

PROJETO.....

Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas

INTEGRANTES DO PROJETO e RA'S

ENZO NUNES ZAMBON MINARDI DY AZEVEDO – 25027841

FABRICIO MARQUES DE SOUZA – 25027377

GUILHERME HARADA MORATO – 25027845

MURILLO LUEDERS AZEREDO DINIZ COSTA – 22010871

São Paulo

2025

Sumário

1 INTRODUÇÃO	4
2. DOCUMENTO DE ABERTURA DO PROJETOS.....	6
2.1 – Project Charter.....	6
2.2 – Histórias do Usuário	8
3. DESIGN SPRINT – Ideação e prototipação do desafio.....	9
3.1 Desafio	9
3.2 Entender Mapear	9
3.3 Ideação – desenho da solução (trilha do usuário)	9
3.4 Prototipagem.....	9
4.REQUISITOS DE SISTEMA	9
4.1 REQUISITOS FUNCIONAIS DE SOFTWARE	9
4.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE	12
5. CASOS DE USO	14
6. DIAGRAMA DE CLASSE.....	17
7. ARQUITETURA DO SISTEMA	17
8. REFERÊNCIAS BIBLIOGRÁFICAS	18

1 INTRODUÇÃO

A crescente urbanização e o aumento populacional têm exercido pressão significativa sobre os recursos naturais e as infraestruturas urbanas. Problemas como o consumo excessivo de energia, desperdício de água, poluição do ar e mobilidade urbana se tornaram desafios críticos para os gestores públicos e para a população.

Nesse cenário, as cidades inteligentes surgem como resposta tecnológica e sustentável para lidar com tais problemas. Essas cidades são projetadas para integrar tecnologias de informação e comunicação (TIC) ao ambiente urbano, criando um ecossistema onde sensores, dispositivos e cidadãos interagem de forma eficiente e orientada por dados.

A Flex Automation propõe o desenvolvimento de um sistema digital capaz de monitorar, controlar e otimizar recursos em uma cidade inteligente, com um foco especial também nas residências inteligentes, permitindo que cada cidadão colabore diretamente para a eficiência geral do sistema urbano.

SmartCities/SmartHouse

Nome da Instituição: Flex Automation

Objetivo da Aplicação:

Desenvolver um dashboard modular e interativo capaz de se integrar a um simulador de cidade inteligente (smart city) e residência automatizada (smart home), proporcionando controle e monitoramento em tempo real de sensores e atuadores. O sistema visa promover a sustentabilidade, a automação eficiente e a gestão inteligente de recursos urbanos e domésticos, contribuindo para a otimização energética, a redução de desperdícios e a melhoria da qualidade de vida.

Objetivos Específicos:

Visualização de dados em tempo real: Desenvolver uma interface intuitiva e dinâmica que exiba informações provenientes de sensores ambientais, de consumo energético e de segurança, permitindo ao usuário acompanhar o status de diferentes dispositivos de forma clara e acessível.

Controle remoto de atuadores: Implementar funcionalidades que simulem a interação com dispositivos físicos, como iluminação, climatização (ar-condicionado, aquecedores), sistemas de segurança (trancas, câmeras) e eletrodomésticos, possibilitando testes de automação em um ambiente simulado antes da aplicação em cenários reais.

Adaptação a múltiplos perfis de usuários: Oferecer níveis diferenciados de acesso e personalização, atendendo tanto ao usuário doméstico (que busca facilidade e praticidade) quanto ao gestor

público ou administrador (que necessita de análises detalhadas e controle centralizado).
Conscientização sobre sustentabilidade: Integrar métricas de eficiência energética e consumo sustentável, fornecendo relatórios e sugestões automatizadas para reduzir desperdícios, incentivando boas práticas ambientais por meio da tecnologia.

Modularidade e escalabilidade: Garantir que a arquitetura do sistema seja flexível e expansível, permitindo a inclusão de novos sensores, atuadores e funcionalidades sem comprometer a segurança ou a estabilidade da plataforma, preparando-a para futuras integrações em ambientes reais.

]Tomada de decisão baseada em dados: Disponibilizar ferramentas de análise preditiva e histórica, como gráficos de tendência e alertas personalizados, para identificar padrões de uso, antever falhas e agir de forma preventiva, aumentando a eficácia na gestão de recursos.

Desafio:

O projeto da Flex Automation, assim como outras iniciativas, trabalha para poder criar cidades inteligentes buscando a sustentabilidade, o melhor uso dos recursos planetários e o menor impacto na natureza. Para que isso ocorra é necessário ter uma alta capacidade de mensuração e controle para a otimização da vida na cidade, desde recursos até o tráfego de pedestres. Também, a conscientização da população de como uma cidade inteligente funciona e/ou é controlada, de forma a instruir sobre as melhores maneiras para a cidade a ser sustentável.

Personas a Serem Atendidas:

-**Usuário final** do sistema, que deseja controlar sua casa de forma a gastar menos e otimizar os recursos da cidade. Considere que o usuário possui conhecimento básico para utilizar dispositivos mobile.

-**Controlador da cidade**, um funcionário da cidade que deve acompanhar um dashboard/mapa/painel informativo da cidade, tratando situações inesperadas, acompanhando os dados dos sensores e acionando os programas da cidade. Considere que ele tem um conhecimento médio para avançado de tecnologia.

Recursos:

https://store.steampowered.com/app/949230/Cities_Skylines_II/

https://store.steampowered.com/app/2741560/SimCity_3000_Unlimited/

<https://planetsmartcity.com/> <https://flexautomation.com.br>

2. DOCUMENTO DE ABERTURA DO PROJETOS

2.1 – Project Charter

Prefácio

Público-alvo:

- Este documento destina-se a:
- Equipe de desenvolvimento
- Stakeholders do projeto
- Gestores de TI e inovação
- Usuários finais (moradores e gestores públicos)

Introdução

Necessidade do Sistema:

Com o crescimento das cidades inteligentes e da automação residencial, surge a demanda por uma ferramenta que permita monitorar e controlar dispositivos IoT de forma integrada, promovendo eficiência energética e sustentabilidade.

Funcionalidades Principais:

Visualização em tempo real de dados de sensores Controle remoto de atuadores (luzes, climatização, segurança) Geração de relatórios de consumo e eficiência

Integração com Outros Sistemas:

O sistema se comunicará com:

Plataformas de simulação de cidades inteligentes Bancos de dados para armazenamento de métricas APIs de dispositivos IoT (ex.: iluminação pública, sensores ambientais) Alinhamento com Objetivos Estratégicos: Redução de custos operacionais Melhoria na gestão de recursos urbanos e residenciais Promoção de tecnologias sustentáveis

Glossário

Termo Definição

Dashboard Interface gráfica que exibe dados e controles em tempo real

IoT (Internet of Things) Dispositivos conectados à internet que coletam e trocam dados

Smart City	Cidade que usa tecnologia para melhorar infraestrutura e serviços
Atuador	Dispositivo que executa ações físicas (ex.: ligar luzes, abrir portas)

Definição de requisitos de usuário

Requisitos Funcionais:

RF01: O sistema deve exibir dados de sensores em gráficos e tabelas.

RF02: O usuário deve poder acionar atuadores remotamente.

RF03: O sistema deve gerar relatórios de consumo energético.

Requisitos Não Funcionais:

RNF01: Tempo de resposta inferior a 2 segundos.

RNF02: Suporte a 100+ usuários simultâneos.

RNF03: Segurança de dados (criptografia e autenticação). Normas Aplicáveis: Padrão MQTT para comunicação com dispositivos IoT. Conformidade com LGPD para tratamento de dados..

Arquitetura do sistema

Visão Geral:

Diagrama de Arquitetura Simplificado

Front-end: Dashboard web responsivo

Back-end: API REST em Node.js

Banco de Dados: Mysql para armazenamento histórico

Integração IoT: Protocolo MQTT para sensores/atuadores

Componentes Reutilizados:

Bibliotecas de gráficos (ex.: Chart.js) Frameworks de autenticação (ex.: OAuth 2.0)

Especificação de requisitos do sistema

Requisitos Detalhados:

RF04: O sistema deve permitir agendamento de ações (ex.: desligar luzes às 22h).

RNF04: Disponibilidade de 99,9% (SLA).

Interfaces Externas: API para integração com sistemas de smart city existentes.

Modelos do sistema

Fluxo de Dados:

Sensores → Broker MQTT → Back-end → Dashboard

Evolução do sistema

Premissas:

Hardware compatível com IoT.

Crescimento escalável para mais dispositivos.

Futuras Melhorias:

Adição de machine learning para previsão de consumo.

Suporte a assistentes de voz (Alexa, Google Assistant).

Apêndices

Requisitos de Hardware:

Servidor: 4GB RAM, 2vCPUs (mínimo)

Dispositivos IoT: Suporte a Wi-Fi/Bluetooth

2.2 –Histórias do Usuário

Alguns detalhes sobre a casa inteligente que cujos dados estão no arquivo anexo:

-2 Pessoas vivem nesta casa

-A casa possui 2 quartos, 1 sala, 1 cozinha e 1 piscina e são identificados respectivamente pelos sensores de ID: 1, 2, 3, 4, 5.

-O gasto energético médio para deixar cada local ligado é:

Quartos (ID 1 e 2) – 1,5KWatts/Hora (Considerando 1 TV,1 lâmpada e um ar-condicionado)

Sala (ID 3) – 50Watts/Hora (Considerando 1 TV e 5 lâmpadas)

Cozinha (ID 4) – 3KWatts/Hora (Considerando 1 Micro-ondas, 1 máquina de lavar louça e 3 lâmpadas)

Piscina (ID 5) – 7KWatts/Hora (Bomba + Aquecedor)

Você tem a possibilidade de adicionar comandos separados para controlar cada um dos elementos descritos acima.

EXEMPLO DA BASE DOS SENSORES

TimeStamp	ID_Sensor	Temperatura	Umidade	Movimento
28/4/25 0:18	3	39	71	0
22/5/25 4:43	4	19	82	0
20/4/25 20:38	3	24	71	0
12/2/25 0:03	1	22	22	0
14/4/25 1:33	2	19	46	1
27/1/25 14:21	2	37	27	0
30/5/25 7:19	1	10	87	0
21/7/25 6:17	1	34	88	0
21/1/25 9:20	3	39	28	0
2/2/25 23:55	4	28	33	0
22/6/25 14:15	3	17	32	0
24/6/25 15:22	2	38	29	0
30/4/25 0:32	2	18	88	1
26/6/25 2:00	2	26	63	0
26/6/25 10:09	2	21	50	0
1/3/25 7:15	5	40	30	1
27/6/25 7:02	3	15	28	1

3.DESIGN SPRINT – Ideação e prototipação do desafio

3.1 Desafio

3.2 Entender Mapear

3.3 Ideação – desenho da solução (trilha do usuário)

3.4 Prototipagem

4.REQUISITOS DE SISTEMA

4.1 REQUISITOS FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

RFS01	
Função	Exibição de dados em tempo real
Descrição	O sistema deve apresentar os dados coletados pelos sensores de forma gráfica e tabular
Entradas	Dados brutos dos sensores (luminosidade, temperatura, consumo energético)
Fonte	Dispositivos IoT conectados via protocolo MQTT
Saídas	Gráficos atualizados em tempo real e tabelas com valores numéricos
Ação	Coletar, processar e exibir dados a cada 5 segundos

RFS02	
Função	Acionamento de dispositivos
Descrição	Permitir que usuários ativem/desativem atuadores remotamente
Entradas	Comandos do usuário (ligar/desligar, ajustar parâmetros)
Fonte	Interface web do dashboard
Saídas	Sinais de controle para os dispositivos físicos
Ação	Enviar comandos via MQTT para os atuadores correspondentes

RFS03	
Função	Criação de documentos analíticos
Descrição	Produzir relatórios periódicos de consumo energético
Entradas	Dados históricos armazenados no banco de dados
Fonte	Banco de dados Mysql
Saídas	Arquivos PDF/CSV com análises comparativas
Ação	Processar dados e gerar relatórios diários/semanais/mensais

--	--

RFS04	
Função	Programação temporal de operações
Descrição	Permitir agendamento de comandos pré-definidos
Entradas	Configurações de tempo e ação (ex.: desligar luzes às 22h)
Fonte	Interface de agendamento do dashboard
Saídas	Comandos automáticos no horário programado
Ação	Armazenar programações e executar no momento especificado

RFS05	
Função	Tolerância a falhas de conexão
Descrição	Manter operação local por 24h em caso de perda de internet
Entradas	Status da conexão de rede
Fonte	Monitoramento de rede
Saídas	Operação em modo offline
Ação	Armazenamento local temporário e sincronização quando reconectar

RFS06	
Função	Acessibilidade
Descrição	Interface deve atender a WCAG 2.1 AA
Entradas	Diretrizes de acessibilidade
Fonte	Normativas internacionais
Saídas	Interface adaptável a diferentes necessidades
Ação	Implementar contraste, leitor de tela e navegação por teclado

4.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

RFS01	
Função	Tempo de resposta
Descrição	O sistema deve responder em menos de 2 segundos
Entradas	Requisições do usuário
Fonte	Interações na interface web
Saídas	Feedback visual imediato
Ação	Otimização de queries e balanceamento de carga

RFS02	
Função	Capacidade de usuários simultâneos
Descrição	Capacidade de usuários simultâneos
Entradas	Acessos concorrentes ao sistema
Fonte	Requisitos de arquitetura
Saídas	Operação estável sob carga
Ação	Implementação de cache e arquitetura escalável

RFS03	
Função	Proteção de dados
Descrição	Implementar autenticação e criptografia
Entradas	Credenciais de acesso
Fonte	Políticas de segurança da organização
Saídas	Dados protegidos e acessos controlados
Ação	Uso de HTTPS, OAuth 2.0 e criptografia AES-256

--	--

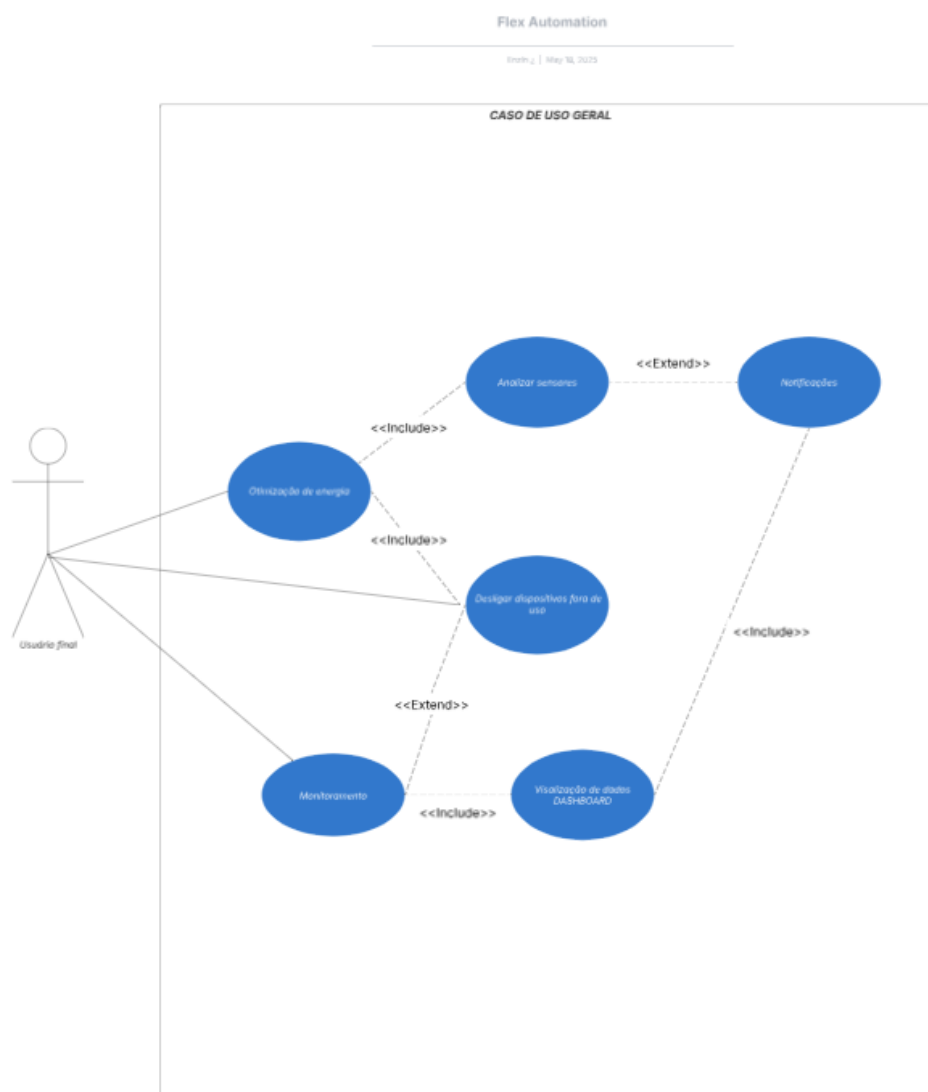
RFS04	
Função	Alertas personalizados
Descrição	Sistema deve enviar notificações quando ocorrerem anomalias ou valores fora do padrão
Entradas	Limites configurados pelo usuário e dados dos sensores
Fonte	Regras de negócio e fluxo de dados em tempo real
Saídas	Notificações visuais (dashboard) e por e-mail
Ação	Comparar valores dos sensores com thresholds e disparar alertas

RFS05	
Função	Controle por voz Descrição
Descrição	Permitir interação com o sistema através de comandos de voz via assistentes virtuais
Entradas	Comandos de voz do usuário
Fonte	Plataformas de assistentes virtuais (Google Assistant, Alexa)
Saídas	Execução dos comandos correspondentes nos dispositivos IoT
Ação	Captura e processamento dos comandos de voz Conversão para texto e interpretação da intenção Validação da autenticação por voz Tradução para comandos do sistema Execução da ação correspondente Confirmação auditiva da operação

RFS06	
Função	Gerenciamento de acessos
Descrição	Definir permissões específicas por funcionalidade para cada perfil de usuário
Entradas	Tipo de perfil (Admi,morador) Lista de funcionalidades do sistema
Fonte	Configuração pelo administrador Políticas de segurança da organização
Saídas	Mapa de permissões por perfil Bloqueio/seleção de funcionalidades na interface

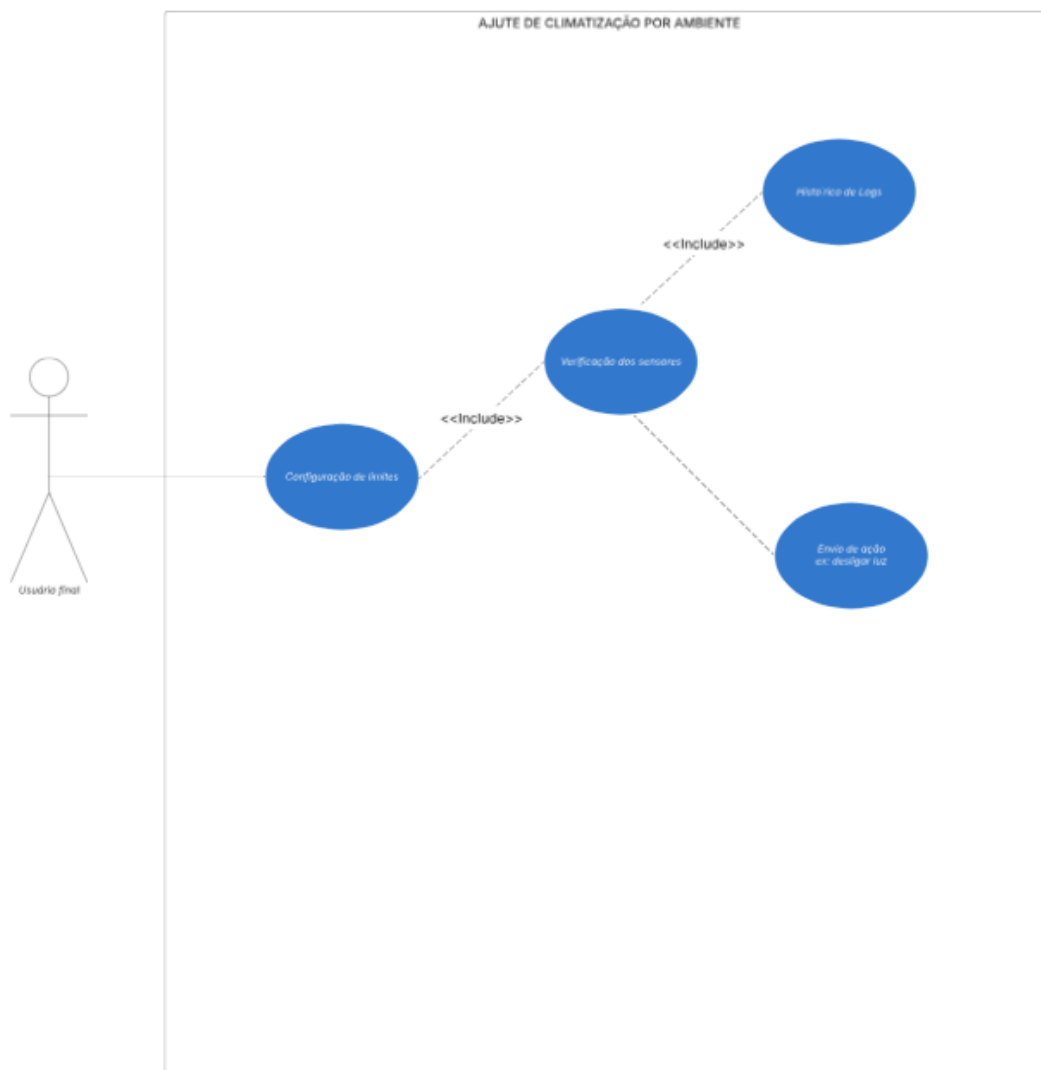
Ação	<p>Vincular ações permitidas a cada perfil</p> <p>Aplicar restrições em tempo real</p> <p>Registrar alterações em log de auditoria</p>
------	--

5. CASOS DE USO



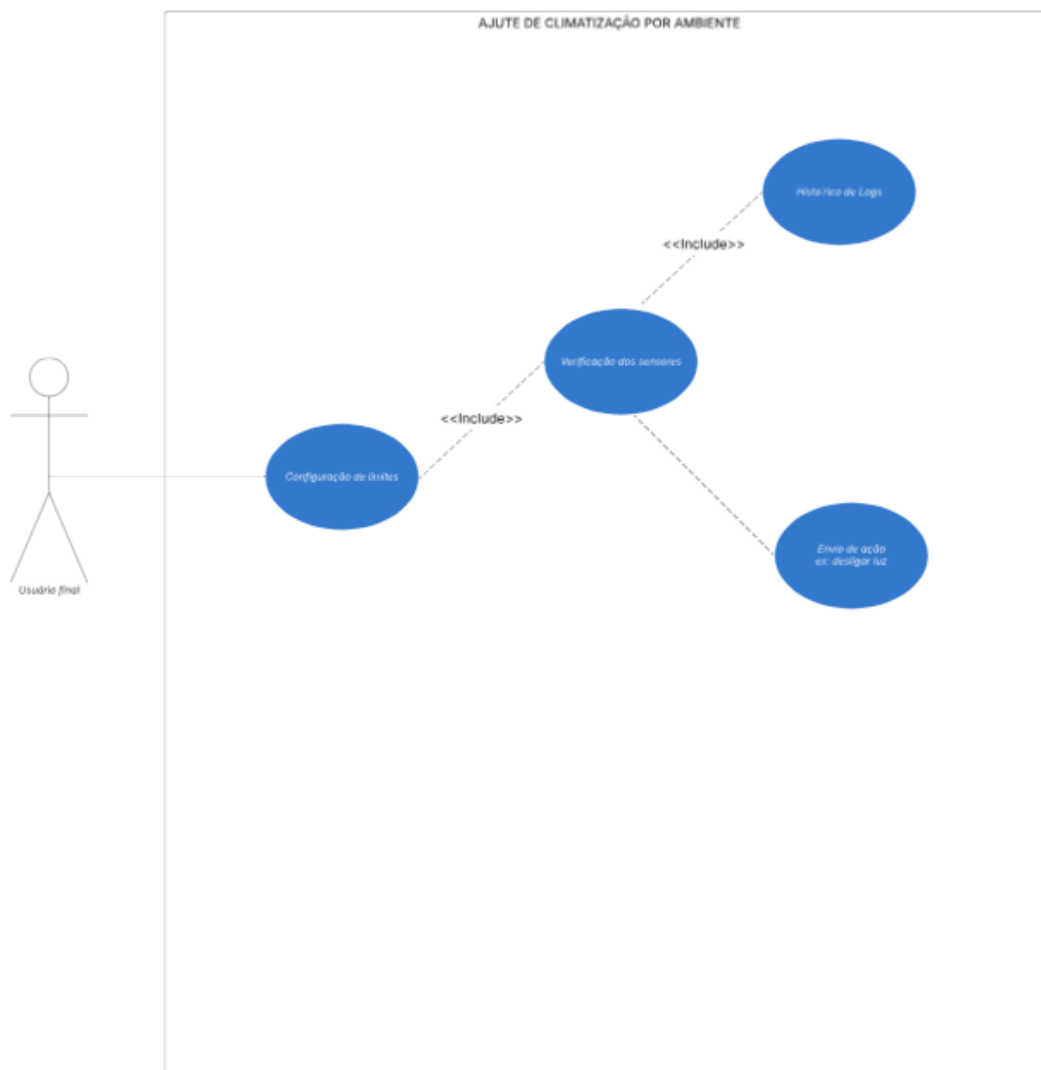
Flex Automation

Diagram 2 | May 16, 2023

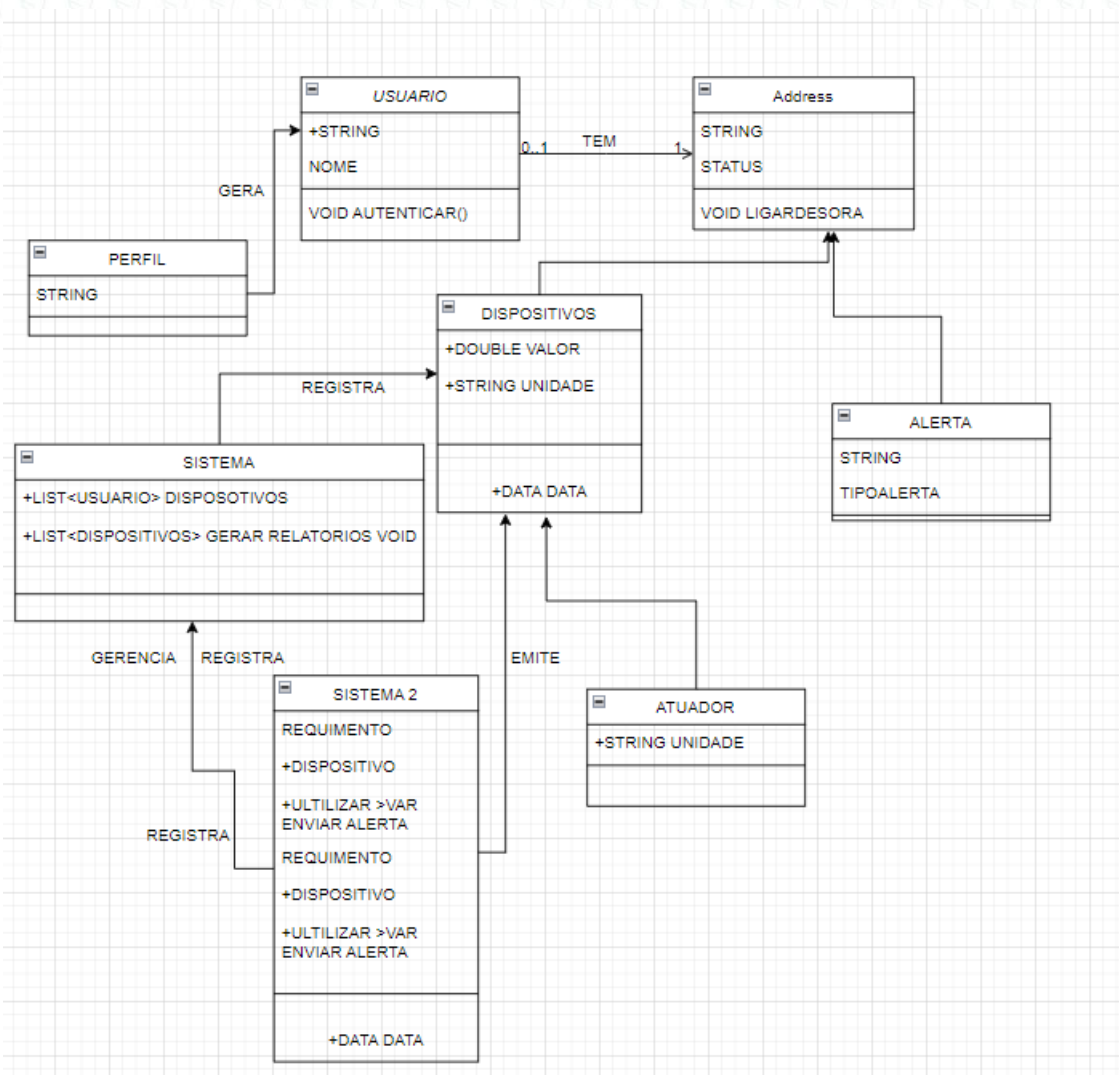


Flex Automation

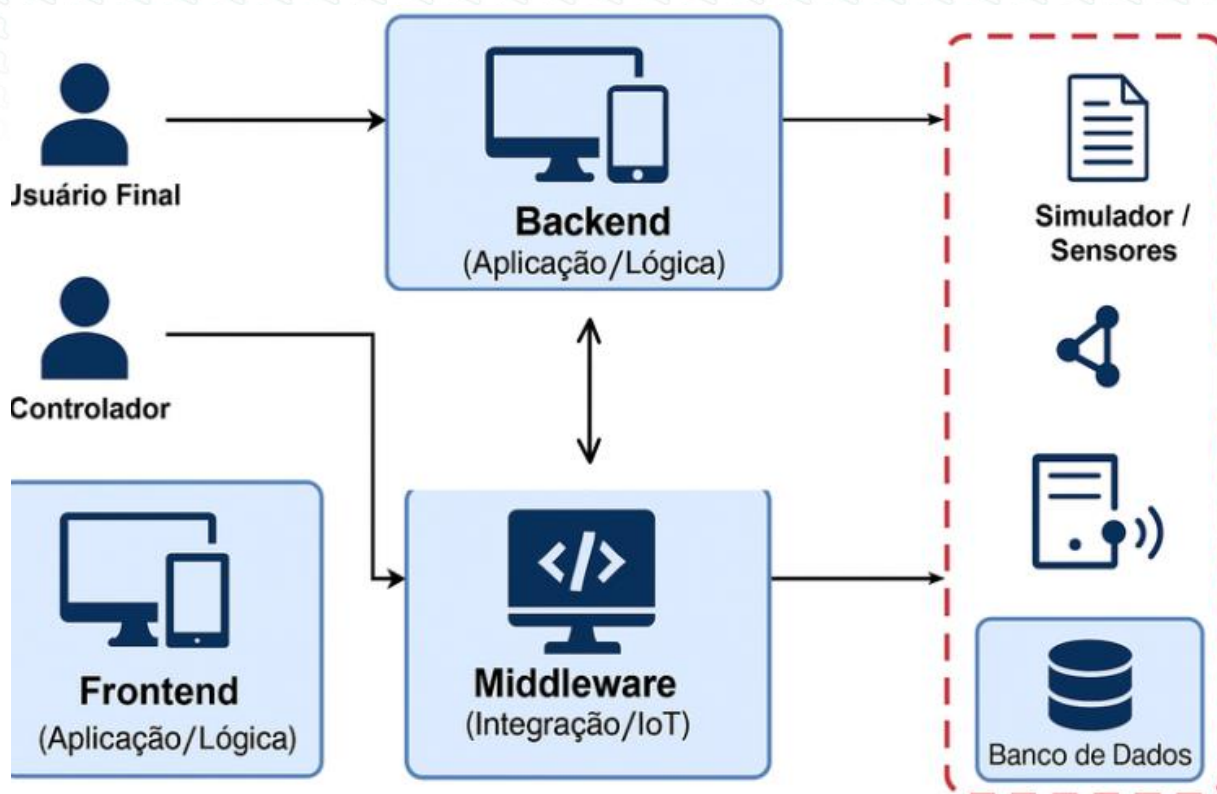
Diagram 2 | May 16, 2023



6. DIAGRAMA DE CLASSE



7. ARQUITETURA DO SISTEMA



8.REFERÊNCIAS BIBLIOGRÁFICAS

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