

**Documentação de um**

**Produto de Software**

**Sistema Integrado de Monitoramento Agro - S.I.M.A**

**Nome dos Alunos:**

**Ana Júlia – 324152342**

**Bruno Okazava dos Santos – 1072012296**

**Eduardo Schutz de Souza - 1072021061**

**Guilherme Sendim – 12624118253**

**João Carlos de Farias Almeida -12724114097**

**Luiza Cavalcante Quina Siqueira – 824118304**

**Mastrangelo Reino – 823224169**

**Mauro Braga – 822231929**

**Nicolai Freitas – 1271818762**

**2024**

**ÍNDICE DETALHADO**

**1.** **Introdução 3**

1.1. Tema 3

1.2. Objetivos a serem alcançados 3

1.3. Escopo principal 3

**2.** **Definição do Modelo de Processo 4**

**3.** **Requisitos do Sistema de Software 4**

3.1. Requisitos Funcionais 4

3.2. Requisitos Não-Funcionais 4

**4.** **Projeto 5**

4.1. Arquitetura Lógica 5

4.2. Arquitetura Física 5

**5.** **Protótipo de Interface 6**

**6.** **Critérios de Qualidade de Software 7**

**7. Testes 8**

7.1. Plano de Testes 8

7.2. Roteiro de Testes 8

1. **Introdução**
   1. **Tema**

Desenvolvimento de um software para monitoramento e controle de condições climáticas em lavouras, utilizando drones e sensores de IoT. O programa é capaz de coletar dados de sensores de temperatura e umidade, bem como de sensores de pH do solo capacitivo, por meio de placas de desenvolvimento conectadas pelo protocolo MQTT. Além disso, drones de asa fixa são utilizados para mapear grandes áreas em um curto espaço de tempo.

Para o processamento dos dados, é utilizado um sistema de nuvem (AWS) e um sistema de backend que captura os dados processados, criando uma API para que o frontend possa gerar relatórios. Com essas informações, o software é capaz de monitorar as condições ambientais das fazendas e fornecer insights para ajudar na tomada de decisões visando melhorar a produção agrícola.

A solução oferecida permite que os produtores rurais tomem decisões mais informadas e precisas sobre o manejo do solo, o uso de insumos, a irrigação e a proteção de cultivos, garantindo maior eficiência e produtividade.

* 1. **Objetivos a serem alcançados**
* O público-alvo deste documento são os desenvolvedores e testadores envolvidos no projeto de sensoriamento de lavoura, bem como o cliente e os usuários finais - tais como agricultores e donos de lavouras, por exemplo - que irão utilizar o software.
* Coleta de dados precisos e confiáveis dos sensores de temperatura, umidade e pH do solo, bem como do mapeamento de grandes áreas por meio de drones de asa fixa.
* Processamento dos dados coletados de forma eficiente, utilizando um sistema de nuvem (AWS) para armazenamento e processamento dos dados e um sistema de backend para captura dos dados processados e criação de uma API para o frontend.
* Geração de relatórios claros e precisos com base nos dados coletados e processados, para que os usuários possam tomar decisões informadas sobre suas operações agrícolas.
* Monitoramento contínuo das condições ambientais das fazendas, permitindo que os usuários identifiquem tendências e problemas antes que se tornem críticos.
* Facilidade de uso do software, permitindo que os usuários acessem e interpretem facilmente as informações coletadas e geradas pelo sistema.
* Customização do software para atender às necessidades específicas dos usuários, permitindo que eles selecionem quais sensores utilizar e como as informações são apresentadas nos relatórios.
* Integração com outros sistemas de gerenciamento de fazendas, permitindo que os usuários acessem informações adicionais para melhorar suas operações agrícolas.
* Melhoria contínua do software com base no feedback dos usuários, garantindo que o sistema esteja sempre atualizado e atenda às necessidades em constante mudança dos usuários.
  1. **Escopo principal**

O escopo principal deste software é o de coletar dados de sensores e drones em tempo real e armazená-los em uma nuvem (AWS). O sistema deve ser capaz de processar esses dados e fornecer informações relevantes sobre a temperatura e umidade do ar, o pH do solo, além de permitir o mapeamento de grandes áreas com drones de asa fixa. O sistema deve ser capaz de lidar com múltiplos dispositivos IoT (placas de desenvolvimento) e sensores simultaneamente e fornecer uma interface de programação de aplicativos (API) para permitir que um front-end gerencie e apresente os dados em relatórios e gráficos para os usuários finais. O backend também deve ser capaz de detectar e tratar falhas e enviar alertas aos usuários em caso de problemas.

1. **Definição do Modelo de Processo**

* Requisitos: Na primeira etapa, definiremos os requisitos do software, identificando quais informações precisam ser coletadas pelos sensores e drones, quais protocolos de comunicação seriam utilizados, quais funcionalidades o software teria e como seria a interface com o usuário.
* Projeto: Na segunda etapa, projetaríamos a arquitetura do software e definiremos os componentes do sistema, incluindo o sistema de nuvem e o sistema de backend. Nesta fase, também seriam definidas as tecnologias a serem utilizadas.
* Implementação: Na terceira etapa, iniciaremos a implementação do software, desenvolvendo os módulos do sistema, integrando os sensores e drones, implementando o sistema de nuvem e o sistema de backend.
* Testes: Na quarta etapa, realizamos os testes de unidade, integração e sistema para garantir que o software esteja funcionando de acordo com os requisitos e especificações.
* Manutenção: Na quinta etapa, manteríamos o software em funcionamento, corrigindo bugs, adicionando novas funcionalidades e melhorando o desempenho do sistema.

Durante todo o processo de desenvolvimento, seguiríamos a metodologia SCRUM, dividindo o desenvolvimento em sprints de duas semanas, com entregas incrementais de funcionalidades. As reuniões diárias seriam realizadas para discutir o progresso do trabalho e identificar quaisquer problemas ou obstáculos que possam estar impedindo o progresso. Ao final de cada sprint, seriam realizadas reuniões de revisão para discutir os resultados alcançados e planejar o próximo sprint.

1. **Requisitos do Sistema de Software**
   1. **Requisitos Funcionais**

* Coleta de dados dos sensores de temperatura e umidade, com leitura em intervalos regulares de tempo;
* Coleta de dados do sensor de pH do solo capacitivo, com leitura em intervalos regulares de tempo;
* Coleta de dados dos drones de asa fixa, com informações sobre a localização, altitude e velocidade;
* Processamento dos dados coletados pelos sensores e drones para gerar informações úteis para o usuário, como mapas de calor e relatórios de produtividade;
* Armazenamento dos dados coletados em um banco de dados seguro e escalável, como o Amazon S3;
* Criação de uma API para o front-end acessar os dados processados e apresentá-los de forma visualmente agradável e intuitiva;
* Integração com outros sistemas e dispositivos, como sistemas de irrigação e fertilização, para otimizar o uso de recursos e aumentar a produtividade da fazenda.
  1. **Requisitos Não-Funcionais**
* Confiabilidade: o sistema deve ter alta disponibilidade e confiabilidade para garantir que os dados coletados sejam precisos e confiáveis.
* Segurança: o sistema deve ser seguro para evitar o acesso não autorizado aos dados coletados e garantir a privacidade dos usuários.
* Desempenho: o sistema deve ser capaz de processar grandes quantidades de dados rapidamente e sem interrupções, de forma a permitir a análise em tempo real.
* Escalabilidade: o sistema deve ser capaz de lidar com um grande número de dispositivos de coleta de dados e ser escalável para suportar o crescimento futuro.
* Usabilidade: o sistema deve ser fácil de usar e intuitivo para permitir que os usuários gerenciem os dispositivos e visualizem os dados coletados de forma eficiente.
* Manutenção: o sistema deve ser fácil de manter e atualizar, de forma a garantir sua confiabilidade e disponibilidade a longo prazo.

1. **Projeto**
   1. **Arquitetura Lógica**

Coleta de dados:

* Os sensores e drones enviam dados em tempo real usando o protocolo MQTT.
* Os dados são recebidos e armazenados em um banco de dados NoSQL, como o Amazon DynamoDB, para permitir a escalabilidade horizontal.
* É importante garantir a segurança dos dados durante a transferência e armazenamento, utilizando criptografia SSL/TLS e autenticação.

Processamento de dados:

* O Amazon Kinesis é utilizado para processar em tempo real os dados recebidos pelos sensores e drones, agregando e analisando os dados antes de serem armazenados no banco de dados.
* Algoritmos de machine learning podem ser utilizados para prever comportamentos e alertas em caso de anomalias.

Backend:

* O Amazon API Gateway é utilizado para criar uma API que expõe os dados processados e armazenados no banco de dados para serem consumidos pelo frontend.
* O Amazon Lambda é utilizado para executar o código de backend, processando as solicitações da API..

Frontend:

* A interface do usuário é desenvolvida em uma aplicação web, utilizando tecnologias como React.
* A aplicação permite a visualização dos dados coletados pelos sensores e drones, bem como a geração de relatórios e gráficos interativos.
* O Amazon CloudFront é utilizado para a distribuição de conteúdo estático para os usuários, melhorando a velocidade de acesso.
  1. **Arquitetura Física**
* Implantação dos sensores e placas de desenvolvimento em locais estratégicos, garantindo que os dados coletados sejam representativos do ambiente monitorado.
* Instalação de antenas de comunicação MQTT em pontos estratégicos para garantir que todos os dados coletados pelos sensores sejam enviados para o servidor.
* Implantação de pontos de coleta de dados para os drones, permitindo que as informações sejam transferidas para o servidor de forma eficiente e rápida.
* Utilização de servidores redundantes em diferentes regiões geográficas para garantir que o software tenha alta disponibilidade e possa continuar funcionando mesmo em caso de falhas em um dos servidores.
* Garantir a segurança dos dados coletados através do uso de criptografia, autenticação e autorização adequadas, bem como backups regulares para evitar perda de dados em caso de falhas de hardware ou software.

1. **Protótipo de Interface**

* Identificar os stakeholders envolvidos no projeto: para garantir que todos os interessados no projeto sejam ouvidos e suas necessidades sejam consideradas, é importante identificar todos os stakeholders envolvidos, incluindo os usuários finais, os donos de fazendas, os desenvolvedores de drones e especialistas em agricultura.
* Levantamento de requisitos: O próximo passo é realizar a coleta de requisitos através de entrevistas com os stakeholders e análise de documentação de referência. Neste processo, é importante definir as necessidades do usuário, os principais recursos e funcionalidades do sistema, as limitações técnicas e de negócios, e quaisquer outras considerações importantes que possam afetar o desenvolvimento do software.
* Detalhamento de requisitos: Com base na lista de requisitos coletados, o próximo passo é detalhar esses requisitos e definir as especificações detalhadas do sistema. Esta fase também inclui a identificação de casos de uso, definição de fluxos de trabalho e a elaboração de protótipos.
* Validação de requisitos: Nesta etapa, os requisitos detalhados são validados com os stakeholders para garantir que eles sejam completos, corretos e atendam às necessidades do usuário. Este processo envolve revisões detalhadas, testes de aceitação e feedback de usuários e especialistas em agricultura.
* Modelagem de interface do usuário: A partir dos requisitos validados, é hora de projetar a interface do usuário. Esta fase inclui a criação de esboços, wireframes e protótipos interativos para ajudar a visualizar e avaliar a experiência do usuário.
* Validação da interface do usuário: A última etapa é validar a interface do usuário com os usuários finais, para garantir que ela seja intuitiva, fácil de usar e atenda às suas necessidades.

1. **Critérios de Qualidade de Software**

* Confiabilidade: a confiabilidade do sistema é essencial para garantir que os dados coletados pelos sensores e drones sejam precisos e confiáveis, sem perdas ou falhas durante o processo de coleta e processamento.
* Segurança: como o sistema lida com informações sensíveis, como dados de medição de temperatura e umidade, é importante que ele seja seguro e protegido contra possíveis ataques externos e vazamentos de dados.
* Desempenho: o sistema deve ser capaz de processar grandes quantidades de dados gerados pelos sensores e drones em tempo hábil para que os relatórios sejam gerados de forma rápida e eficiente.
* Manutenibilidade: como o sistema é composto por diferentes componentes, é importante que ele seja fácil de manter e atualizar, evitando possíveis problemas de compatibilidade e facilitando a correção de bugs.
* Usabilidade: o sistema deve ser fácil de usar para o usuário final, com uma interface intuitiva e clara que permita a visualização dos dados coletados e gerados pelos relatórios de forma simples e acessível.

**7. Testes**

**7.1. Plano de Testes**

* Teste de integração
* Teste de unidade
* Teste de sistema
* Teste de usabilidade
* Teste de desempenho
* Teste de segurança
* Teste de regressão
* teste de aceitação

**7.2. Roteiro de Testes**

1. Teste de configuração:
   * Verificar se o sensor de drone está devidamente configurado;
   * Verificar se os sensores de temperatura e umidade, e o sensor de pH do solo capacitivo estão devidamente conectados;
   * Verificar se o sistema de nuvem (AWS) está devidamente configurado;
   * Verificar se o sistema de backend está devidamente configurado;
   * Verificar se a API do sistema de backend está devidamente configurada;
   * Verificar se o front-end está devidamente configurado.
2. Teste de coleta de dados:
   * Verificar se os sensores de temperatura e umidade estão coletando os dados corretamente;
   * Verificar se o sensor de pH do solo capacitivo está coletando os dados corretamente;
   * Verificar se o drone está capturando as imagens corretamente;
   * Verificar se o sistema de coleta de dados está enviando os dados corretamente para o sistema de nuvem (AWS);
   * Verificar se o sistema de nuvem (AWS) está armazenando os dados corretamente.
3. Teste de processamento de dados:
   * Verificar se o sistema de processamento de dados está processando os dados corretamente;
   * Verificar se o sistema de backend está recebendo os dados corretamente do sistema de nuvem (AWS);
   * Verificar se o sistema de backend está processando os dados corretamente;
   * Verificar se o sistema de backend está armazenando os dados corretamente.
4. Teste de geração de relatórios:
   * Verificar se o front-end está recebendo os dados corretamente do sistema de backend;
   * Verificar se o front-end está gerando os relatórios corretamente;
   * Verificar se os relatórios gerados estão apresentando as informações corretas e completas.
5. Teste de integração:
   * Verificar se todos os sistemas estão integrados corretamente;
   * Verificar se os dados estão sendo transmitidos corretamente entre os sistemas;
   * Verificar se o fluxo de dados entre os sistemas está correto;
   * Verificar se as interfaces entre os sistemas estão funcionando corretamente.
6. Teste de desempenho:
   * Verificar se o sistema está funcionando corretamente em diferentes condições de uso, como com diferentes volumes de dados;
   * Verificar se o sistema está respondendo rapidamente aos comandos do usuário;
   * Verificar se o sistema está executando as tarefas de forma eficiente e sem erros;
   * Verificar se o sistema está usando os recursos de hardware de forma eficiente.