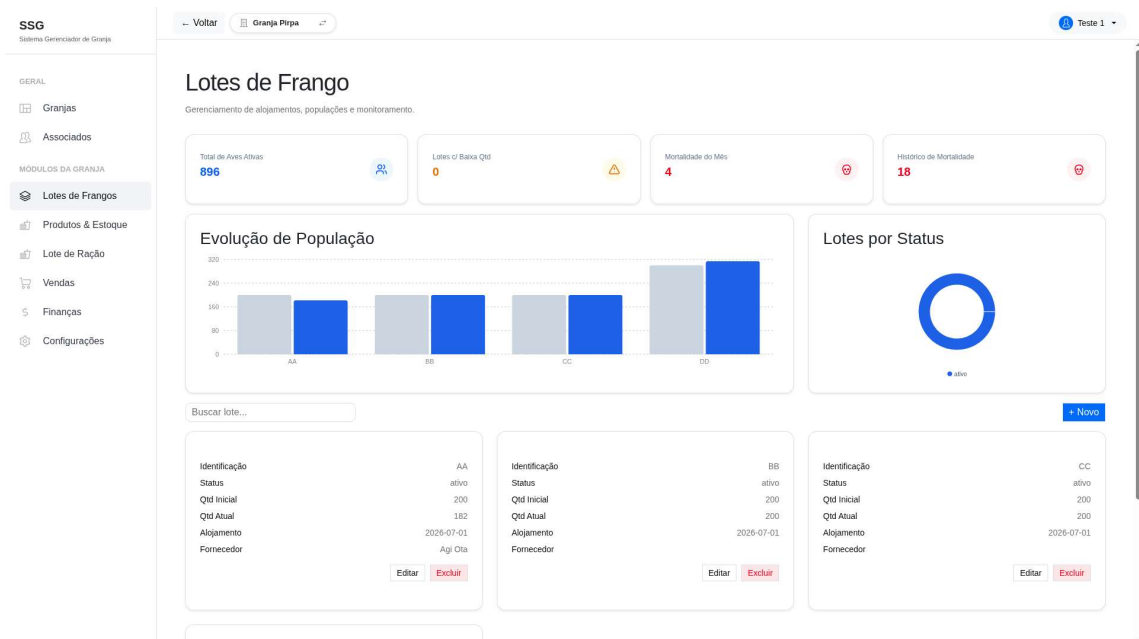
Fonte: Elaborado pelo autor.

Figura 7 – Tela Associados



Fonte: Elaborado pelo autor.

Figura 8 - Tela Lotes de Frango



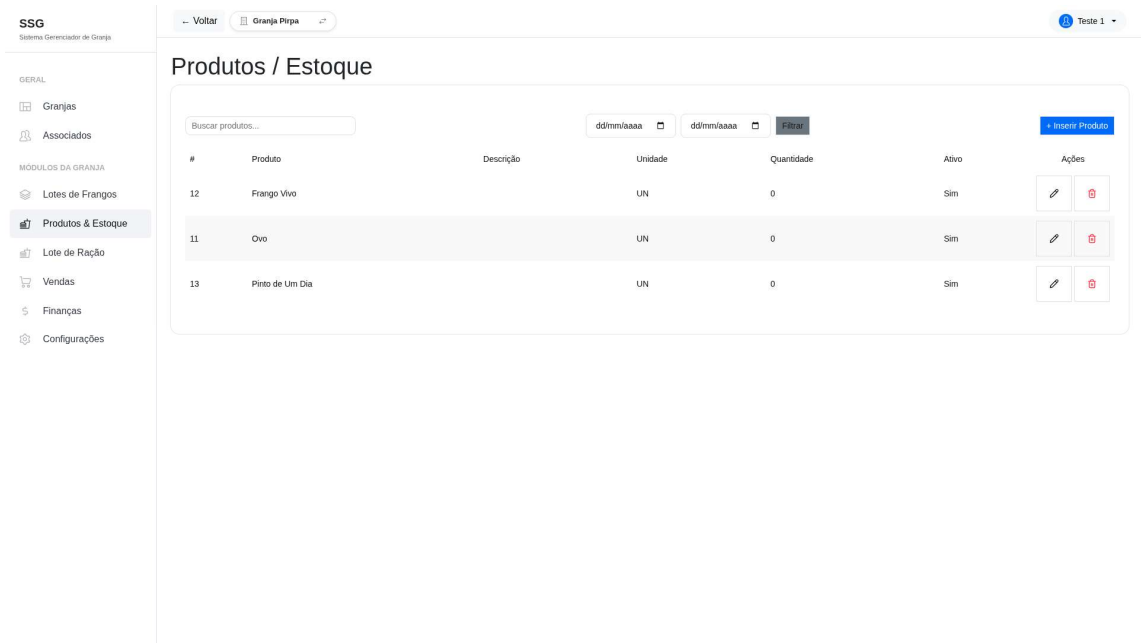
Fonte: Elaborado pelo autor.

Figura 9 - Tela de Mortalidade da Granja



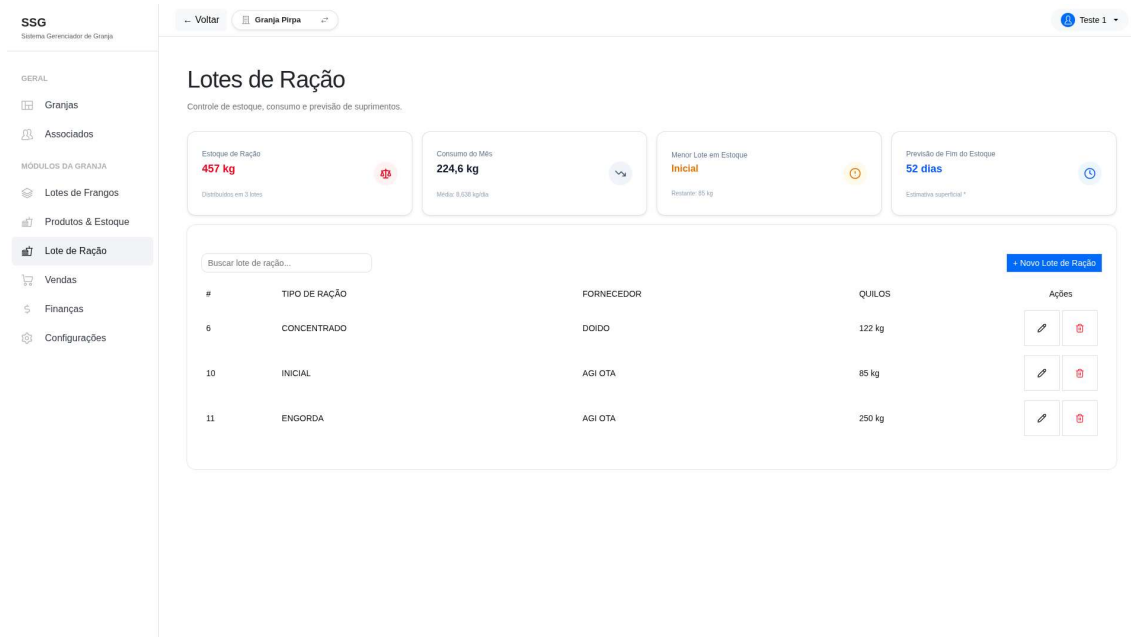
Fonte: Elaborado pelo autor.

Figura 10 – Tela de Produtos/Estoque



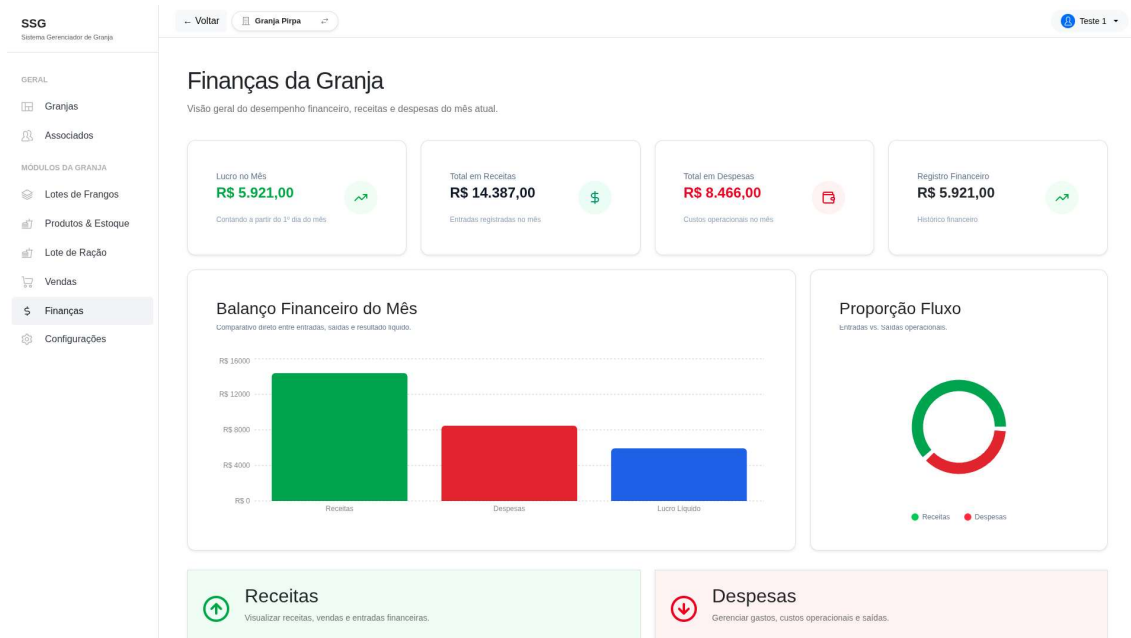
Fonte: Elaborado pelo autor.

Figura 11 – Tela de Ração



Fonte: Elaborado pelo autor.

Figura 12 – Tela de Finanças da Granja



Fonte: Elaborado pelo autor

Figura 13 – Tela de Vendas

SSG

Sistema Gerenciador de Granja

Voltar

Granja Pirpa

Teste 1

GERAL

Granjas

Associados

MÓDULOS DA GRANJA

Lotes de Frangos

Produtos & Estoque

Lote de Ração

Vendas

Finanças

Configurações

Vendas

Cientes

Buscar venda...

dd/mm/aaaa

dd/mm/aaaa

Filtrar

+ Inserir Venda

#	Cliente	Tipo Venda	Status Financeiro	Valor Total	Data	Ações
55	Agi Ota	Venda Direta	Atrasado	R\$ 12,00	27/07/2026	 
52	Agi Ota	Venda Direta	Pago	R\$ 99,00	26/07/2026	 
53	Ocaba	Venda Indireta	Em Espera	R\$ 44,00	26/07/2026	 
54	Ocaba	Venda Indireta	Pago	R\$ 99,00	26/07/2026	 
47	Agi Ota	Venda Direta	Pago	R\$ 25,00	22/07/2026	 
51	Consumidor Final	Venda Direta	Pago	R\$ 112,50	22/07/2026	 

Fonte: Elaborado pelo autor.

Figura 14 – Tela de Receitas

SSG

Sistema Gerenciador de Granja

Voltar

Granja Pirpa

Teste 1

GERAL

Granjas

Associados

MÓDULOS DA GRANJA

Lotes de Frangos

Produtos & Estoque

Lote de Ração

Vendas

Finanças

Configurações

Receitas

Total arrecadado no mês

R\$ 14.387,00

Total de receita no mês

12

Maior receita no mês

R\$ 6.550,00

Buscar receita...

dd/mm/aaaa

dd/mm/aaaa

Todos os Tipos

Todos os Status

Filtrar

+ Nova Receita

#	TIPO DE RECEITA	STATUS	TIPO DE VENDA	DATA	VALOR	DESCRIÇÃO	Ações
72	VENDA PRODUTO	PAGO	Receita Direta	31/07/2026	R\$ 222,00	-	 
73	VENDA PRODUTO	PAGO	Receita Direta	31/07/2026	R\$ 2.323,00	-	 
81	VENDA PRODUTO	PAGO	Receita Direta	27/07/2026	R\$ 3.344,00	-	 
85	VENDA PRODUTO	ATRASADO	Venda Direta	27/07/2026	R\$ 12,00	-	 
82	VENDA PRODUTO	PAGO	Venda Direta	26/07/2026	R\$ 99,00	-	 
84	VENDA PRODUTO	PAGO	Venda Indireta	26/07/2026	R\$ 99,00	-	 

Fonte: Elaborado pelo autor.

Figura 15 – Tela de Despesas

Despesas

Total Gasto no mês: R\$ 8.466,00

Maior gasto no mês: R\$ 3.200,00

#	TIPO	STATUS	LOTE DE FRANGO	DATA	VENCIMENTO	VALOR	DESCRIÇÃO	Ações
32	Ração Frangos	Pago	DD	31/07/2026	-	R\$ 657,00		
31	Ração Frangos	Atrasado		31/07/2026	-	R\$ 567,00		
17	Ração Frangos	Atrasado	DD	31/07/2026	-	R\$ 323,00		
18	Ração Frangos	Pago	DD	31/07/2026	-	R\$ 232,00		
19	Ração Frangos	Pago	DD	31/07/2026	-	R\$ 333,00		
26	Ração Frangos	Pago	DD	31/07/2026	-	R\$ 120,00		

Fonte: Elaborado pelo autor.

Figura 16 – Tela de Configurações da Granja

Configurações da Granja

Finanças

Status Financeiro Novo

Buscar status financeiro...

#	Nome	Ações
16	Pago	
17	Atrasado	
18	Em Espera	

Tipo Despesa Novo

Buscar tipo despesa...

#	Nome	Ações
31	Ração Frangos	

Fonte: Elaborado pelo autor.

4.5 Testes de Carga

Os testes de carga foram realizados com o objetivo de avaliar o comportamento da aplicação em diferentes níveis de concorrência, verificando sua capacidade de manter o desempenho e a disponibilidade à medida que o número de usuários simultâneos aumentava. Para isso, foram definidos três cenários de execução: carga leve, carga intermediária e carga pesada.

Em todos os cenários, a carga foi aplicada de forma gradual, iniciando com 0 usuários virtuais, aumentando progressivamente até o pico estabelecido para cada teste e, ao final, reduzindo novamente até 0 usuários, simulando um comportamento mais próximo da utilização real do sistema. Durante cada iteração, cada usuário virtual executou um conjunto de 17 requisições HTTP, contemplando as principais funcionalidades do sistema, incluindo consultas aos módulos de granja e financeiro.

Ao término de cada cenário, foram analisadas métricas como tempo médio de resposta, percentis P90 e P95, taxa de requisições por segundo, quantidade total de requisições processadas, taxa de sucesso das requisições (HTTP 200) e ocorrência de falhas, permitindo avaliar o comportamento da aplicação sob diferentes níveis de carga e sua capacidade de escalabilidade.

Para análise do desempenho foram consideradas as métricas de tempo médio de resposta, tempo máximo de resposta e os percentis P90 e P95. Os percentis representam o tempo abaixo do qual 90% e 95% das requisições foram concluídas, respectivamente, sendo amplamente utilizados para avaliar a experiência da maioria dos usuários em cenários de carga.

Tabela 2 - Resultados dos Testes de Carga

Métrica	Teste Leve	Teste Intermediário	Teste Pesado
Pico de usuários virtuais (VUs)	200	600	1.500
Requisições processadas	49.964	91.529	110.856
Taxa de requisições	233,84 req/s	426,68 req/s	498,22 req/s
Tempo médio de resposta	28,21 ms	330,15 ms	908,68 ms
Mediana	11,96 ms	270,66 ms	690,92 ms
P90	82,90 ms	777,36 ms	2,25 s
P95	109,53 ms	871,40 ms	2,37 s
Falhas HTTP	0%	0%	0%
Requisito de 1 s (média)	Atendido	Atendido	Atendido

Fonte: Elaborado pelo autor.

5 CONCLUSÃO

O desenvolvimento do SGG – Sistema Gerenciador de Granjas respondeu diretamente à necessidade de modernização tecnológica na gestão avícola de pequeno e médio porte. Historicamente dependente de métodos manuais ou de planilhas descentralizadas, o setor enfrenta gargalos operacionais que impactam a precisão do controle financeiro e o acompanhamento dos lotes de produção. A solução proposta neste trabalho conseguiu centralizar e automatizar esses processos em uma plataforma web acessível e intuitiva.

A escolha da arquitetura tecnológica — combinando a reatividade do React no *frontend*, a robustez do Flask (Python) no *backend* e a confiabilidade do PostgreSQL na camada de persistência de dados — demonstrou ser altamente eficiente. A implementação das rotinas de gerenciamento de lotes, insumos, clientes, fornecedores e relatórios financeiros atendeu integralmente aos requisitos funcionais e não funcionais estabelecidos na etapa de modelagem do sistema.

Adicionalmente, a validação da aplicação por meio dos testes de carga executados com a ferramenta Locust comprovou a viabilidade técnica da infraestrutura. Sob cenários de simulação de acessos simultâneos, o sistema manteve tempos de resposta médios inferiores a 1 segundo e apresentou taxa de erro nula, assegurando a estabilidade e o desempenho necessários para o uso cotidiano pelos produtores.

Por fim, o projeto não apenas cumpre seu papel acadêmico no âmbito do Projeto Integrador em Sistemas para Internet (PISI), mas também entrega uma ferramenta funcional com real potencial de impacto produtivo e econômico para os avicultores. Como perspectivas para trabalhos futuros, propõe-se:

- O desenvolvimento de um aplicativo mobile com suporte ao funcionamento *offline*, permitindo o registro de dados diretamente nos galpões de manejo sem dependência de conexão contínua com a internet;
- A integração com algoritmos de *Machine Learning* para a previsão automatizada do consumo de ração e estimativas de mortalidade/ganho de peso dos lotes;
- O envio de alertas e notificações automáticas via e-mail ou WhatsApp para avisos de reposição de estoque de insumos e datas de descarte de lotes.
- Ampliação de controle sanitário, permitindo registros mais complexos e especializados, em especial a vacinação de animais.

Diante dos resultados obtidos, conclui-se que o SGG atingiu os objetivos propostos, demonstrando ser uma solução viável para apoiar a gestão de pequenas e médias granjas, contribuindo para a organização das informações, melhoria dos processos administrativos e suporte à tomada de decisões.

REFERÊNCIAS

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Censo Agropecuário 2017: resultados definitivos**. Rio de Janeiro: IBGE, 2019. Disponível em: <https://www.ibge.gov.br/>. Acesso em: 18 jul. 2026.

LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informação gerenciais**. 11. ed. São Paulo: Pearson Education do Brasil, 2014.

O'BRIEN, James A.; MARAKAS, George M. **Administração de sistemas de informação**. 15. ed. Porto Alegre: AMGH, 2013.

SERVIÇO BRASILEIRO DE APOIO ÀS MICRO E PEQUENAS EMPRESAS (SEBRAE). **Pequenos negócios em números**. Brasília, DF: Sebrae, 2023. Disponível em: <https://sebrae.com.br/>. Acesso em: 18 jul. 2026.

ALCANTARA, Isabela Romanha de; SCHMIDT, João Guilherme Araujo; VIAN, Carlos Eduardo de Freitas; BELARDO, Guilherme. **Agriculture 4.0: Origin and features in the world and Brazil**. *Quaestum*, Maringá, v. 2, n. 1, 2021. DOI: 10.22167/2675-441X-20210564.

EMBRAPA. **Visão 2030: o futuro da agricultura brasileira**. Brasília, DF: Embrapa, 2023.

LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informação gerenciais**. 17. ed. São Paulo: Pearson, 2021.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software: uma abordagem profissional**. 9. ed. Porto Alegre: AMGH, 2022.

SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019.

NIELSEN, Jakob. **Usability Engineering**. Boston: Academic Press, 1993.

REACT. **React Documentation**. Meta Platforms, Inc., 2026. Disponível em: <https://react.dev/>. Acesso em: 29 jul. 2026.

PALLETS. **Flask Documentation**. Pallets Projects, 2026. Disponível em: <https://flask.palletsprojects.com/>. Acesso em: 29 jul. 2026.

PALLETS. **Flask-SQLAlchemy Documentation**. Pallets Projects, 2026. Disponível em: <https://flask-sqlalchemy.palletsprojects.com/>. Acesso em: 29 jul. 2026.

MIGUEL GRINBERG. **Flask-Migrate Documentation**. 2026. Disponível em: <https://flask-migrate.readthedocs.io/>. Acesso em: 29 jul. 2026.

FLASK-CACHING. **Flask-Caching Documentation**. 2026. Disponível em: <https://flask-caching.readthedocs.io/>. Acesso em: 29 jul. 2026.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL Documentation**. 2026. Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 29 jul. 2026.

REDIS. **Redis Documentation**. Redis Ltd., 2026. Disponível em: <https://redis.io/docs/>. Acesso em: 29 jul. 2026.

DOCKER INC. **Docker Documentation**. 2026. Disponível em: <https://docs.docker.com/>. Acesso em: 29 jul. 2026.

AXIOS. **Axios Documentation**. OpenJS Foundation, 2026. Disponível em: <https://axios-http.com/>. Acesso em: 29 jul. 2026.

NGINX. **NGINX Documentation**. F5, Inc., 2026. Disponível em: <https://nginx.org/en/docs/>. Acesso em: 29 jul. 2026.