

Relatório Técnico: Sistema de Automação Chicken House

Projeto: Sistema de Controle de Ambiência e Alimentação Avícola

Versão: 1.0 **Data:** 25/11/2025

1. Visão Geral e Objetivo do Projeto

1.1. Contextualização do Problema

A gestão manual de aviários enfrenta desafios significativos relacionados à precisão dos manejos diários. A falha no controle da iluminação e a dosagem incorreta de ração são os principais fatores que levam ao desperdício financeiro e à queda na produtividade de ovos. Atualmente, a dependência exclusiva da intervenção humana para ligar/desligar luzes e liberar ração gera inconsistências operacionais e riscos ao bem-estar das aves.

1.2. Objetivo Geral

O Sistema de Automação Chicken House é uma solução de software integrada a hardware projetada para monitorar e atuar sobre o ambiente avícola. O objetivo central é modernizar o processo produtivo, garantindo que as normas zootécnicas sejam cumpridas rigorosamente sem a necessidade de supervisão humana constante. O sistema visa transformar dados brutos (leituras de sensores e inputs de biometria) em ações de controle precisas.

1.3. Objetivos Específicos e Benefícios

O sistema foi desenvolvido para atender a três pilares fundamentais:

- Otimização da Ambiência (Fotoperíodo): Assegurar o cumprimento do ciclo de luz de 14 a 16 horas diárias, essencial para o estímulo hormonal e a produção de ovos. O sistema não apenas liga e desliga lâmpadas, mas gerencia a intensidade luminosa (10-20 lux) para simular condições naturais, reduzindo o estresse das aves.

- Eficiência Nutricional e Redução de Custos: A alimentação representa o maior custo de produção. O sistema elimina o desperdício ao calcular a quantidade exata de ração baseada no peso médio e na fase de vida do lote (pintinho, crescimento, postura). Isso previne tanto a desnutrição quanto a obesidade das aves, fatores que impactam diretamente a lucratividade.

- Segurança e Auditoria: Prover ao Gestor uma ferramenta de rastreabilidade. Através de logs imutáveis e relatórios de consumo, o sistema permite identificar falhas operacionais, monitorar o uso de energia e auditar a quantidade de insumos utilizados, garantindo transparência na gestão da granja.

2. Escopo Funcional (Módulos)

O sistema foi dividido em três subsistemas lógicos baseados nos fluxos de dados identificados:

2.1. Módulo de Iluminação (Controle de Ambiência)

- **Responsabilidade:** Gerenciar o fotoperíodo das aves.
- **Entradas:** Leitura de sensores de luminosidade (Lux) e horários configurados pelo usuário.
- **Saídas:** Comandos de Ligar/Desligar para o relé da iluminação.
- **Requisitos Atendidos:** RF01, RF02, RF03.

2.2. Módulo de Nutrição (Alimentação Automática)

- **Responsabilidade:** Calcular e distribuir a dieta balanceada.
- **Entradas:** Quantidade de aves, peso médio do lote e horários de trato.
- **Saídas:** Acionamento do motor do distribuidor de ração.
- **Requisitos Atendidos:** RF04, RF05, RF07.

2.3. Módulo de Gestão e Auditoria

- **Responsabilidade:** Monitoramento, segurança e relatórios.
- **Entradas:** Logs de operação e status dos sensores.
- **Saídas:** Relatórios de consumo, alertas de falha e logs imutáveis.
- **Requisitos Atendidos:** RF06, RF08, RF09, RNF03.

3. Dicionário de Dados

3.1. Entidades de Dados

Nome do Dado	Tipo de Dado	Tamanho/Formato	Descrição	Restrições / Validação	Origem (Req)
qtd_galinhas	Inteiro	-	Número total de aves vivas no lote atual.	Deve ser > 0. Atualizado apenas por usuários autorizados.	RF07
peso_medio	Real (Double)	#.### kg	Peso médio unitário das aves para cálculo nutricional.	Máx: 5.0kg (preventivo). Não aceita negativo.	RF04
hora_acendimento	Hora	HH:mm	Horário programado para ligar a luz artificial.	Formato 24h.	RF02
hora_desligamento	Hora	HH:mm	Horário programado para desligar a luz.	Diferente da hora de acendimento.	RF02
intensidade_luz	Inteiro	Lux	Leitura atual do sensor de luminosidade.	Faixa ideal: 10-20 Lux.	RF03
log_racao_diaria	String/Texto	-	Registro concatenado de data e consumo total.	Imutável (Read-only após gravação).	RF06, RNF03
status_sensor	Booleano	True/False	Indica saúde do hardware (True=Ok, False=Falha).	Monitorado em tempo real (< 2s).	RF09, RNF01

3.2. Fluxos de Dados (Entrada/Saída)

Fluxo	Origem	Destino	Estrutura de Dados Envolvida	Descrição
Configurar Timer	Funcionário	Sistema	hora_acendimento, hora_desligamento	Definição do ciclo circadiano das aves.
Input Biometria	Funcionário	Sistema	qtd_galinhas, peso_medio	Atualização dos dados vitais para cálculo de ração.
Sinal de Controle	Sistema	Lâmpada (Hw)	Binário (0/1)	Comando enviado ao relé de potência.
Alerta de Falha	Sistema	Interface (GUI)	status_sensor, Timestamp	Notificação visual imediata de erro crítico.

4. Matriz de Rastreabilidade de Testes

Esta seção conecta os 10 Casos de Teste desenvolvidos (Unitários em Java) aos Requisitos Funcionais (RF) e Não Funcionais (RNF) documentados.

ID Teste	Nome do Cenário	Requisito Validado	Justificativa Técnica
CT-01	Configuração de Iluminação	RF02, RNF05	Garante persistência correta das variáveis de horário.
CT-02	Automação da Lâmpada	RF01	Valida a lógica condicional (if time >= start && time < end).
CT-03	Cálculo de Ração (Matemático)	RF04	Valida a fórmula: (Qtd * Peso * Fator).
CT-04	Integração Distribuidor	RF05	Valida o disparo do comando no horário exato (Trigger).
CT-05	Formatação de Logs	RF06, RNF03	Assegura a integridade da String gerada para auditoria.
CT-06	Atualização de Estoque	RF07	Testa operações aritméticas de saldo de animais.
CT-07	Trava de Segurança (Ração)	RF04 (Restrição)	Bloqueia valores double acima do limiar biológico seguro.
CT-08	Monitoramento de Sensores	RF09, RNF01	Detecta exceções ou valores nulos/negativos nos inputs.
CT-09	Agendador de Alimentação	RF05	Verifica matching de Strings de horário em listas/arrays.
CT-10	Acumulador Financeiro	RF08	Valida a soma incremental de variáveis double (Consumo Mês).

5. Considerações Finais e Próximos Passos

O sistema demonstra alta aderência aos requisitos de segurança alimentar e bem-estar animal. A implementação dos testes unitários listados na seção 4 garante a robustez da Regra de Negócio antes da integração com o hardware físico (Arduino/ESP32 e sensores).

Recomendação: A próxima etapa deve focar na criação da Interface Gráfica (Java Swing) conforme especificado no RNF04, conectando os campos de texto aos métodos validados nos testes unitários.