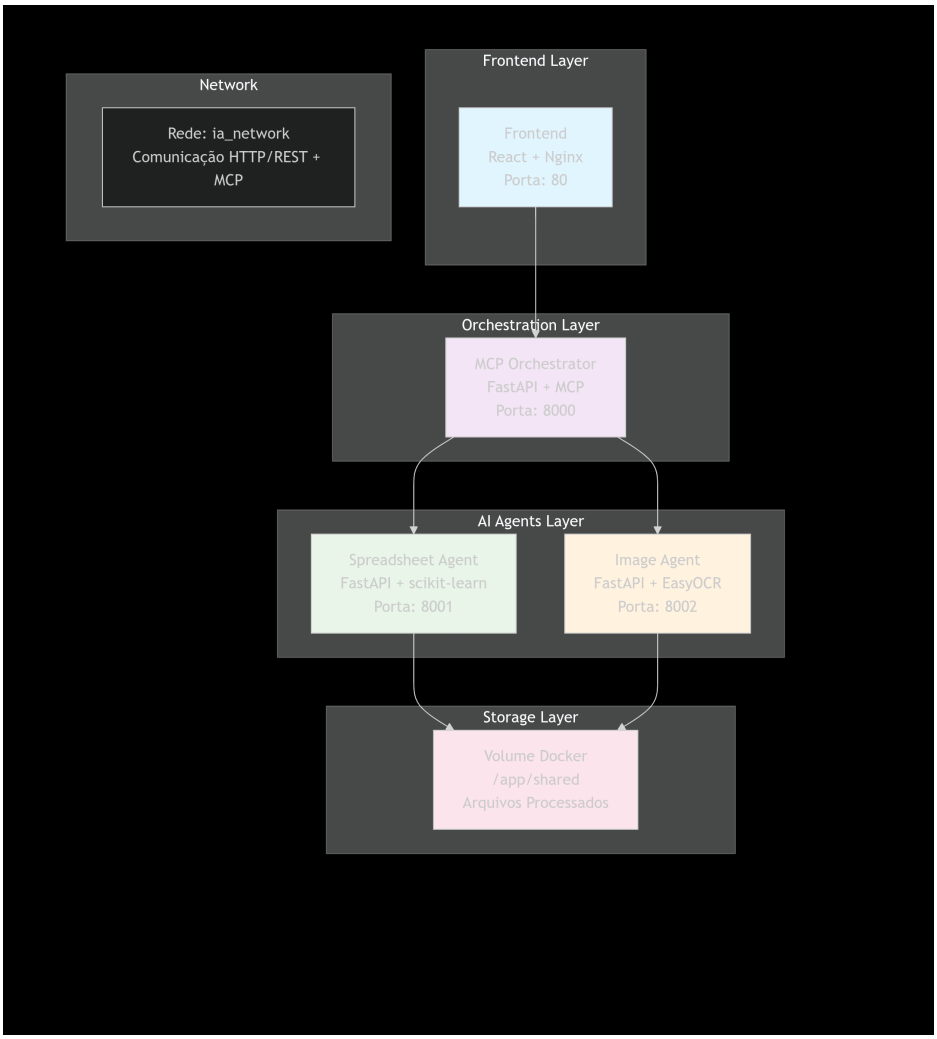
📋 Documentação Arquitetônica - IA Data Analyzer

🎯 Visão Geral do Sistema

Sistema distribuído com múltiplos agentes de IA para análise automatizada de dados, implementando comunicação via MCP (Model Context Protocol) e containerização Docker.

🏗️ Arquitetura do Sistema

Diagrama de Componentes



🔧 Componentes

1. Frontend Web (React)

- **Porta:** 80
- **Tecnologias:** React, Vite, Axios
- **Função:** Interface do usuário estilo ChatGPT

2. MCP Orchestrator (FastAPI)

- **Porta:** 8000
- **Tecnologias:** FastAPI, Python, MCP
- **Função:** Roteamento inteligente entre agentes

3. Spreadsheet Agent (FastAPI)

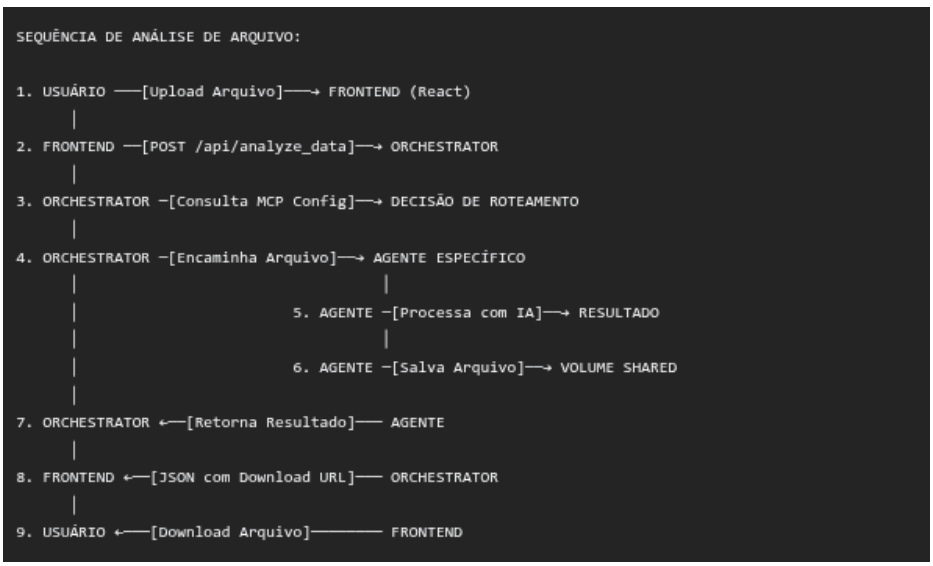
- **Porta:** 8001
- **IA:** Isolation Forest (scikit-learn)
- **Função:** Detecção de outliers em planilhas

4. Image Agent (FastAPI)

- **Porta:** 8002
- **IA:** EasyOCR + OpenCV
- **Função:** Extração de dados numéricos de imagens

🔄 Fluxo de Comunicação MCP

Fluxograma



Configuração MCP (mcp_config.yml)

```yaml
agents:
  spreadsheet_agent:
    endpoint: "http://spreadsheet_agent:8001"
    capabilities: ["outlier_detection", "data_cleaning"]
    input_types: [".csv", ".xlsx"]

  image_agent:
    endpoint: "http://image_agent:8002"
    capabilities: ["ocr_extraction", "image_analysis"]
    input_types: [".jpg", ".png", ".jpeg"]
```

Regras de Roteamento

- **Arquivos CSV/XLSX** → Spreadsheet Agent
- **Arquivos JPG/PNG** → Image Agent

🛡️ Modelagem de Ameaças

Ameaças Identificadas

A1: Vazamento de Dados

- **Risco:** Alto
- **Medidas:**
  - Criptografia em trânsito (HTTPS)
  - Isolamento de rede Docker
  - Validação de tipos de arquivo

A2: Ataque DDoS

- **Risco:** Médio
- **Medidas:**
  - Rate limiting nos endpoints
  - Health checks automáticos
  - Containerização com limites de recursos

A3: Upload Malicioso

- **Risco:** Médio
- **Medidas:**
  - Validação estrita de tipos MIME

- Sanitização de nomes de arquivo
- Execução em containers isolados

Medidas de Mitigação Implementadas

1. **Isolamento de Rede**
   - Rede Docker privada para comunicação interna
   - Exposição apenas das portas necessárias
2. **Validação de Entrada**
   - Verificação de tipos de arquivo
   - Limites de tamanho de upload
   - Sanitização de dados
3. **Monitoramento**
   - Logs estruturados por container
   - Health checks automáticos
   - Métricas de performance

📊 Visão de Implantação

Infraestrutura Docker

```
services:
  frontend:    # React + Nginx
  orchestrator: # FastAPI + MCP
  spreadsheet_agent: # FastAPI + scikit-learn
  image_agent:      # FastAPI + EasyOCR
```

Comunicação de Rede

- **Frontend** → **Orchestrator** (porta 8000)
- **Orchestrator** → **Agents** (portas 8001, 8002)
- **Todos os serviços** na rede `ia_network`

🚀 Considerações de Performance

Otimizações Implementadas

- **Containerização:** Isolamento e escalabilidade

- **Cache de Modelos:** EasyOCR carrega modelos uma vez
- **Processamento Assíncrono:** FastAPI com async/await
- **Otimização de Imagens:** OpenCV headless

Métricas de Monitoramento

- Tempo de resposta por agente
- Taxa de acerto do OCR
- Detecção de outliers
- Uso de recursos por container

🤔 Melhorias Futuras

1. **Segurança**
    - Implementar autenticação JWT
    - Adicionar HTTPS
    - Logs de auditoria
2. **Performance**
    - Cache Redis para resultados
    - Load balancing entre agents
    - Otimização de modelos
3. **Funcionalidades**
    - Suporte a mais tipos de arquivo
    - Análise em tempo real
    - Dashboard de métricas