



**Universidade Federal de Goiás
Instituto de Informática
Trabalho Padrões de Arquitetura de Software**

MediAr

Documento de Arquitetura de Software

**André Lopes, 201703739
Rogério Rodrigues Rocha, 202203530**

**Goiânia
Outubro de 2024**

1. Introdução

1.1 Finalidade

O objetivo deste documento é descrever a arquitetura do sistema de monitoramento de poluição urbana MediAr. Ele é destinado às partes interessadas, incluindo desenvolvedores, gerentes de projeto e a equipe de implantação. O foco deste documento é garantir que a arquitetura atenda aos requisitos funcionais e não funcionais, proporcionando uma base sólida para o desenvolvimento e a manutenção do sistema.

1.2 Escopo

Este documento abrange os requisitos funcionais e não funcionais, detalhando os aspectos arquiteturais, os padrões utilizados, e as visões da arquitetura que serão implementadas no sistema. O sistema permitirá o monitoramento da qualidade do ar em tempo real, consultas históricas e comparações de dados de poluição, bem como informações educativas sobre os impactos ambientais.

1.3 Definições, Acrônimos e Abreviações

Aplicativo: Software destinado ao monitoramento de poluição.

Poluição Urbana: Qualidade do ar medida em áreas urbanas.

Stakeholder: Indivíduo ou grupo interessado no sistema (usuários, desenvolvedores, gerentes de projeto).

Visão Arquitetural: Representação de aspectos da arquitetura do sistema para um público específico.

Ponto de Vista Arquitetural: Abordagem utilizada para descrever e interpretar uma visão arquitetural.

Cliente-Servidor: Modelo de arquitetura de software em que o cliente faz requisições ao servidor, que processa e retorna as respostas. O cliente é a interface com o usuário, enquanto o servidor gerencia a lógica de negócio e o armazenamento de dados.

REST: Estilo de arquitetura para comunicação entre sistemas distribuídos. Utiliza métodos HTTP (GET, POST, PUT, DELETE) e recursos identificados por URLs para troca de dados entre clientes e servidores de forma simples e escalável.

MVC (Model-View-Controller): Padrão de arquitetura que separa a aplicação em três componentes principais: Model (responsável pela lógica e dados), View (interface com o usuário) e Controller (responsável pela comunicação entre Model e View). Esse padrão facilita a organização e manutenção de sistemas.

Frontend (Cliente): Parte da aplicação que interage diretamente com o usuário, geralmente composta por interfaces gráficas desenvolvidas com tecnologias como HTML, CSS e JavaScript. O frontend é responsável por enviar requisições ao backend e exibir os resultados.

Backend (Servidor): Parte da aplicação responsável pelo processamento de dados, lógica de negócios e integração com bancos de dados e APIs. O backend lida com as requisições do frontend e retorna as respostas adequadas, sendo executado no servidor.

Banco de Dados: Sistema de armazenamento e gerenciamento de dados estruturados ou não estruturados. Ele é utilizado pela aplicação para armazenar informações de maneira persistente e fornecer dados ao backend conforme necessário.

API: Conjunto de definições e protocolos que permite a comunicação entre diferentes sistemas. As APIs facilitam a integração de funcionalidades entre o frontend e o backend, bem como entre diferentes serviços.

1.4 Referências

ISO/IEC/IEEE 42010 - Normas para Descrição Arquitetural.

ISO/IEC 9126 - Normas para Qualidade de Software.

4+1 View Model - Modelo de representação arquitetural.

1.5 Visão Geral

O documento detalha os requisitos e restrições, os atributos de qualidade priorizados (segurança, usabilidade, escalabilidade), e os padrões arquiteturais aplicados (Cliente-Servidor, REST e MVC). As visões arquiteturais são descritas para comunicar a estrutura do sistema.

2. Contexto da Arquitetura

2.1 Funcionalidades e Restrições Arquiteturais

O sistema será estruturado de forma a atender aos seguintes Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF):

RF01 – Monitorar a qualidade do ar em tempo real

RF02 – Permitir consulta ao histórico de poluição

RF03 - Exibir dados de poluição de maneira gráfica

RF04 – Permitir filtragem de dados por data e localização

RF05 – Gerar relatório comparativo de poluição

RF06 - Página informativa sobre os impactos da poluição

RNF01 - Disponibilidade

RNF02 - Desempenho

RNF03 - Segurança de Dados

RNF04 - Escalabilidade

RNF05 - Compatibilidade com navegadores

RNF06 - Usabilidade

2.2 Atributos de Qualidade Prioritários

Os principais atributos de qualidade priorizados são:

Segurança: Implementação de camadas de segurança para proteger a privacidade dos dados dos usuários.

Usabilidade: Interface amigável e acessível, garantindo uma experiência intuitiva para usuários com pouca experiência.

Escalabilidade: Estrutura modular para suportar crescimento no número de usuários e manutenibilidade do sistema.

3. Representação da Arquitetura

3.1 Padrões Arquiteturais Adotados

O sistema utilizará uma arquitetura híbrida composta por:

Cliente-Servidor: Para a comunicação entre o frontend (cliente) e o backend (servidor).

REST: Para definição dos métodos de comunicação entre cliente e servidor, permitindo interação leve e eficiente.

MVC (Model-View-Controller): Para organizar as funcionalidades do sistema e manter a separação entre lógica de negócios, interface e dados.

3.2 Componentes Principais

Os principais componentes do sistema são:

Frontend (Cliente): Responsável por exibir a interface e interagir com o usuário. Será implementado em HTML/CSS e JavaScript, utilizando bibliotecas gráficas como Chart.js para exibir os gráficos de poluição.

Backend (Servidor): Implementado em Node.js com Express.js, será responsável por tratar as requisições do frontend e acessar os dados de poluição.

Banco de Dados: Utilização de um banco de dados relacional (PostgreSQL) para armazenar os dados históricos e em tempo real.

API Externa: Conectada ao sistema para obter dados de poluição em tempo real, como o AQICN (Air Quality Index China).

4. Visões Arquiteturais

4.1 Visão Lógica

4.2 Visão de Desenvolvimento

4.3 Visão Física

4.4 Visão de Segurança

5. Decisões Arquiteturais

5.1 Decisões Tomadas

5.2 Decisões Alternativas e Justificativas

6. Conclusão