

# PLANEJAMENTO E IMPLEMENTAÇÃO

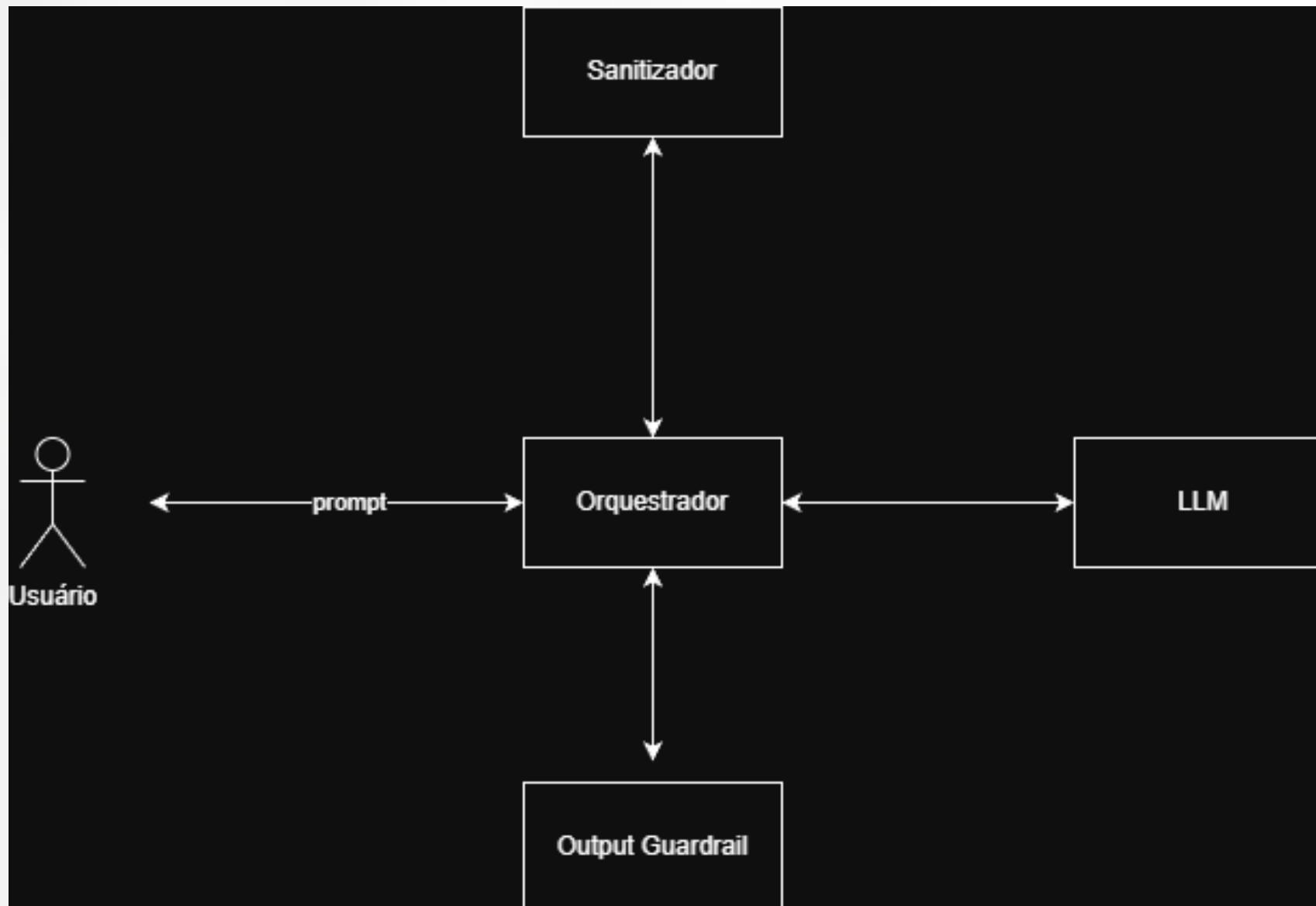
Grupo: Segurança de prompts em modelos de LLMS

Juan Gustavo, Lucas Emanoel, Lucas Messias, Joás Vitor e  
João Victor

# ARQUITETURA

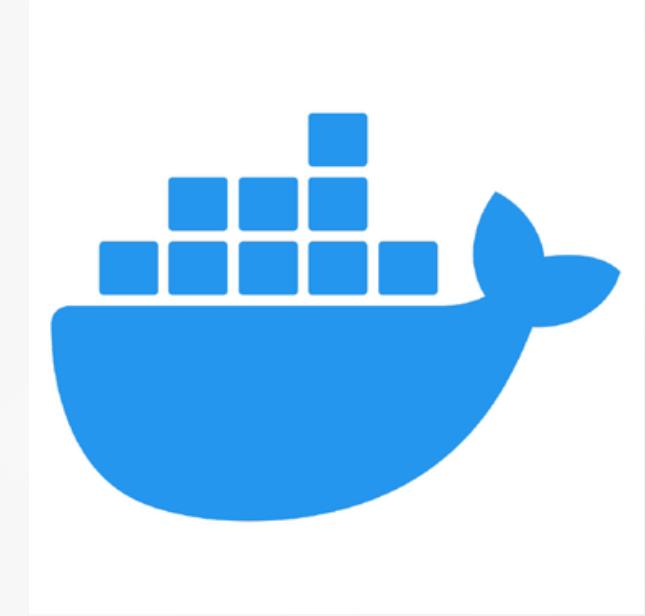
Composta por 4 entidades:

- Usuário (prompt)
- orquestrador
- sanitizador
- Serviço de LLM
- Output guardrail



# TECNOLOGIAS

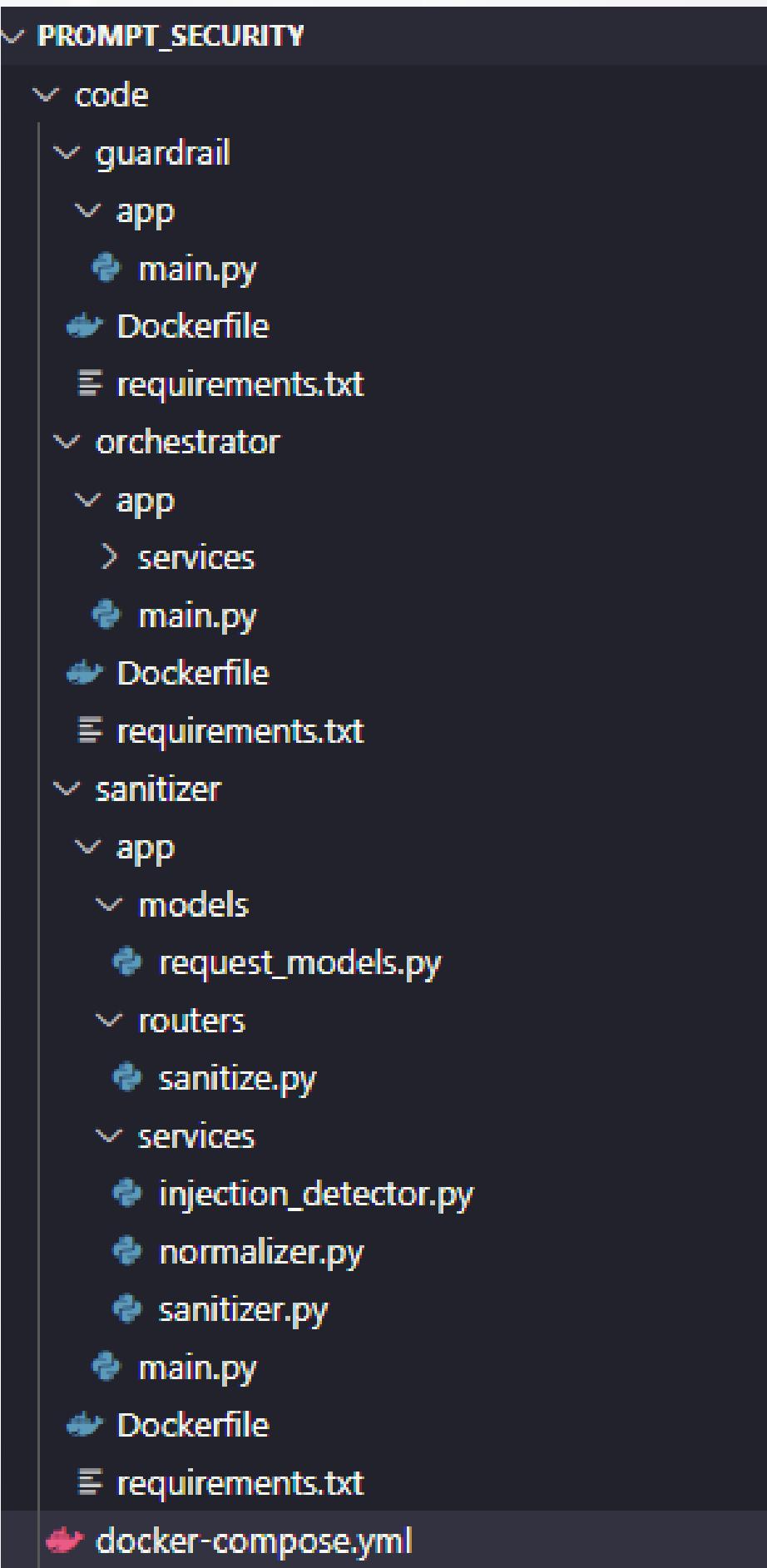
- Python
- FastApi/Uvicorn
- OpenAi Api
- Docker
- Draw.io



# ESTRUTURA DO PROJETO

Foi montado o código inicial das entidades envolvidas no projeto

Foi montado o arquivo Dockerfile e docker-compose para automatizar a criação da estrutura do projeto



```
PROMPT_SECURITY
  code
    guardrail
      app
        main.py
        Dockerfile
        requirements.txt
    orchestrator
      app
        services
        main.py
        Dockerfile
        requirements.txt
    sanitizer
      app
        models
          request_models.py
        routers
          sanitize.py
        services
          injection_detector.py
          normalizer.py
          sanitizer.py
        main.py
        Dockerfile
        requirements.txt
  docker-compose.yml
```



```
version: '3.9'
services:
  sanitizer:
    build: ./sanitizer
    container_name: sanitizer_service
    ports:
      - "8000:8000"
    networks:
      - llm_net
  orchestrator:
    build: ./orchestrator
    container_name: orchestrator_service
    depends_on:
      - sanitizer
    ports:
      - "7000:7000"
    networks:
      - llm_net
  guardrail:
    build: ./guardrail
    container_name: output_guardrail_service
    ports:
      - "6000:6000"
    networks:
      - llm_net
networks:
  llm_net:
    driver: bridge
```



# OBRIGADO