

# Sobre o sistema SkyControl

## Descrição do Sistema e Visões Arquiteturais (Concerns)

Esta seção define o escopo do problema e como a arquitetura soluciona as preocupações principais (concorrência, distribuição e latência).

### O Problema

Em cenários de monitoramento aéreo, operadores humanos enfrentam dificuldades para gerenciar múltiplos drones simultaneamente. O desafio principal é a **sobrecarga cognitiva**: acompanhar a posição geográfica, o nível de bateria e alertas de emergência de uma frota inteira em tempo real, enquanto se tenta coordenar movimentos táticos (formações). Sistemas monolíticos tradicionais falham aqui devido à latência na atualização de dados e ao acoplamento que trava a interface quando o processamento de dados é intenso.

### A Solução

O SkyControl é uma plataforma de Comando e Controle (C2) distribuída. Ela resolve o problema segregando a responsabilidade em três camadas físicas distintas:

1. **Camada de Borda (Edge/IoT):** Drones simulados autônomos que gerenciam sua própria física e segurança.
2. **Camada de Mensageria:** Um *broker* de alta performance que garante a entrega de dados mesmo em picos de carga.
3. **Camada de Apresentação e Controle:** Um backend reativo que processa regras de negócio e "empurra" (*push*) dados para a interface visual.

### Pontos de Vista e Preocupações

- **Visão Lógica (Estrutura):** O sistema utiliza o padrão **MVC** no Backend para organizar a API REST e Serviços, enquanto o Simulador utiliza o padrão **Strategy/State** para gerenciar o comportamento de voo.
- **Visão de Processos (Concorrência):**
  - Simulador: Cada drone roda em sua própria Thread (implementando Runnable em SimulatedDrone.java), garantindo que o cálculo de rota de um não bloqueie o outro.
  - Backend: Utiliza listeners assíncronos (@RabbitListener) para processar telemetria sem bloquear a thread principal da API HTTP.
- **Visão de Dados (Persistência):** A persistência é feita de forma leve utilizando arquivos JSON (JsonDatabaseService.java), priorizando a velocidade de leitura/escrita para o protótipo.

