

PROJETO Smart Cities

Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas

INTEGRANTES DO PROJETO e RA'S

Dandara Monike Santos Oliveira - 25027244

João Victor Verissimo Alves - 25027337

Jonathan Paiva Araujo - 25027171

Lucas Soares Corsino – 25027113

São Paulo

2025

1. Introdução

A tecnologia está cada vez mais presente no nosso dia a dia, com isso surgem novas formas de tornar cidades e casas mais inteligentes. Este projeto tem como objetivo desenvolver um dashboard simples e funcional para monitorar sensores e controlar atuadores em tempo real.

A ideia é criar uma solução prática, acessível e intuitiva, que ajude o usuário a entender e interagir melhor com seus espaços por meio da tecnologia. Ao longo deste documento, serão apresentados os principais requisitos, funcionalidades e características do sistema proposto.

2. DOCUMENTO DE ABERTURA DO PROJETO

Prefácio

Este documento destina-se a todos os membros da equipe do projeto integrador, professores orientadores e stakeholders interessados na construção de um sistema voltado para cidades inteligentes. A versão atual representa a primeira entrega formal da equipe, com o intuito de documentar o escopo, requisitos iniciais e estrutura geral do projeto. Caso haja versões futuras, elas terão como objetivo refletir melhorias e ajustes identificados durante o andamento do projeto.

Introdução

A proposta deste projeto nasce da necessidade de transformar cidades e casas comuns em ambientes inteligentes, mais sustentáveis e eficientes. A aplicação busca criar um dashboard funcional que permita o controle de sensores e atuadores em tempo real, contribuindo para o gerenciamento eficaz dos recursos urbanos e residenciais.

Com foco em facilitar a interação entre usuários e a infraestrutura inteligente, o sistema será conectado a um simulador que representará uma cidade ou casa inteligente. Através dessa integração, será possível testar, analisar e demonstrar o funcionamento de soluções voltadas para a sustentabilidade e bem-estar da população.

Essa proposta se alinha aos objetivos da Flex Automation, empresa referência no setor, que

visa desenvolver tecnologias com baixo impacto ambiental, otimizando o uso dos recursos naturais e promovendo qualidade de vida.

Glossário

- Dashboard: Interface gráfica que centraliza informações e controles do sistema em tempo real.
- Sensor: Dispositivo que capta informações do ambiente (ex: temperatura, umidade, presença).
- Atuador: Dispositivo que realiza ações no ambiente, como ligar/desligar luzes ou ajustar climatização.
- Simulador: Ferramenta que reproduz o comportamento de uma cidade/casa inteligente para fins de teste e validação.
- Smart City: Cidade que utiliza tecnologia para melhorar a gestão de recursos e serviços urbanos.
- Flex Automation: Empresa que atua no desenvolvimento de soluções inteligentes para automação residencial e urbana.

Definição de requisitos de usuário

O sistema atenderá o usuário final (residente): Pessoa que deseja monitorar e controlar sua residência, otimizando o consumo de energia, água e demais recursos. Terá acesso a uma interface amigável via dispositivos móveis.

Requisitos não funcionais também são considerados nesta seção, como a necessidade de que o sistema seja seguro, responsivo, acessível e de fácil manutenção. As boas práticas de usabilidade e acessibilidade também serão observadas durante o desenvolvimento.

Arquitetura do sistema

O sistema será baseado em uma arquitetura modular em camadas, composta por:

- Interface de usuário (frontend): acessível por navegador ou aplicativo móvel.
- Camada de serviço (backend): responsável pelo processamento dos dados e comunicação com o simulador.
- Camada de dados: banco de dados que armazenará logs, configurações e estado dos sensores/atuadores.

- Integração com o simulador: via API, o sistema enviará comandos e receberá feedback em tempo real.

Componentes reutilizáveis, como bibliotecas de gráficos e painéis interativos, serão integrados para acelerar o desenvolvimento e garantir confiabilidade.

Especificação de requisitos do sistema

O sistema incluirá tanto requisitos funcionais (como o controle de dispositivos, envio de alertas e visualização de dados) quanto não funcionais (como tempo de resposta, segurança e portabilidade). A próxima seção detalhará essas especificações com maior profundidade.

Modelos do sistema

Serão desenvolvidos modelos gráficos, como:

- Diagramas de caso de uso para ilustrar interações entre os usuários e o sistema;
- Diagramas de classes para representar a estrutura do software;
- Diagrama de arquitetura para demonstrar a organização das camadas e suas interações.

Evolução do sistema

Espera-se que o sistema evolua de acordo com a ampliação das funcionalidades do simulador e as necessidades dos usuários. Mudanças no hardware ou no ambiente digital (como novas tecnologias de IoT) poderão demandar ajustes futuros, e a arquitetura foi pensada para ser escalável e flexível a essas mudanças.

Apêndices

O projeto prevê o uso de um banco de dados relacional leve (ex: SQLite ou MySQL) e hospedagem local durante a fase de testes. O hardware mínimo necessário inclui um servidor de testes e dispositivos simulados conectados à internet. As informações específicas sobre as configurações técnicas serão detalhadas nos apêndices técnicos e anexos posteriores.

3. REQUISITOS DE SISTEMA

3.1 REQUISITOS FUNCIONAIS DE SOFTWARE

RFS01

Função: Monitoramento de Sensores

Descrição: Captura dados de sensores como temperatura, umidade, presença, etc.

Entradas: Leituras dos sensores conectados

Fonte: Sensores simulados

Saídas: Dados de ambiente atualizados no dashboard

Ação: Atualizar painel em tempo real

RFS02

Função: Controle de Atuadores

Descrição: Permite ligar/desligar dispositivos como luzes, climatização, entre outros.

Entradas: Comando do usuário via interface

Fonte: Interface do usuário

Saídas: Mudança de estado do dispositivo

Ação: Enviar comando ao atuador

RFS03

Função: Exibição de Painel de Controle

Descrição: Apresenta informações de sensores e estado de atuadores em uma interface visual

Entradas: Dados processados pelo backend

Fonte: Banco de dados e backend

Saídas: Painel gráfico atualizado

Ação: Renderizar dados em tempo real

RFS04

Função: Geração de Relatórios

Descrição: Gera relatórios sobre consumo e funcionamento do sistema

Entradas: Histórico de dados

Fonte: Banco de dados

Saídas: PDF ou visualização de relatório

Ação: Criar e exportar documento

RFS05

Função: Notificações e Alertas

Descrição: Dispara alertas em caso de situações críticas (ex: alta temperatura)

Entradas: Leituras fora do padrão

Fonte: Sensores

Saídas: Alerta visual/sonoro

Ação: Emitir aviso ao usuário

RFS06

Função: Autenticação de Usuário

Descrição: Garante acesso seguro à plataforma

Entradas: Credenciais do usuário

Fonte: Formulário de login

Saídas: Sessão de usuário ativa

Ação: Validar e liberar acesso

3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

RNFS01

Função: Segurança

Descrição: O sistema deve garantir acesso apenas a usuários autenticados e criptografar dados sensíveis.

Entradas: Credenciais do usuário, comandos e dados transmitidos

Fonte: Interface de login, APIs e banco de dados

Saídas: Acesso autorizado/negado, comandos seguros

Ação: Validar usuários, criptografar dados e registrar logs de acesso

RNFS02

Função: Performance

Descrição: O dashboard deve atualizar os dados dos sensores em tempo real, com latência inferior a 2 segundos.

Entradas: Dados dos sensores, comandos de controle

Fonte: Dispositivos IoT e interface do usuário

Saídas: Atualizações visuais e respostas do sistema

Ação: Processar os dados rapidamente e atualizar a interface em até 2 segundos

RNFS03

Função: Usabilidade

Descrição: A interface deve ser amigável, intuitiva e acessível via desktop e dispositivos móveis.

Entradas: Interações do usuário (cliques, seleções, comandos)

Fonte: Interface gráfica do dashboard

Saídas: Feedback visual claro e acessível

Ação: Exibir informações e ações com clareza e boa organização

RNFS04

Função: Disponibilidade

Descrição: O sistema deve estar disponível 99% do tempo durante os testes simulados.

Entradas: Requisições de dados e comandos

Fonte: Usuários e dispositivos conectados

Saídas: Dados atualizados, comandos executados

Ação: Garantir operação contínua com monitoramento de falhas e redundância

RNFS05

Função: Portabilidade

Descrição: O sistema deve ser capaz de rodar em diferentes plataformas (Windows, Linux, Android).

Entradas: Acesso via navegador ou aplicativo

Fonte: Diversos sistemas operacionais e dispositivos

Saídas: Interface responsiva e funcional

Ação: Adaptar a interface automaticamente ao dispositivo utilizado

RNFS06

Função: Manutenibilidade

Descrição: O código deve seguir boas práticas de organização e permitir futuras manutenções.

Entradas: Novos requisitos ou correções futuras

Fonte: Equipe de desenvolvimento

Saídas: Versões atualizadas e com baixo impacto em funcionalidades existentes

Ação: Aplicar modificações no código com base em boas práticas e versionamento

4. CASOS DE USO

CU01 - Monitorar Sensores Ambientais

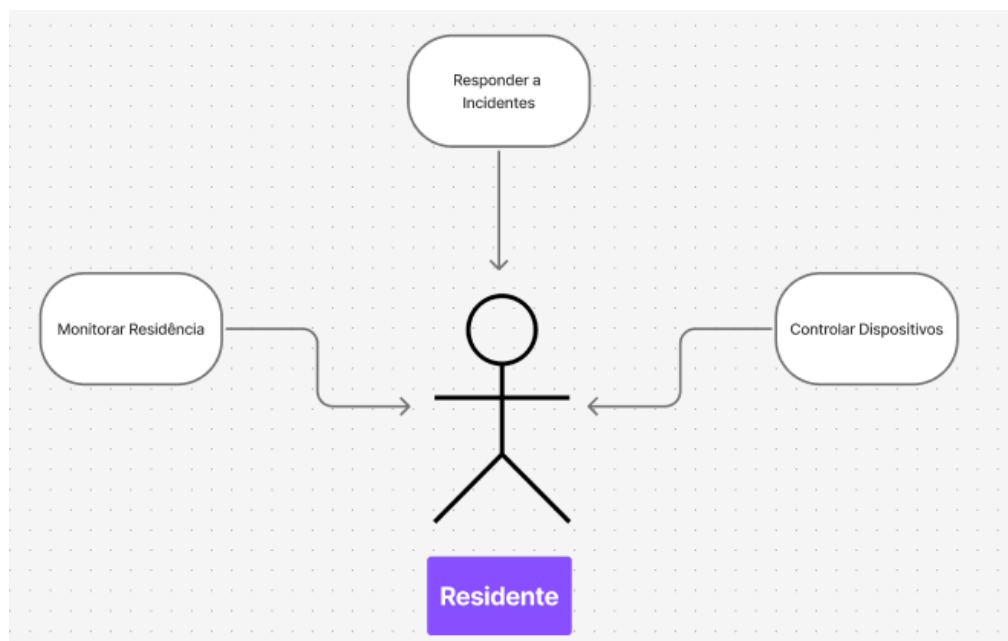
Descrição: Este caso de uso descreve a ação do usuário em visualizar os dados em tempo real coletados pelos sensores instalados em uma casa/cidade inteligente.

CU02 - Controlar Atuadores

Descrição: Este caso de uso descreve a ação do usuário de ativar ou desativar dispositivos como luzes, climatização e irrigação por meio do painel de controle.

CU03 - Gerar Relatórios de Consumo

Descrição: Este caso de uso descreve a ação do controlador da cidade ao solicitar um relatório de histórico de consumo e operação dos dispositivos conectados.

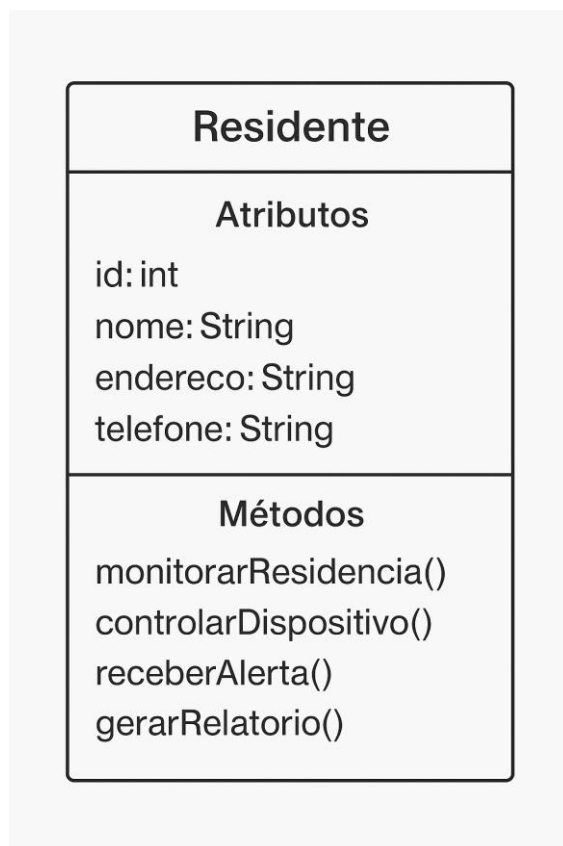


5. DIAGRAMA DE CLASSE

O sistema será estruturado com as seguintes classes principais:

- Classe Sensor: representa os sensores físicos ou simulados, contendo atributos como tipo, valor atual e status.
- Classe Atuador: representa dispositivos que podem ser acionados, contendo atributos como tipo, estado e comandos.
- Classe Usuario: armazena dados dos usuários e suas permissões.
- Classe Dashboard: coordena a interface entre sensores, atuadores e usuários.
- Classe Relatorio: gera e exporta os dados para visualização ou análise.

Essas classes se relacionam entre si por meio de métodos de controle, autenticação e visualização de dados.



6. ARQUITETURA DO SISTEMA

A arquitetura do sistema será baseada em três camadas principais:

1. Camada de Apresentação (Frontend):

- Responsável pela interface gráfica com o usuário.
- Desenvolvida para acesso via navegador e dispositivos móveis.

2. Camada de Lógica (Backend):

- Responsável pelo processamento dos dados e regras de negócio.
- Recebe informações dos sensores e envia comandos aos atuadores.

3. Camada de Dados:

- Armazena todas as informações históricas, configurações de sensores e usuários.
- Utiliza banco de dados relacional leve para testes locais.

O sistema se comunica com um simulador via API para testes de integração e visualização em tempo real.

7. REFERÊNCIAS BIBLIOGRÁFICAS

SOMMERVILLE, I. Engenharia de Software. 11ª Edição. São Paulo: Pearson Addison-Wesley, 2017.

FLEX AUTOMATION. Projeto Smart Cities. Disponível em: <https://flexautomation.com.br/>

Planet Smart City. Cidades Inteligentes. Disponível em: <https://planetsmartcity.com/>