



UNIVERSIDADE FEDERAL DE GOIÁS (UFG)

INSTITUTO DE INFORMÁTICA - SEMESTRE SELETIVO 2024/2

CURSO: ENGENHARIA DE SOFTWARE

DISCIPLINA: PADRÕES DE ARQUITETURA DE SOFTWARE

PROFESSOR: JACSON RODRIGUES BARBOSA

Acadêmicos:

Janderson Oliveira da Silva

Tayná Crisllen José de Santana

Monitoramento de Chuvas e Prevenção de Enchentes

AquaGuard

Gyn

1. Introdução

1.1 Finalidade

Este documento tem como objetivo detalhar a arquitetura de software do Sistema de Monitoramento de Chuvas e Prevenção de Enchentes (AquaGuard Gyn), direcionado aos stakeholders, incluindo gerentes de projeto, clientes, desenvolvedores e equipes de implantação. O foco principal é prover uma base para o desenvolvimento e implantação de uma solução eficaz e escalável.

1.2 Escopo

O AquaGuard Gyn utiliza uma arquitetura distribuída para monitorar dados climáticos em tempo real, analisá-los e emitir alertas de enchentes. Os componentes do sistema incluem sensores climáticos IoT, servidores de processamento, armazenamento de dados na nuvem e interfaces de usuário responsivas para visualização e notificação.

1.3 Definições, Acrônimos e Abreviações

- **SMCPE:** Sistema de Monitoramento de Chuvas e Prevenção de Enchentes.
- **IoT:** Internet das Coisas.
- **REST:** Representational State Transfer.
- **API:** Interface de Programação de Aplicativos.
- **MQTT:** Protocolo de Telemetria de Fila de Mensagens.

1.4 Referências

- ISO/IEC 9126: Qualidade de software, para definição de atributos de qualidade como manutenibilidade, escalabilidade e confiabilidade.

- ISO/IEC 27001: Segurança da informação, assegurando proteção contra acessos não autorizados.
- ISO 19156: Observações e medições, aplicável a sensores e dados ambientais.
- ISO 25010: Modelos de qualidade de software e sistemas.

2. Contexto da Arquitetura

2.1 Funcionalidades e Restrições Arquiteturais

- **Funcionalidades:**
 - Monitoramento de dados climáticos em tempo real (temperatura, umidade, nível de chuva e de rios).
 - Processamento de dados para previsão de enchentes.
 - Emissão de alertas através de SMS, e-mail e notificações push.
 - Armazenamento histórico de dados para análise.
- **Restrições:**
 - Alta disponibilidade e tolerância a falhas.
 - Segurança de dados sensíveis.
 - Conformidade com padrões de interoperabilidade IoT (como MQTT).

2.2. Requisitos Funcionais

- **Monitoramento de Nível de Água:**
 - O sistema irá monitorar em tempo real os níveis de água em pontos críticos (rios, canais, lagos) usando sensores de medição (como sensores de pressão ou câmeras de monitoramento).
- **Previsão Meteorológica:**
 - O sistema deverá integrar-se a serviços meteorológicos para monitorar condições climáticas como chuva intensa, tempestades ou enchentes previstas, que podem contribuir para o alagamento.
- **Análise de Dados em Tempo Real:**

- O sistema realizará a análise contínua dos dados de sensores de nível de água e previsão meteorológica para detectar padrões e alertar sobre possíveis riscos de alagamento.
- **Geração de Alertas:**
 - O sistema deverá gerar alertas automatizados para áreas de risco, com base em critérios de nível de água ou previsão de chuva. Esses alertas devem ser enviados via notificações em dispositivos móveis (SMS, e-mail, app) e sistemas de sirenes ou alto-falantes públicos.
- **Mapeamento de Áreas de Risco:**
 - Exibição em tempo real das áreas monitoradas e mapeamento de zonas de risco de alagamento, integrando-se com mapas geográficos (como Google Maps ou sistemas GIS) para indicar áreas críticas.
- **Alertas Personalizados:**
 - O sistema deverá permitir que as autoridades ou usuários definam parâmetros personalizados de alerta (por exemplo, alertar para uma determinada área quando o nível de água atingir 3 metros ou quando a previsão de chuva ultrapassar X mm).
- **Notificação para Autoridades e Defesa Civil:**
 - O sistema deverá enviar notificações diretamente para os responsáveis pela gestão de desastres naturais, como equipes da Defesa Civil, bombeiros, e autoridades locais.
- **Relatórios e Histórico:**
 - O sistema permitirá gerar relatórios históricos sobre eventos passados de alagamento, com dados sobre o nível de água, previsão meteorológica e eficácia dos alertas.
- **Interface de Usuário (UI):**
 - A interface do sistema deverá ser simples e intuitiva, com dashboards para visualização rápida do status de monitoramento e alertas recebidos.

2.3 Requisitos Não Funcionais

- **Desempenho:** O sistema deve ser capaz de processar e analisar dados em tempo real, com baixa latência, para fornecer alertas oportunos.
- **Escalabilidade:** O sistema deve ser capaz de expandir para cobrir novas áreas ou aumentar a quantidade de sensores monitorados sem perda significativa de desempenho.

- **Confiabilidade:** O sistema deve ser altamente confiável, com backup e redundância para garantir que os dados e alertas sejam sempre precisos e entregues no tempo certo.
- **Usabilidade:** A interface de usuário deve ser acessível e fácil de usar, com funcionalidades de monitoramento e alerta acessíveis para diferentes tipos de usuários (autoridades, cidadãos, equipe de emergência).
- **Segurança:** O sistema deve garantir a segurança dos dados, especialmente os relacionados a informações sensíveis, como os dados de localização e alerta.

2.4 Atributos de Qualidade Prioritários

- **Escalabilidade:** Para lidar com o crescimento do número de sensores e usuários.
- **Confiabilidade:** Garantir a entrega de alertas sem falhas.
- **Segurança:** Proteção contra acessos não autorizados e manipulação de dados.
- **Manutenibilidade:** Uso de arquitetura modular para facilitar atualizações.

3. Representação da Arquitetura

3.1 Visão Geral

A arquitetura do AquaGuard Gyn é composta por quatro camadas principais:

1. Camada de Coleta de Dados:

- Sensores IoT distribuem dados em tempo real.
- Comunicação via protocolo MQTT.

2. Camada de Processamento:

- Servidores de processamento realizam análise de dados.
- Uso de algoritmos de machine learning para previsão de enchentes.

3. Camada de Armazenamento:

- Banco de dados na nuvem para armazenamento de dados históricos.
- Suporte a consultas em tempo real e análise histórica.

4. Camada de Aplicação:

- Interfaces de usuário (web e mobile) para visualização de dados.

- APIs REST para integração com sistemas externos.

3.2 Estilos Arquiteturais

- Arquitetura baseada em microsserviços para maior flexibilidade e escalabilidade.
- Uso de padrões REST para comunicação entre componentes.
- Implementação de filas de mensagens para processar dados assíncronos.

4. Casos de Uso

4.1 Identificação dos Casos de Uso

Atores:

- Usuário (Cidadão)
- Sistema de Monitoramento de Chuvas
- Administração Municipal
- Equipe de Emergência
- Serviços de Meteorologia
- Sensores de Chuvas
- Banco de Dados (Armazenamento de Dados)
- Casos de uso:

Usuário (Cidadão):

- Consultar Previsões de Chuvas
- Receber Alertas de Enchentes
- Acompanhar Histórico de Chuvas

Sistema de Monitoramento de Chuvas:

- Coletar Dados de Sensores de Chuvas
- Analisar Dados de Precipitação
- Gerar Alertas em Caso de Risco de Enchente

Administração Municipal:

- Visualizar Dados de Precipitação e Alertas
- Planejar Ações de Prevenção de Enchentes
- Enviar alertas para a população

Equipe de Emergência:

- Monitorar Alertas de Enchentes
- Atuar nas Áreas Afetadas
- Receber Informações em Tempo Real

Serviços de Meteorologia:

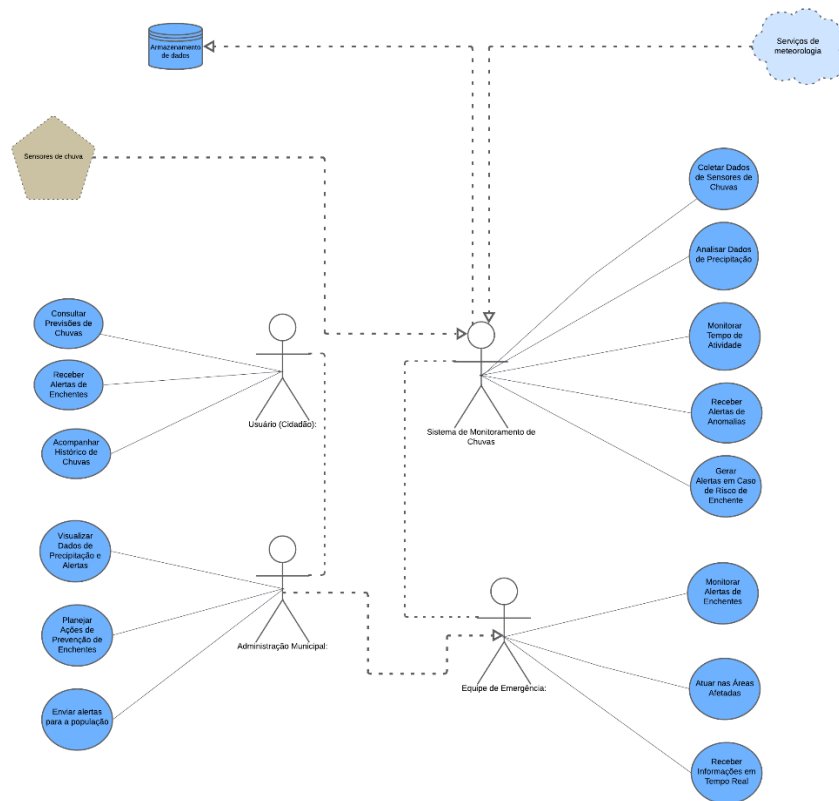
- Fornecer Dados Meteorológicos
- Validar Pré-selecionadores de Chuvas

Sensores de Chuvas:

- Captar Níveis de Precipitação
- Enviar Dados ao Sistema de Monitoramento

Banco de Dados (Armazenamento de Dados):

- Armazenar Dados de Precipitação
- Registrar Alertas Gerados



5. Detalhamento dos Componentes

5.1 Sensores Climáticos

- Dispositivos IoT para coleta de dados.
- Envia leituras em intervalos regulares para o servidor central via MQTT.

5.2 Servidores de Processamento

- Realizam análises em tempo real.
- Gerenciam previsões e geram alertas baseados em dados recebidos.

5.3 Banco de Dados

- Armazenamento histórico e acesso rápido a dados.
- Uso de banco relacional (ex: PostgreSQL) e NoSQL (ex: MongoDB).

5.4 Interface de Usuário

- Aplicativos responsivos para acesso a relatórios e alertas.

- Opção de exportar dados para outros sistemas

6. Tecnologias Utilizadas

- **Linguagens:** Python para processamento e JavaScript para interfaces.
- **Protocolos:** MQTT para coleta e REST para APIs.
- **Plataformas:** AWS para hospedagem e processamento.
- **Bibliotecas:** Pandas e Scikit-learn para análise de dados.

7. Limitações do Sistema

- **Cobertura de Sensores:** O sistema pode ter limitações quanto à cobertura dos sensores de nível de água, dependendo da infraestrutura disponível.
- **Conectividade:** O sistema depende de uma boa infraestrutura de comunicação (internet, redes móveis) para envio de alertas e monitoramento remoto, o que pode ser um desafio em áreas remotas.
- **Precisão das Previsões:** O sistema pode estar sujeito a limitações na precisão das previsões meteorológicas e dos dados de sensores.

8. Requisitos de Interface

- **Sensores e Dispositivos:** O sistema precisará se integrar com sensores de nível de água, estações meteorológicas e outros dispositivos de monitoramento.
- **Interface de Mapeamento:** Integração com ferramentas de mapeamento (GIS, Google Maps) para exibição das áreas monitoradas e análise geográfica dos alagamentos.
- **Integração com Sistemas de Notificação:** O sistema deve ser capaz de enviar alertas via SMS, e-mail e outros canais, como aplicativos móveis ou sistemas de sirenes.

9. Critérios de Aceitação

- O sistema será considerado aceito se:
 - Os alertas de alagamento forem gerados corretamente com base em dados de sensores e previsões meteorológicas.
 - As notificações forem enviadas corretamente a autoridades e cidadãos.
 - A interface do usuário for intuitiva e funcional.

- O sistema estiver disponível e operando com alta confiabilidade durante os períodos críticos de risco.
-

10. Prazos e Marcos

- Fase de Pesquisa e Planejamento: 2 meses.
- Desenvolvimento da Plataforma de Monitoramento e Sensores: 6 meses.
- Desenvolvimento de Interface de Usuário e Integração com Mapas: 3 meses.
- Testes e Validação: 2 meses.
- Implementação e Lançamento: 1 mês.

11. Recursos Necessários

- **Infraestrutura de Sensores:** Sensores de nível de água, estações meteorológicas e dispositivos de comunicação.
- **Profissionais:** Desenvolvedores de software, especialistas em sistemas de monitoramento ambiental, designers de UI/UX, e engenheiros de infraestrutura.
- **Orçamento Estimado:** O custo total dependerá da quantidade de áreas e sensores a serem cobertos e da complexidade da integração dos sistemas de alerta.

12. Exclusões do Escopo

- **Monitoramento de Outros Tipos de Desastres Naturais:** O sistema não incluirá a monitorização de desastres como terremotos ou incêndios.
- **Cobertura Global:** O sistema será inicialmente focado em áreas específicas, com possibilidade de expansão futura.

13. Riscos Identificados

- **Falhas de Sensor:** Possíveis falhas nos sensores de monitoramento de nível de água, que podem afetar a precisão dos dados.
- **Conectividade em Áreas Remotas:** A falta de infraestrutura de comunicação em áreas remotas pode afetar a entrega de alertas em tempo real.
- **Previsões Meteorológicas Inadequadas:** A dependência de dados meteorológicos pode ser afetada por imprecisões nas previsões.

14. Plano de Implementação e Manutenção

- **Implantação Inicial:** O sistema será inicialmente implantado em áreas críticas e monitoradas de forma contínua.
- **Manutenção e Atualizações:** O sistema terá atualizações periódicas, além de manutenção preventiva para garantir a eficiência do monitoramento e dos alertas.

15. Conclusão

- Esta arquitetura busca atender aos requisitos funcionais e não funcionais do AquaGuard Gyn, garantindo um sistema escalável, seguro e de fácil manutenção para monitoramento de chuvas e prevenção de enchentes.