

## **PROJETO**

**Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas**

### **INTEGRANTES DO PROJETO e RA'S**

**André de Sousa Pereira: 25027905**

**Felipe Nunes: 252027941**

**Gregory: 25027942**

**Icaro Souza: 25027842**

São Paulo

2025

## Sumário

### 1 INTRODUÇÃO

### 2. DOCUMENTO DE ABERTURA DO PROJETOS..

#### 2.1 – Project Charter

#### 2.2 – Histórias do Usuário.

### 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..

#### 4.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE..

### 5. CASOS DE USO..

### 6. DIAGRAMA DE CLASSE..

### 7. ARQUITETURA DO SISTEMA..

### 8. REFERÊNCIAS BIBLIOGRÁFICAS..

# 1 INTRODUÇÃO

Problemas de Smart Cities

## Smart Cities/Smart House

**Nome da Instituição:** Flex Automation

**Objetivo da Aplicação:**

ADS1 O objetivo do desafio é gerar um dashboard de uma cidade/casa inteligente que permita o controle de sensores e atuadores.

Este desafio busca, de forma modular, introduzir como uma cidade/casa inteligente pode ser controlada, tratando seus dados de forma a aprimorar o sistema e otimizando a sustentabilidade.

Seu dashboard deverá receber e enviar sinais de/para um simulador de casa/cidade inteligente, provenientes da rede/internet. O servidor será fornecido pelos professores.

### 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 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/949230/Cities_Skylines_II/)

[https://store.steampowered.com/app/2741560/SimCity\\_3000\\_Unlimited/](https://store.steampowered.com/app/2741560/SimCity_3000_Unlimited/)

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

## **2. DOCUMENTO DE ABERTURA DO PROJETO**

### **2.1 – Project Charter**

#### **Prefácio**

Este projeto foi desenvolvido para moradores de casas inteligentes que desejam ter mais controle e consciência sobre seus gastos do dia a dia. A proposta é criar um dashboard simples, intuitivo e visualmente agradável, que centralize as informações de consumo da casa, como energia, água e gás.

#### **Introdução**

Este trabalho tem como objetivo o desenvolvimento de um dashboard para controle de uma casa inteligente, reunindo funções como controle de iluminação, climatização, segurança, consumo de energia e automações programadas. A interface será projetada para ser intuitiva, permitindo ações como ligar/desligar dispositivos, monitorar temperatura e segurança, além de exibir dados em tempo real.

#### **Glossário**

Casa inteligente, Dashboard, Consumo energético, Gasto hídrico, Automação residencial, IoT (Internet das Coisas), Eficiência energética, Interface amigável

#### **Definição de requisitos de usuário**

#### **Arquitetura do sistema**

- **Módulo de Sensoriamento (Coleta de Dados)**  
Responsável por captar os dados dos dispositivos físicos instalados na residência:
- Sensores de consumo de energia elétrica (medidores inteligentes).
- Sensores de presença ou luminosidade para controle automatizado da luz.
- Relógio em tempo real (RTC) e Temporizadores para registrar padrões de consumo por hora/dia/semana.
- Controladores embarcados fazem a leitura dos sensores e enviam os dados para o sistema.

#### **Módulo de Comunicação**

Realiza a transmissão dos dados dos sensores para o servidor:

- Protocolos como MQTT ou HTTP são utilizados para comunicação leve e eficiente.
- Pode usar Wi-Fi ou Zigbee, dependendo do ambiente da casa.

#### **Módulo de Processamento e Armazenamento**

- Os dados recebidos são armazenados em banco de dados (SQL ou NoSQL).
- Um serviço de backend realiza o processamento dos dados, gera relatórios, identifica padrões e envia comandos automatizados (ex: desligar luzes após determinado horário).

#### **Módulo de Visualização (Dashboard)**

- Interface web responsiva acessível via navegador ou app móvel.
- Exibe gráficos de consumo, alertas de uso excessivo e permite o controle remoto das luzes.
- Mostra o status em tempo real de cada cômodo (luz acesa/apagada, consumo atual etc).
- sistema de missões diárias para o menor consumo
- sistema de pontuação

## **Módulo de Controle e Alarme**

- **Gera alertas visuais caso haja consumo anormal ou falha nos dispositivos.**
- **Integra com sistemas de notificação (push).**

## **Fonte de Energia**

- **Todos os módulos físicos são alimentados energia solar**

## **Especificação de requisitos do sistema**

**Desempenho e Tempo de Resposta:** O sistema deve apresentar alto desempenho, garantindo que as operações executadas sejam rápidas, fluidas e sem travamentos perceptíveis ao usuário. O tempo de resposta para ações críticas, como a atualização de dados dos sensores, comandos do usuário e alterações na interface, deve ser minimizado.

**Exibir Consumo Energético por Cômodo:** A plataforma deve permitir ao usuário visualizar o consumo de energia elétrica detalhado por cômodo da residência. Essas informações devem ser apresentadas de forma clara, gráfica e intuitiva.

**Monitorar Dados dos Sensores em Tempo Real:** Os dados provenientes dos sensores (temperatura, umidade, presença, luminosidade, entre outros) devem ser monitorados e exibidos em tempo real na interface do sistema, permitindo ao usuário acompanhar o estado atual de cada ambiente da casa.

**Interface Responsiva e Compatibilidade Cross-Device:** A interface do sistema deve ser responsiva, adaptando-se automaticamente a diferentes tamanhos e resoluções de tela, garantindo uma boa experiência de uso tanto em desktops quanto em dispositivos móveis (smartphones, tablets).

## **Especificações de Hardware**

### **Requisitos Mínimos do Sistema**

Para garantir o funcionamento eficiente do Dashboard da SmartHouse, recomenda-se a seguinte configuração mínima de hardware:

- **Processador:** Intel Core i5 ou equivalente ARM (para sistemas embarcados)

- Memória RAM: 8 GB
- Armazenamento: SSD de 120 GB
- Placa de Rede: Suporte a Wi-Fi 802.11ac e Ethernet 10/100/1000
- Sistema Operacional: Ubuntu Server 20.04 LTS / Windows 10

Sistema de notificações: O sistema deve emitir alertas visuais quando o consumo de energia ou água ultrapassar uma média pré definida, deve notificar o usuário caso um sensor esteja desconectado ou apresente falha e deve alertar sobre consumo atípico com sugestões de economia através do “premios”

#### Reinicialização e Recuperação

O sistema deve retomar a operação automaticamente após queda de energia ou reinício certificando-se a restauração e a recuperação dos dados mais recentes disponíveis após falha ou reinício.

### **Modelos do sistema**

Este sistema utiliza modelos gráficos para representar visualmente os relacionamentos entre os componentes internos da dashboard de casa inteligente e a forma como ela interage com o ambiente físico da residência. Tais modelos auxiliam na compreensão da arquitetura, do fluxo de informações e das conexões entre sensores, interface de usuário e controle dos dispositivos.

### **Evolução do sistema**

A aplicação foi projetada com base em pressupostos fundamentais relacionados ao ambiente doméstico atual, composto por dois moradores e cinco ambientes monitorados por sensores inteligentes (quartos, sala, cozinha e piscina). O sistema assume uma infraestrutura de sensores compatível com envio contínuo de dados, bem como uma conexão estável à internet para comunicação entre os dispositivos físicos e a dashboard.

## 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**

RFS01	
<b>Função</b>	Monitorar dados dos sensores em tempo real
<b>Descrição</b>	O sistema deve capturar e exibir em tempo real as informações enviadas pelos sensores da casa (temperatura, umidade e movimento).
<b>Entradas</b>	Sensor
<b>Fonte</b>	Sensores instalados nos cômodos da casa inteligente
<b>Saídas</b>	Painel com dados atualizados por ambiente
<b>Ação</b>	Exibir os dados em cartões visuais na tela inicial e na aba de sensores, com atualização automática ou manual.

RFS02	
<b>Função</b>	Exibir consumo energético por cômodo
<b>Descrição</b>	O sistema deve calcular e apresentar o consumo energético estimado de cada ambiente com base nos sensores ativos e nos valores pré-definidos de consumo por local.
<b>Entradas</b>	Consumo energético
<b>Fonte</b>	Banco de dados interno + Sensores de movimento
<b>Saídas</b>	Gráficos de barra e linha

<b>Ação</b>	Apresentar visualmente o consumo atual e histórico (últimas 24h) por meio de gráficos interativos.
-------------	--

<b>RFS03</b>	
<b>Função</b>	Visualização de consumo energético por ambiente
<b>Descrição</b>	O sistema deve permitir que o usuário visualize o consumo energético individual de cada ambiente da casa (quartos, sala, cozinha, piscina), em tempo real ou por período selecionado.
<b>Entradas</b>	Intervalo de tempo selecionado pelo usuário
<b>Fonte</b>	Leituras dos sensores associadas a cada ambiente
<b>Saídas</b>	Gráfico de consumo energético específico por ambiente
<b>Ação</b>	Mostrar visualmente, através de gráficos, o consumo energético por local, permitindo comparações e análises detalhadas.

<b>RFS04</b>	
<b>Função</b>	Controle manual de dispositivos por ambiente
<b>Descrição</b>	O sistema deve permitir que o usuário ligue ou desligue dispositivos (TV, lâmpadas, ar-condicionado,

	bomba da piscina, etc.) de cada ambiente diretamente pela interface da dashboard.
<b>Entradas</b>	Botão ou switch de controle do dispositivo
<b>Fonte</b>	Comando de usuário via dashboard
<b>Saídas</b>	Estado atualizado do dispositivo (ligado/desligado)
<b>Ação</b>	Enviar comando ao sistema para alterar o estado do dispositivo selecionado, atualizando a visualização correspondente.

<b>RFS05</b>	
<b>Função</b>	Geração de relatórios históricos de consumo energético
<b>Descrição</b>	O sistema deve permitir que o usuário gere relatórios em PDF com o histórico de consumo energético geral e por ambiente, em períodos configuráveis.
<b>Entradas</b>	Data inicial e final selecionadas pelo usuário
<b>Fonte</b>	Dados armazenados no banco de dados
<b>Saídas</b>	Documento PDF com gráficos e dados consolidados
<b>Ação</b>	Coletar dados históricos e gerar relatório visual em PDF, com opção de download ou envio por e-mail.

<b>RFS06</b>	
<b>Função</b>	Deteção de presença e exibição em tempo real

<b>Descrição</b>	O sistema deve exibir em tempo real a presença de pessoas nos ambientes da casa com base nos dados do sensor de movimento.
<b>Entradas</b>	Leituras do sensor de movimento (0 ou 1)
<b>Fonte</b>	Sensores instalados em cada ambiente
<b>Saídas</b>	Indicador visual de presença (ícones ou cores nos ambientes)
<b>Ação</b>	Exibir visualmente a presença ou ausência de pessoas em cada ambiente, de forma intuitiva e em tempo real.

## 4.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

RFS01	
<b>Função</b>	Desempenho e tempo de resposta
<b>Descrição</b>	O sistema deve apresentar tempo de resposta inferior a 1 segundo para ações críticas, como a atualização dos dados de sensores e execução de comandos de controle.
<b>Entradas</b>	Comandos enviados via dashboard
<b>Fonte</b>	Solicitações dos usuários e atualizações dos sensores
<b>Saídas</b>	Dados exibidos na interface

<b>Ação</b>	Assegurar que as ações do usuário e os dados dos sensores sejam processados rapidamente, garantindo fluidez e experiência positiva.
-------------	---

RFS02	
<b>Função</b>	Interface responsiva e compatibilidade cross-device
<b>Descrição</b>	A aplicação deve ser responsiva e adaptável a diferentes dispositivos (computadores, tablets e smartphones), mantendo usabilidade e leitura clara em todas as resoluções.
<b>Entradas</b>	Comportamento de interação (toque, clique, arrastar)
<b>Fonte</b>	Dispositivo de acesso do usuário
<b>Saídas</b>	Interface redimensionada e reordenada automaticamente
<b>Ação</b>	Utilizar frameworks modernos de frontend (ex: Tailwind, Bootstrap ou CSS Grid) para garantir que todos os elementos da dashboard se adaptem corretamente a qualquer tela.

RFS03	
<b>Função</b>	Responsividade da Interface
<b>Descrição</b>	A dashboard deve adaptar-se automaticamente a diferentes tamanhos de tela (computadores, tablets e smartphones), garantindo boa usabilidade em qualquer dispositivo.
<b>Entradas</b>	Acesso do usuário por diferentes tipos de dispositivos
<b>Fonte</b>	Requisitos de design responsivo (CSS, frameworks como Tailwind, Bootstrap, etc.)

<b>Saídas</b>	Interface ajustada corretamente ao tamanho do dispositivo
<b>Ação</b>	Aplicar práticas de design responsivo para manter a legibilidade, navegação e usabilidade em qualquer tela.

<b>RFS04</b>	
<b>Função</b>	Tempo de Resposta em Tempo Real
<b>Descrição</b>	O sistema deve atualizar os dados em tempo real com um tempo de resposta inferior a 2 segundos após o recebimento das informações dos sensores.
<b>Fonte</b>	Sistema de sensores IoT da casa inteligente
<b>Saídas</b>	Dados atualizados na interface do usuário
<b>Ação</b>	Renderizar os dados recebidos quase instantaneamente para garantir precisão e confiabilidade na visualização.

<b>RFS05</b>	
<b>Função</b>	Escalabilidade do Sistema
<b>Descrição</b>	O sistema deve ser escalável, permitindo a inclusão de novos sensores, ambientes ou funcionalidades sem comprometer a performance ou exigir grandes reestruturações.
<b>Entradas</b>	Dados de novos sensores ou funcionalidades adicionadas
<b>Fonte</b>	Administração ou evolução da infraestrutura física e lógica da casa
<b>Saídas</b>	Integração e visualização fluida dos novos dados

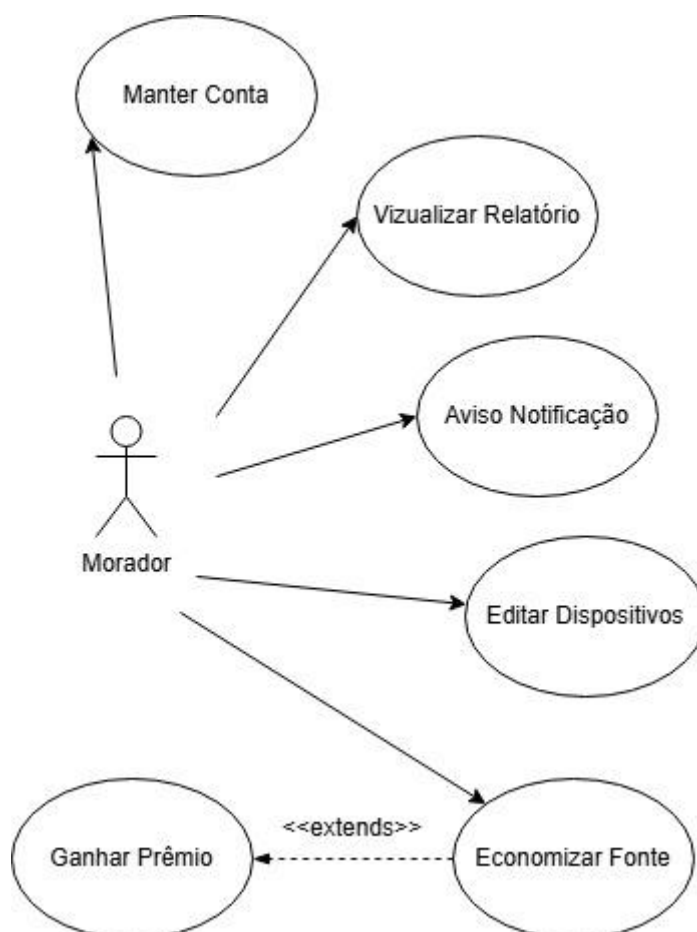
<b>Ação</b>	Utilizar arquitetura modular para permitir crescimento contínuo sem comprometer o desempenho.
-------------	---

<b>RFS06</b>	
<b>Função</b>	Armazenamento Seguro de Dados
<b>Descrição</b>	O sistema deve armazenar dados de sensores e interações do usuário de forma segura, com backups automáticos e controle de acesso baseado em autenticação.
<b>Entradas</b>	O sistema deve armazenar dados de sensores e interações do usuário de forma segura, com backups automáticos e controle de acesso baseado em autenticação.
<b>Fonte</b>	Fluxos de entrada do sistema
<b>Saídas</b>	Dados protegidos contra perdas ou acessos não autorizados
<b>Ação</b>	Implementar políticas de backup regular, criptografia e autenticação para garantir a integridade e privacidade das informações.

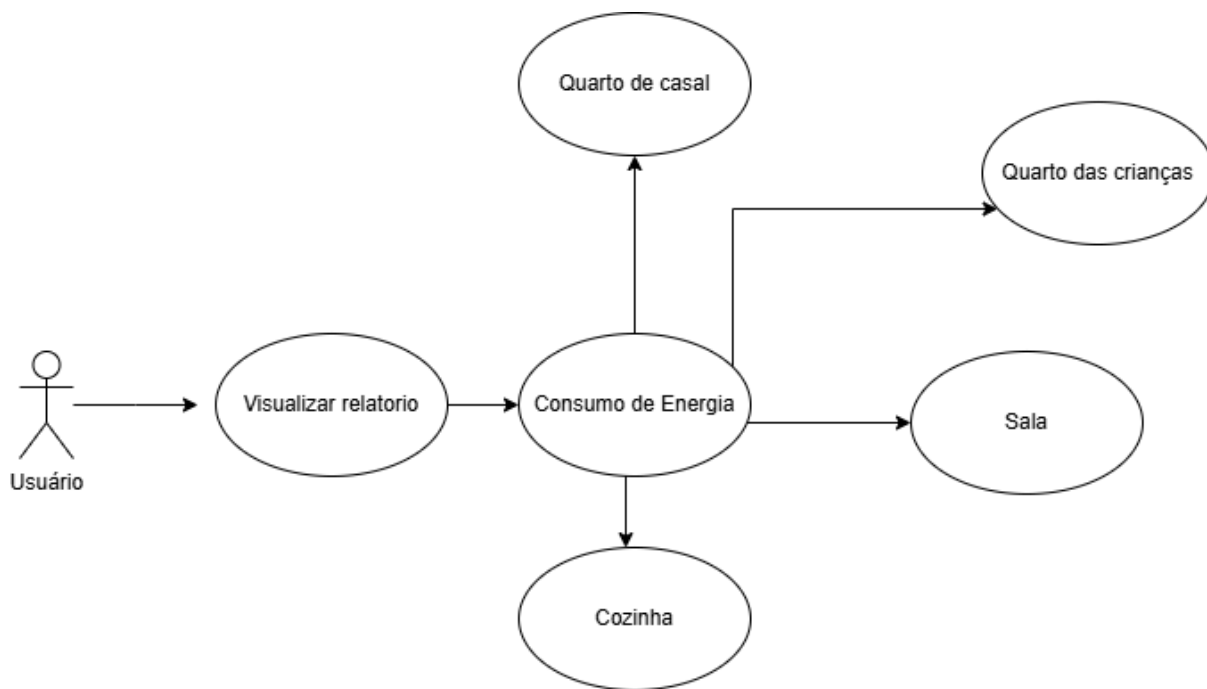


## 5. CASOS DE USO

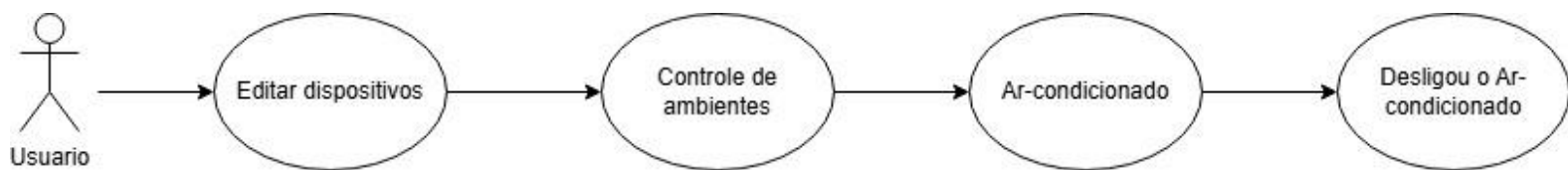
### Imagens



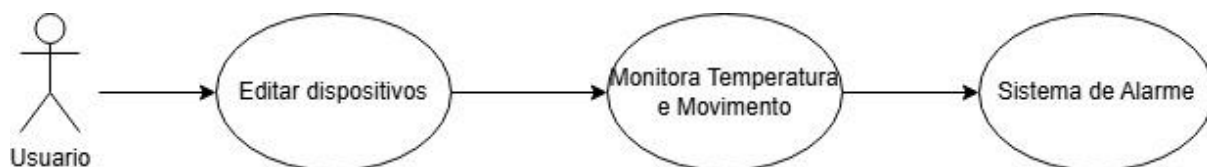
## Exemplo de Caso de uso Imagem 1



## Exemplo de Caso de uso Imagem 2



## Exemplo de Caso de uso Imagem 3



## Caso de Uso 01: Visualizar Consumo de Energia por Ambiente

Identificador: C.U 01

Visualizar consumo energético por ambiente

Atores: Usuário autenticado

Descrição: O usuário acessa a dashboard e visualiza gráficos atualizados com o consumo de energia de cada ambiente da casa (quartos, sala, cozinha, piscina).

Pré-condições: Usuário deve estar autenticado no sistema.

Fluxo principal:

1. Usuário acessa a aba “Consumo de Energia”.
2. O sistema carrega os dados em tempo real dos sensores.
3. Os dados são agrupados por ambiente com base no ID do sensor.
4. O sistema exibe os gráficos de consumo individual e total.
5. O usuário pode filtrar os dados por período (dia, semana, mês).

Pós-condições: Os gráficos são atualizados com base nos filtros aplicados.

Exceções:

- Falha na conexão com os sensores → exibir mensagem de erro.
  - Falta de dados para o período → exibir aviso informando ausência de dados.
- 

## Caso de Uso 02: Controlar Dispositivos da Casa

Identificador: C.U 02

Nome: Ativar ou desativar dispositivos por ambiente

Atores: Usuário autenticado

Descrição: O usuário pode controlar dispositivos dos ambientes (luz, ar-condicionado, bomba da piscina) diretamente pela dashboard.

Pré-condições: O usuário deve estar logado e o ambiente deve possuir dispositivos integrados ao sistema.

Fluxo principal:

1. Usuário acessa a aba “Controles”.
2. O sistema exibe os ambientes com seus dispositivos disponíveis.
3. Usuário ativa ou desativa um dispositivo com um botão.
4. O sistema envia o comando ao dispositivo correspondente.

5. O status é atualizado visualmente na interface.

Pós-condições: O dispositivo responde ao comando e o status fica visível.

Exceções:

- Dispositivo sem conexão → exibir status “offline” e bloquear ação.
  - Comando falhou → exibir notificação de erro.
- 

## **Caso de Uso 03: Monitorar Temperatura e Movimento**

Identificador: C.U 03

Nome: Acompanhar dados ambientais

Atores: Usuário autenticado

Descrição: O usuário visualiza, em tempo real, os dados de temperatura, umidade e detecção de movimento em cada ambiente.

Pré-condições: Usuário autenticado e sensores ativos.

Fluxo principal:

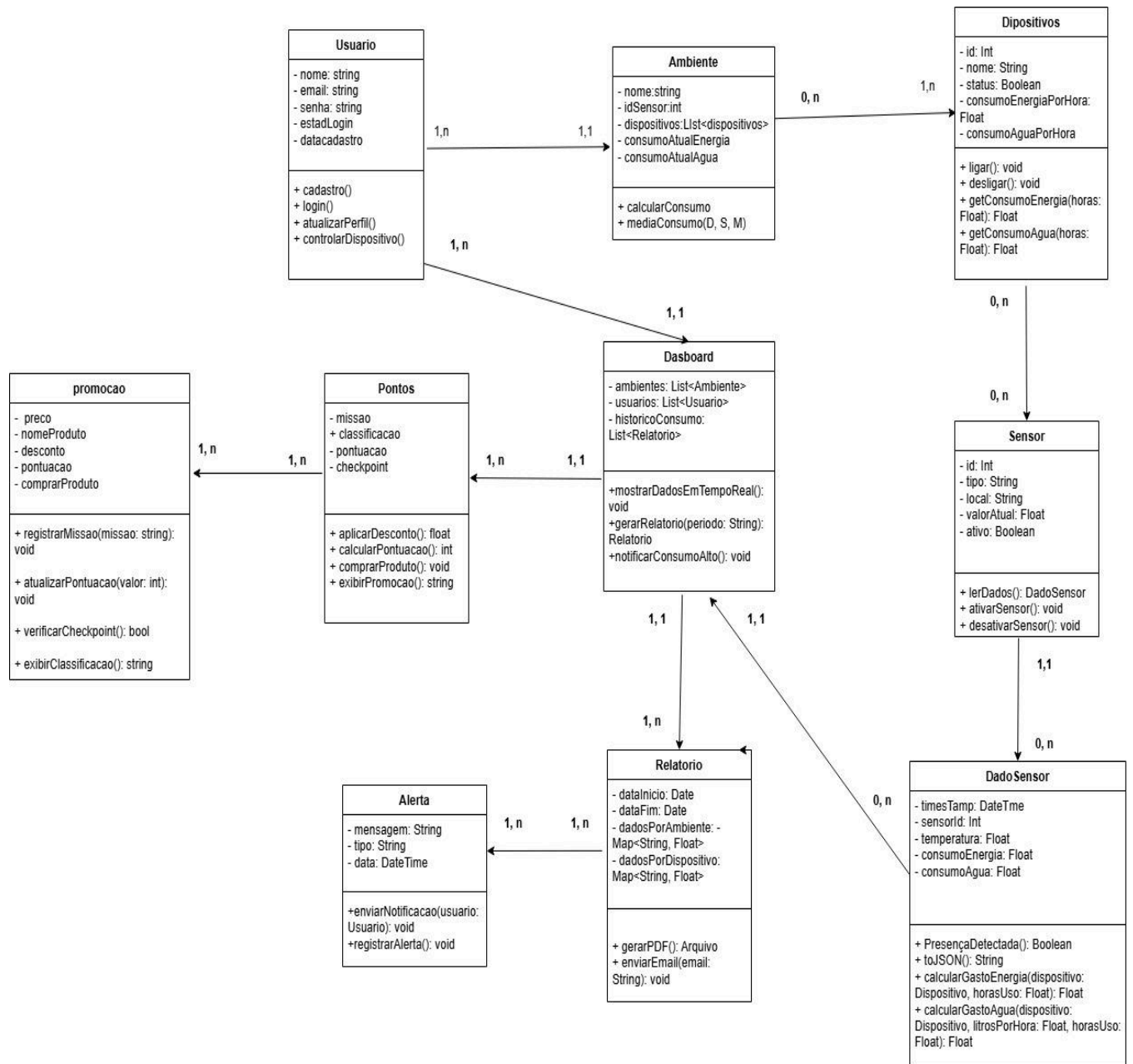
1. Usuário acessa a aba “Ambientes”.
2. O sistema exibe a lista de ambientes com temperatura, umidade e status de movimento.
3. O sistema atualiza os dados automaticamente.
4. O usuário pode expandir um ambiente para ver o histórico.

Pós-condições: Dados atualizados constantemente e acessíveis.

Exceções:

- Sensor desconectado → exibir último valor registrado e alerta.
- Dados corrompidos → ocultar valor e registrar erro no backend.

## 6. DIAGRAMA DE CLASSE



# 7. ARQUITETURA DO SISTEMA

A arquitetura do sistema de monitoramento e controle para casa inteligente foi projetada com foco em modularidade, escalabilidade e facilidade de integração com sensores físicos, bancos de dados em nuvem e interface de usuário amigável.

## 7.1 Visão Geral da Arquitetura

O sistema adota uma arquitetura baseada em camadas, com os seguintes componentes principais:

- **Camada de Sensoriamento (IoT):**  
Conjunto de sensores distribuídos pelos ambientes da casa (quartos, sala, cozinha, piscina), responsáveis pela coleta de dados de temperatura, umidade, movimento e consumo de energia.
- **Camada de Comunicação:**  
Os sensores transmitem os dados por meio de protocolo MQTT/HTTP para um gateway local ou diretamente para a nuvem.
- **Backend (Servidor / Banco de Dados):**  
Utiliza uma API central (por exemplo, hospedada no Supabase ou MongoDB) que armazena os dados recebidos e fornece endpoints para consultas e comandos.
  - Banco de dados relacional armazena os registros de sensores, consumo energético, controle de dispositivos e dados dos usuários.
  - Serviços de autenticação e autorização.
- **Frontend (Dashboard Web):**  
Desenvolvido com tecnologias modernas como, React, Vue.js, TailWind. Responsável por exibir:
  - Gráficos de consumo por ambiente.
  - Controles de dispositivos (liga/desliga).
  - Visualização de temperatura, umidade e movimento.
  - Menu lateral com navegação entre módulos.
- **Camada de Controle:**  
Permite que comandos do usuário sejam convertidos em ações nos dispositivos da casa (ex: ligar ar-condicionado, desativar bomba da piscina).

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

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

**ANDRADE, Joheván de Gois.** *Estudo e projeto de transformação de uma casa tradicional numa casa inteligente, baseada em IoT*. 2024. Dissertação (Mestrado) – Universidade da Madeira, Portugal, 2024. Disponível em: ProQuest Dissertations & Theses. Acesso em: 12 abr. 2025.

**INTELBRAS.** *Casa inteligente: entenda o conceito e como funciona*. Disponível em: <https://blog.intelbras.com.br/casa-inteligente/>. Acesso em: 12 abr. 2025.

**COSTA, Arthur Linhares; ALCANTARA, Lucas Laranja; ZIPINOTTI, Rafael Loss; ALMEIDA, Rayne Cruz de; PEREIRA, Victor Oliveira; SOUZA, Francisca Katia de Barbosa de; FERREIRA, Jamilli Ricarto.** *Casa inteligente*. Vila Velha - ES: Centro Estadual de Educação Técnica Vasco Coutinho, [s.d.]. Trabalho acadêmico – Segundo Módulo do Ensino Técnico em Informática.