Arquitetura de Software - 2025.01

Trabalho Final

Prof. Gilmar Ferreira Arantes

Alunos: Jair, João Gabriel França, Leonardo Moreira, Matheus Cascão, Vitor Paulo

2025





Sumário

- 1. <u>Componentes</u>
- 2. Tema
- 3. Requisitos Funcionais
- 4. Requisitos de Qualidade
- 5. <u>Diagramas</u>
 - 5.1. <u>Diagrama de contexto</u>
 - 5.2. <u>Diagrama de containers</u>
 - 5.3. <u>Diagrama de componentes</u>
 - 5.4. <u>Diagrama de caso de uso iteração 1</u>
 - 5.5. <u>Diagrama de caso de uso iteração 2</u>
- 6. <u>Considerações Finais</u>



INF



Matheus Cascão



Jair Rodrigues



João Gabriel



Leonardo Moreira



Vitor Godoi



Tema

Tema: Sistema de Agricultura Inteligente com IoT





A Agricultura Inteligente, aliada à Internet das Coisas (IoT), representa uma transformação digital no campo, tornando o processo agrícola mais eficiente, sustentável e produtivo. Por meio da integração de sensores, redes de comunicação e algoritmos inteligentes, os agricultores conseguem tomar decisões baseadas em dados em tempo real, reduzindo desperdícios, melhorando a produtividade e minimizando impactos ambientais.

Benefícios Esperados:

- Aumento da produtividade
- Redução de custos operacionais
- Sustentabilidade no uso de recursos naturais
- Tomada de decisão baseada em dados



Nº	Requisitos Funcionais	Prioridade
1	O sistema deve adaptar a irrigação do solo com base em previsões metereológicas (umidade, chuva, seca).	ALTA
2	Rede de sensores de solo com comunicação sem fio que colete dados em tempo real sobre umidade, temperatura, pH e nutrientes em diferentes zonas da plantação, permitindo irrigação e fertilização automatizadas e personalizadas para cada microárea da lavoura.	ALTA
3	O sistema deve permitir que o pragas ou outros problemas nas plantações sejam identificados e um alerta seja gerado aos usúarios.	ALTA
4	O sistema deve fornecer recomendações de plantio com base em análises de solo.	ALTA
5	O sistema deve analisar as condições do solo e do clima para prever o melhor momento para a colheita	ALTA
7	O sistema deve utilizar câmeras e sensores IoT combinados com IA para identificar precocemente a previamente a presença de pragas na plantação, permitindo ações preventivas e reduzindo o uso de agrotóxicos.	ALTA
8	O sistema deve implementar não só sensores para medir estatísticas de plantio e terra, mas também para monitorar equipamentos e máquinas agrícolas, capazes de prever manutenções e checagem de elementos de segurança, como: pressão de pneus, temperatura, qualidade do óleo e notificar tudo isso por meio do app.	ALTA
9	O software deve acionar automaticamente o sistema de irrigação caso os os níveis de umidade do solo estiverem abaixo do ideal.	ALTA
10	O sistema deve utilizar algorítmos de aprendizagem de máquina e estatística para prever e otimizar a produção. Dados históricos devem ser utilizados para sugerir melhorias no processo agrícola.	ALTA

INF





Nº	Requisito de Qualidade	Prioridade
6	O sistema deve funcionar mesmo com conexão limitada, usando sincronização offline e transmissão dos dados quando a rede estiver disponível.	ALTA



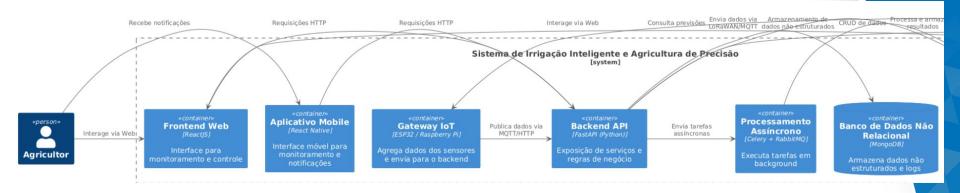
Nível	Influência do requisito 6
Contexto	Menções à sincronização offline nas descrições do frontend e banco.
Contêineres	Uso explícito de armazenamento local (SQLite/IndexedDB) e caminho de sincronização com backend.
Componentes	Módulo dedicado à sincronização offline, com detalhamento técnico da lógica de integração.



Diagramas

Diagrama de Contexto - Primeira Iteração







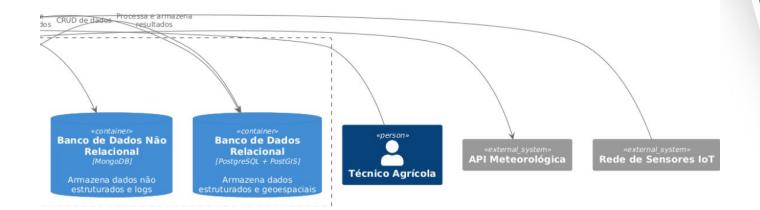
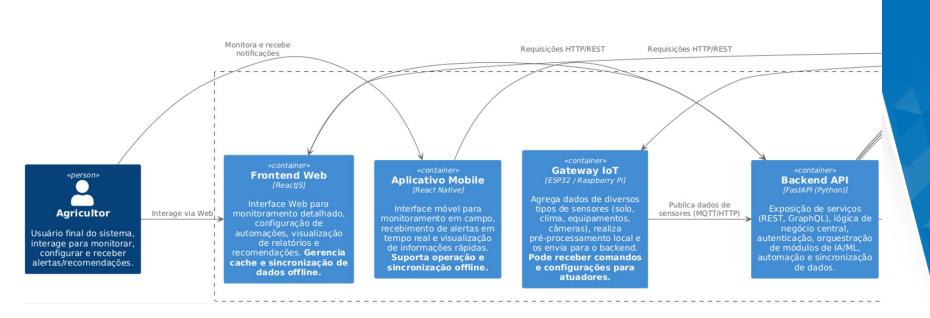
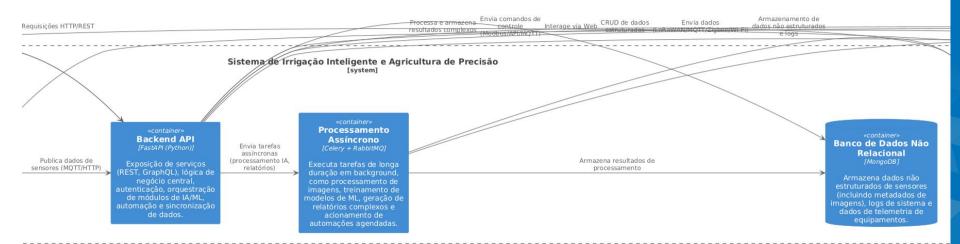


Diagrama de Contexto - Segunda Iteração

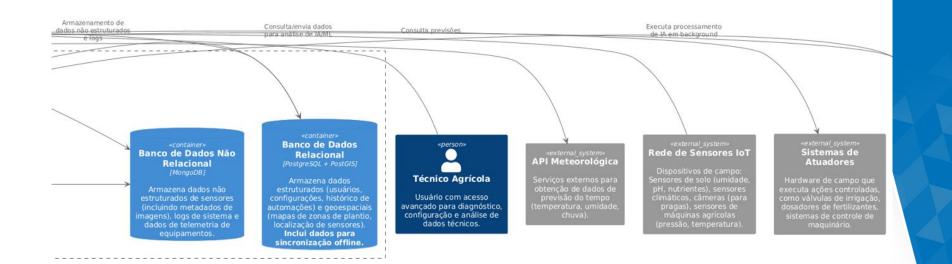














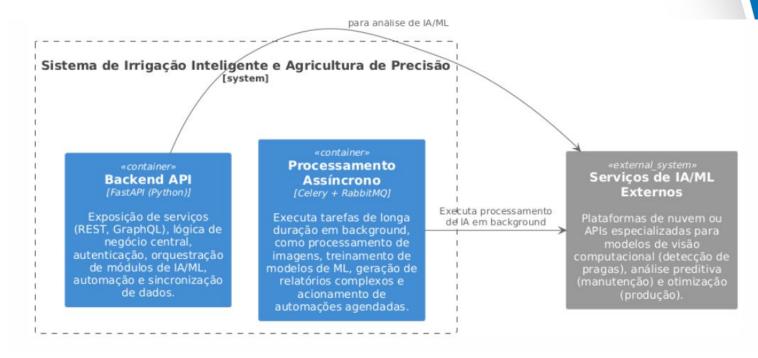


Diagrama de Contexto 2 - Relacionamentos

INSTITUTO DE INFORMÁTICA

Agricultor interage com aplicação web (frontend)

Técnico agrícola interage com **aplicação web** (frontend)

Agricultor monitora e recebe notificações pela aplicação mobile

Aplicação web interage com API via Requisições HTTP/REST

Aplicação mobile interage com API via Requisições HTTP/REST

API interage com banco de dados relacional por meio de um CRUD de dados estruturados

API interage com banco de dados não relacional para fazer armazenamento de dados não estruturados e logs

API interage com **worker** para enviar tarefas assíncronas (processamento IA, relatórios)

Worker interage com banco de dados relacional para processar e armazenar resultados complexos

Worker interage com banco de dados não relacional para armazenar resultados de processamento

Gateway IoT interage com **API** para publicar dados de sensores (MQTT/HTTP)

API interage com api meteorológica para consultar previsões e dados sobre o tempo

Sensores enviam dados (LoRaWAN/MQTT/Zigbee/Wi-Fi) para Gateway IoT

API interage com sistemas de atuadores para enviar comandos de controle (Modbus/API/MQTT)

API interage com os serviços externos de IA para consultar/enviar dados para análise de IA/ML

Worker interage com serviços externos de IA para executar o processamento de IA em background



Diagrama de Contêineres - Sistema de Monitoramento e Automação Agrícola Agricultor/Técnico plantação, configura automações e recebe alertas/recomendações Usa [HTTPS] Sistema de Monitoramento e Automação Agrícola [system] Aplicação Frontend interaçõe venmobile para interação com o usuário: dashboards, alertas, relatórios, recomendações e configurações. [HTTPS/JSON (JWT Auth)] // for JWT Aplicação Backend (API) [FastAPI (Python), Celery] Envia comandos de Recebe dados e envia Obtém dados Lê/Escreve controle comandos/configurações [MQTT/HTTPS] meteorológicos [HTTPS/JSON] [Modbus/API MQTT] // for Modbus APIs Meteorológicas Externas Banco de Dados Fornece dados de previsão do tempo (umidade, chuva seca). Ex: OpenWeatherMap, Meteostat. [LoRaWAN, Zigbee Wi-Fi] // for LoRa Sensores de Solo e Clima (IoT)

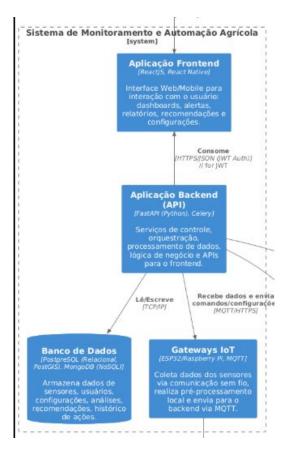






Diagrama de Contêineres - Sistema de Monitoramento e Automação Agrícola













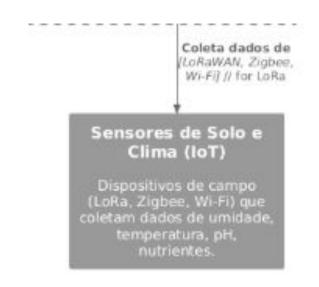
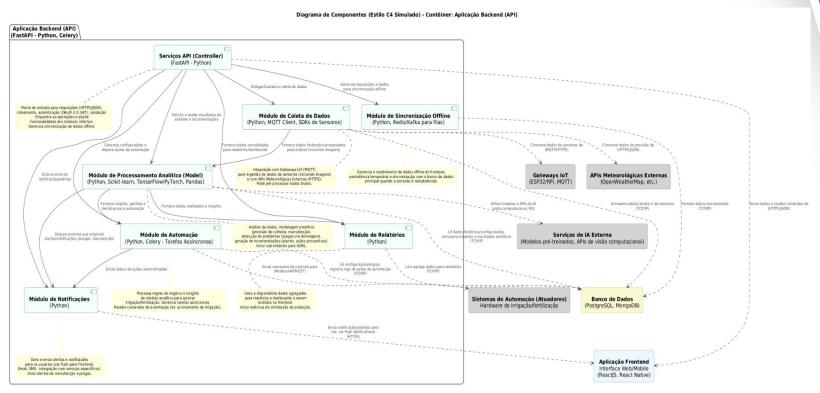




Diagrama de Componentes





funcionalidades dos módulos internos.

Gerencia sincronização de dados offline.

Aciona envio de

notificações/alertas

Módulo de Notificações

Gera e envia alertas e notificações para os usuários (via Push para Frontend, Email, SMS - integração com serviços específicos). Inclui alertas de manutenção e pragas.

Dispara eventos que originam

alertas/notificações (pragas, manutenção)

Gerencia configurações e

dispara ações de automação

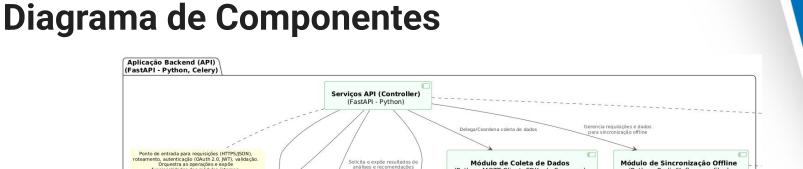
Fornece insights, gatilhos e

decisões para automação

Envia status de ações automatizadas

Módulo de Processamento Analítico (Model)

(Python, Scikit-learn, TensorFlow/PyTorch, Pandas)



Fornece dados analisados e insights

Módulo de Automação

(Python, Celery - Tarefas Assíncronas)

Processa regras de negócio e insights

do módulo analítico para acionar

irrigação/fertilização. Gerencia tarefas assíncronas.

Recebe comandos de automação (ex: acionamento de irrigação).

Fornece dados consolidados

para relatórios/dashboards

(Python, MQTT Client, SDKs de Sensores)

Integração com Gateways IoT (MOTT)

para ingestão de dados de sensores (incluindo imagens)

e com APIs Meteorológicas Externas (HTTPS)

Análise de dados, modelagem preditiva (previsão de colheita, manutenção),

detecção de problemas (pragas via IA/imagens),

geração de recomendações (plantio, ações preventivas).

Inclui sub-módulos para IA/ML.

Gera e disponibiliza dados agregados

para relatórios e dashboards a serem

exibidos no frontend. Inclui métricas de otimização de produção.

Envia comandos de controle para

(Modbus/API/MOTT)

ornece dados brutos/pré-processados

para análise (incluindo imagens)

(Python, Redis/Kafka para filas)

Gerencia o recebimento de dados offline do frontend,

persistência temporária e sincronização com o banco de dados

principal quando a conexão é restabelecida.

Módulo de Relatórios

(Python)

Lê configurações/regras,

registra logs de ações de automação

Envia notificações/alertas para (ex: via Push Notifications)

Lê da dos his

armazena mode

Lê e agrega dados para relatórios



INFORMÁTICA

Diagrama de Componentes



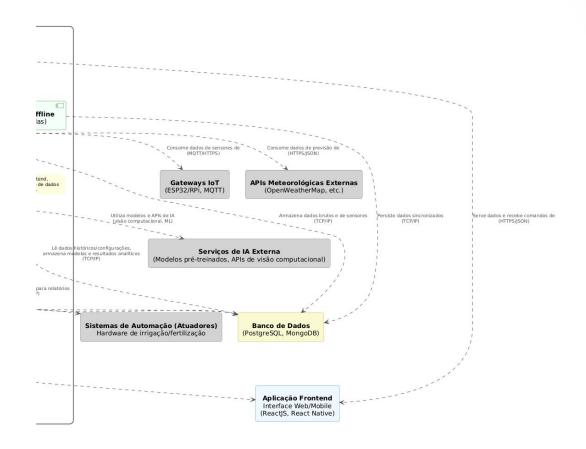
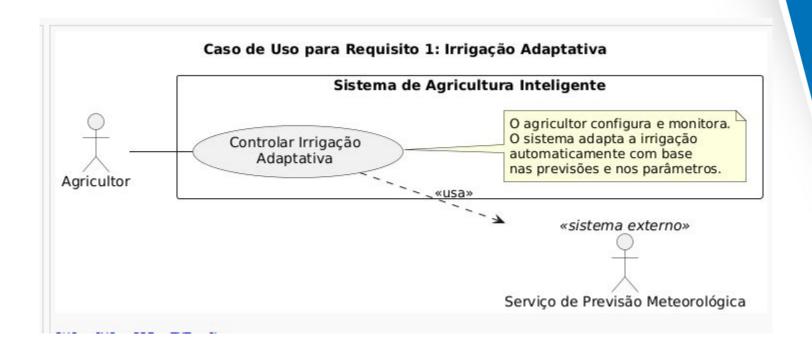


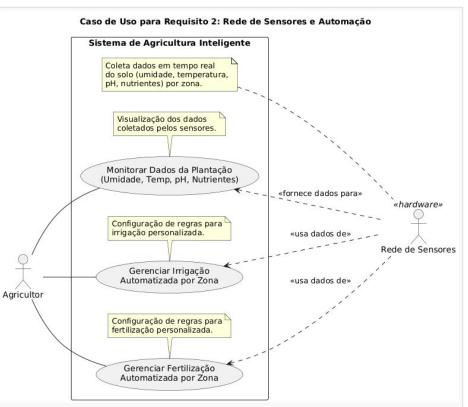
Diagrama de Casos de uso - Primeira Iteração

INF INSTITUTO DE INFORMÁTICA

O sistema deve controlar a irrigação de forma automatizada e adaptativa de acordo com solo e clima.



O sistema deve possuir uma rede de sensores para coleta de informações para parametrização e automação.



O sistema deve identificar problemas e gerenciar alertas





O sistema deve sugerir melhores opções de cultura.





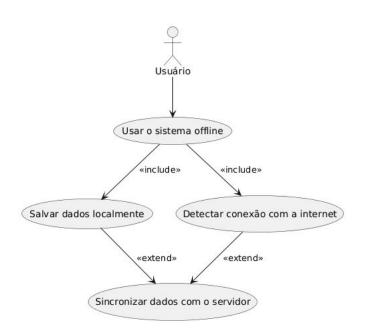
INSTITUTO DE INFORMÁTICA

O sistema deve prever o momento certo para ser realizada colheita.



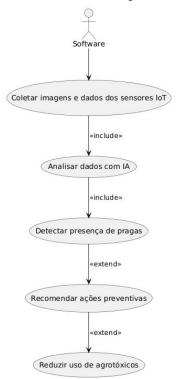
Diagrama de Casos de uso - Segunda Iteração

O sistema deve funcionar mesmo com conexão limitada, usando sincronização offline e transmissão dos dados quando a rede estiver disponível.



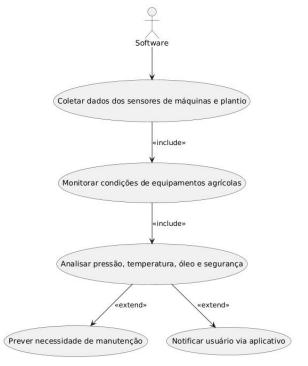
R07

O sistema deve utilizar câmeras e sensores IoT combinados com IA para identificar precocemente a previamente a presença de pragas na plantação, permitindo ações preventivas e reduzindo o uso de agrotóxicos.



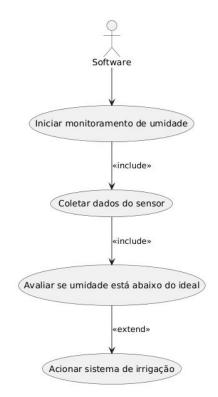
INF

O sistema deve implementar não só sensores para medir estatísticas de plantio e terra, mas também para monitorar equipamentos e máquinas agrícolas, capazes de prever manutenções e checagem de elementos de segurança, como: pressão de pneus, temperatura, qualidade do óleo e notificar tudo isso por meio do app.



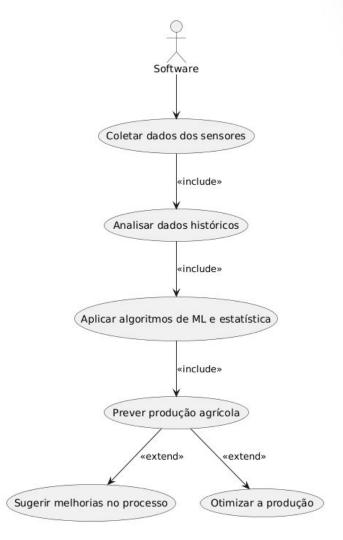
R09

O software deve acionar automaticamente o sistema de irrigação caso os os níveis de umidade do solo estiverem abaixo do ideal.





O sistema deve utilizar algoritmos de aprendizagem de máquina e estatística para prever e otimizar a produção. Dados históricos devem ser utilizados para sugerir melhorias no processo agrícola.







Sistema de Agricultura Inteligente com IoT - Possíveis tecnologias

Camada	Tecnologias
Infraestrutura	Sensores IoT, LoRaWAN, MQTT, Gateways ESP32/RPi
Persistência	MongoDB, PostgreSQL
Negócio (Model)	Python, scikit-learn, TensorFlow/PyTorch, pandas
Controle (Controller)	FastAPI, RabbitMQ
Apresentação (View)	ReactJS, React Native
Integração e Comunicação	OpenWeatherMap, Kafka (opcional)
Segurança	OAuth 2.0, JWT, TLS/SSL, Keycloak
DevOps e Deployment	Docker, Kubernetes, GitHub Actions, Prometheus, AWS/GCP/Azure



Sistema de Agricultura Inteligente com IoT

O trabalho inquestionavelmente nos trouxe uma maior visão sobre o processo de elaboração/documentação da Arquitetura de Software, ao passo que desenhamos, a partir dos requisitos, toda a visão arquitetural do sistema. Desde a concepção do estilo, dos padrões até as decisões sobre os modelos de representação da arquitetura. Além disso, ampliou nossos conhecimentos no domínio do IoT e da agricultura.

Ferramentas utilizadas:

- https://plantuml.com/
- PlantUML Web Server

Obrigado! Fim!!



