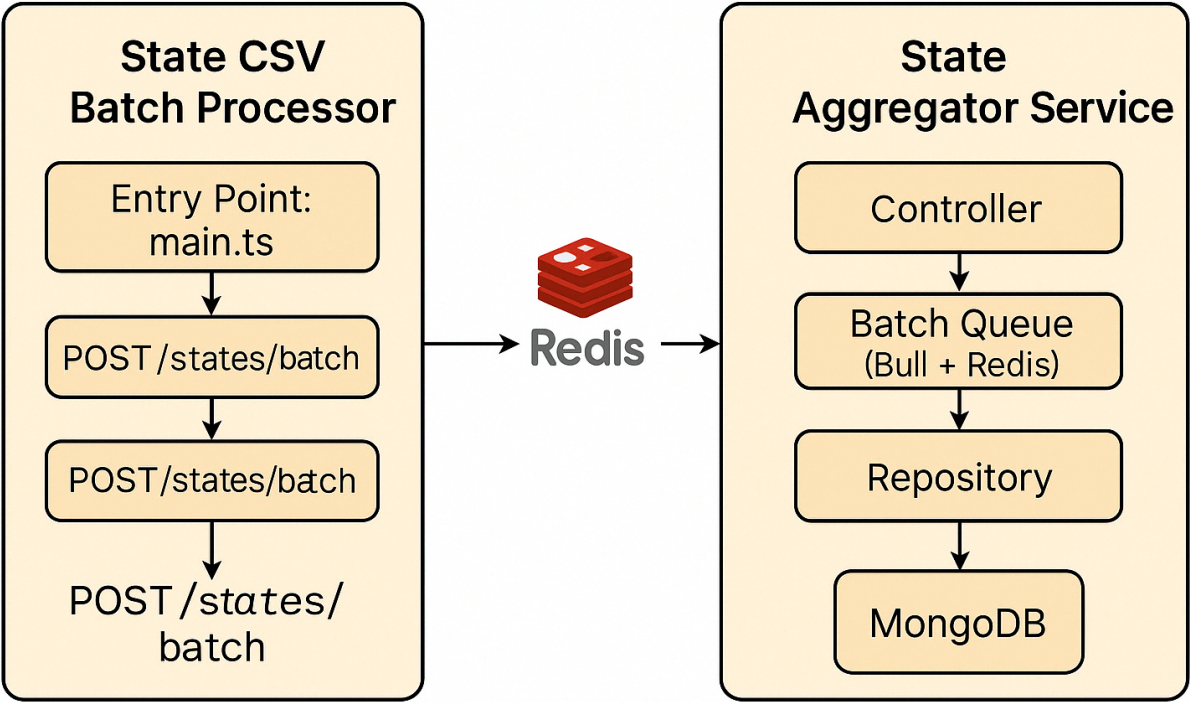
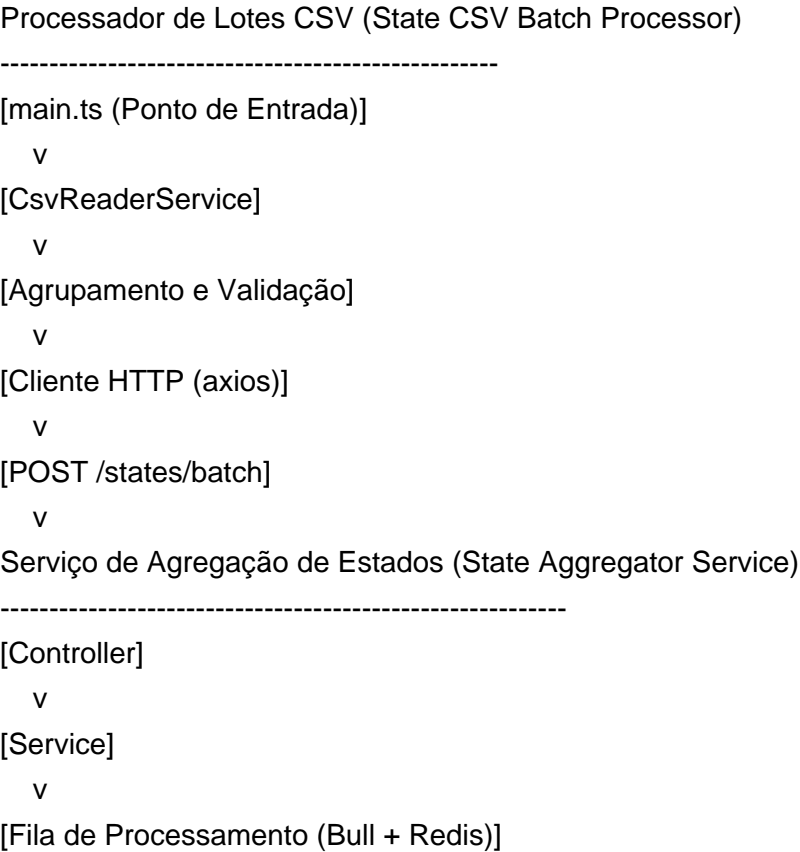


Documentação da Arquitetura - Processamento em Lote com Redis



Architecture Design

1. Visual em Camadas



# Documentação da Arquitetura - Processamento em Lote com Redis

v

[Repository]

v

[MongoDB]

## 2. Objetivo da Arquitetura

- Ingestão Automatizada de Dados: processar arquivos CSV com registros populacionais de forma eficiente.
- Transferência Confiável: envio dos dados em lotes via HTTP com escalabilidade e tolerância a falhas.
- Agregação e Armazenamento Centralizados: os dados são recebidos e armazenados agregadamente no MongoDB.
- Processamento Assíncrono e Escalável: uso do Redis como backend de fila (Bull) desacopla a API do processamento.
- Acesso Fácil aos Dados: a API permite consulta, monitoramento e gerenciamento dos registros agregados.

## 3. Redis na Arquitetura

- Propósito: Redis é usado como armazenamento em memória para o sistema de filas Bull, permitindo o enfileiramento e processamento assíncrono dos lotes.
- Funcionamento: ao receber um lote pela rota /states/batch, ele é colocado em uma fila com backend Redis. Workers (que podem ser escalados) consomem a fila e processam os dados no MongoDB.
- Benefícios:
  - Desacopla requisições HTTP do processamento.
  - Suporte a tentativas automáticas, atrasos e agendamento.
  - Monitoramento da fila e dos jobs em tempo real.
  - Escalabilidade horizontal com múltiplos workers.

## 4. Resumo

- O processador de CSV automatiza a leitura e agrupamento dos dados e envia para o backend.
- O serviço agregador de estados recebe, enfileira, processa e armazena os dados.
- O Redis, com Bull, atua como fila escalável e confiável para processamentos pesados.
- O MongoDB armazena os dados agregados finais.

Essa separação garante foco, manutenção, escalabilidade e robustez para ambientes de produção.