

PROJETO.....**Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas****INTEGRANTES DO PROJETO e RA'S**

André Luis De Sousa Rodrigues	-	25027358
Nelson Dos Reis Gomes Souza	-	25027592
Nicolas hayato Nitta	-	25027686
Samuel Cavalcanti Nunes Coronel	-	25027802

São Paulo

2025

Sumário

1 INTRODUÇÃO	3
2. DOCUMENTO DE ABERTURA DO PROJETOS.....	4
2.1 – Project Charter	4
2.2 – Histórias do Usuário	5
3. DESIGN SPRINT – Ideação e prototipação do desafio	6
3.1 Desafio.....	6
3.2 Entender Mapear	6
3.3 Ideação – desenho da solução (trilha do usuário)	6
3.4 Prototipagem.....	6
4. REQUISITOS DE SISTEMA.....	6
4.1 REQUISITOS FUNCIONAIS DE SOFTWARE	6
4.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE	10
5. CASOS DE USO	13
6. DIAGRAMA DE CLASSE.....	13
7. ARQUITETURA DO SISTEMA	13
8. REFERÊNCIAS BIBLIOGRÁFICAS.....	13

1 INTRODUÇÃO

Tendo.....

Incluir o documento do Projeto do PI – 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 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

Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.

Introdução

Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.

Glossário

Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.

Definição de requisitos de usuário

Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.

Arquitetura do sistema

Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.

Especificação de requisitos do sistema

Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.

Modelos do sistema

Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.

Evolução do sistema

Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.

Apêndices

Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.

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	Umididade	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	Permitir o registro de usuários
Descrição	O sistema deve permitir o registro de usuários com dados mínimos

Entradas	Nome, e-mail, senha
Fonte	Interface de cadastro de usuário
Saídas	Confirmação de registro, mensagem de erro em caso de falha
Ação	Armazena os dados do usuário no banco de dados após validação

RFS02	
Função	Coletar dados de consumo de energia
Descrição	O sistema deve permitir o registro do consumo de energia elétrica do usuário, com dados mínimos necessários
Entradas	Leitura do medidor (valor em kWh), data da leitura
Fonte	Interface de coleta de dados de consumo
Saídas	Confirmação do registro, mensagem de erro em caso de falha
Ação	Armazena os dados de consumo no banco de dados após validação (por exemplo, verificar se a leitura do medidor está dentro de um intervalo válido).
RFS03	
Função	Exibir consumo de energia em intervalos diários, semanais e mensais
Descrição	O sistema deve calcular e exibir o consumo de energia elétrica para diferentes intervalos de tempo (diário, semanal e mensal).
Entradas	Dados de consumo registrados (valor de kWh e data da leitura)
Fonte	Banco de dados ou interface de consulta de consumo de energia

Saídas	Consumo diário (kWh)
	Consumo semanal (kWh)
	Consumo mensal (kWh)
	Mensagem de erro caso não haja dados suficientes ou válidos
Ação	<p>Para o consumo diário: Calcula o total de energia consumida no dia específico, somando todas as leituras feitas nesse dia.</p> <p>Para o consumo semanal: Soma os dados de consumo da semana, levando em consideração a data da leitura e agrupando os dados pela semana (usualmente começando no domingo ou segunda-feira).</p> <p>Para o consumo mensal: Soma os dados de consumo do mês, agrupando as leituras pela data de registro (mês e ano).</p>

RFS04	
Função	Exibir consumo de energia e metas/desafios
Descrição	O sistema deve calcular e exibir o consumo de energia elétrica para diferentes intervalos de tempo (diário, semanal e mensal), além de apresentar metas de consumo e desafios relacionados ao uso responsável de energia.
Entradas	Dados de consumo registrados (valor de kWh e data da leitura)
Fonte	Banco de dados ou interface de consulta de consumo de energia
Saídas	Consumo diário (kWh)
	Consumo semanal (kWh)
	Consumo mensal (kWh)
	Metas de consumo (diárias, semanais e mensais)
	Desafios para reduzir o consumo
	Mensagem de erro caso não haja dados suficientes ou válidos

<p>Ação</p>	<p>Consumo diário, semanal e mensal: Calcular e exibir o total de consumo para os períodos solicitados.</p> <p>Metas de consumo: Comparar o consumo real com uma meta predefinida para cada intervalo (diário, semanal e mensal).</p> <p>Desafios: Criar desafios para o usuário com base no consumo (por exemplo, reduzir o consumo diário em 10%, alcançar uma meta de redução no mês, etc.).</p>
--------------------	---

RFS05	
<p>Função</p>	<p>Coletar dados de consumo de energia e atribuir pontos</p>
<p>Descrição</p>	<p>O sistema deve permitir o registro de consumo de energia do usuário, calcular se as metas de consumo foram atingidas e conceder pontos por isso. Também deve incluir desafios de redução de consumo com recompensa de pontos extras.</p>
<p>Entradas</p>	<p>Leitura do medidor (valor em kWh)</p> <p>Data da leitura</p>
<p>Fonte</p>	<p>Interface de coleta de dados de consumo (aplicativo ou website)</p>
<p>Saídas</p>	<p>Consumo diário, semanal e mensal (em kWh)</p> <p>Metas de consumo diárias, semanais e mensais</p> <p>Desafios para redução do consumo</p> <p>Pontos ganhos conforme atingimento de metas e desafios</p>
<p>Ação</p>	<p>Armazena os dados de consumo no banco de dados após validação</p> <p>Calcula o consumo total diário, semanal e mensal</p> <p>Compara o consumo com as metas e atribui pontos</p> <p>Oferece desafios para redução de consumo com recompensa em pontos</p>

--	--

RFS06	
Função	Exibir e permitir a troca de pontos por recompensas
Descrição	O sistema permite que o usuário acesse a loja de recompensas e troque seus pontos por benefícios, como descontos ou produtos virtuais.
Entradas	Pontuação acumulada do usuário Lista de recompensas disponíveis (por exemplo: descontos, produtos, badges)
Fonte	Sistema de pontos do usuário
Saídas	Itens disponíveis para troca Confirmação da troca de recompensa Mensagem de erro caso o usuário não tenha pontos suficientes
Ação	Exibe as recompensas disponíveis Permite ao usuário trocar seus pontos por uma recompensa Atualiza a pontuação do usuário após a troca

4.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

RFS01	
Função	responsivo
Descrição	Funcionando bem em celulares e navegadores
Entradas	

Fonte	
Saídas	
Ação	Tem que rodar bem em todos os dispositivos

RFS02	
Função	Alta usabilidade
Descrição	Cores acessíveis, textos claros
Entradas	
Fonte	
Saídas	Saída inferior a 2 segundos em conexões lentas
Ação	

RFS03	
Função	Ser rápido
Descrição	Tempo de resposta inferior a 2 segundos
Entradas	
Fonte	
Saídas	
Ação	Ser rápido e menor que 2 segundos

RFS04	
Função	Ser escalável
Descrição	Sistema deve ser escalável, podendo ser replicável em outras cidades
Entradas	
Fonte	
Saídas	
Ação	Ser escalável e adaptável

RFS05	
Função	Ter segurança
Descrição	Plataforma deve garantir segurança do usuário
Entradas	
Fonte	
Saídas	
Ação	Garantir a segurança

RFS06	
Função	Ser tecnológico e ser econômico em recurso
Descrição	Vai garantir que o computador consuma menos memória e processamento
Entradas	
Fonte	
Saídas	
Ação	Otimizar o APP

5. CASOS DE USO

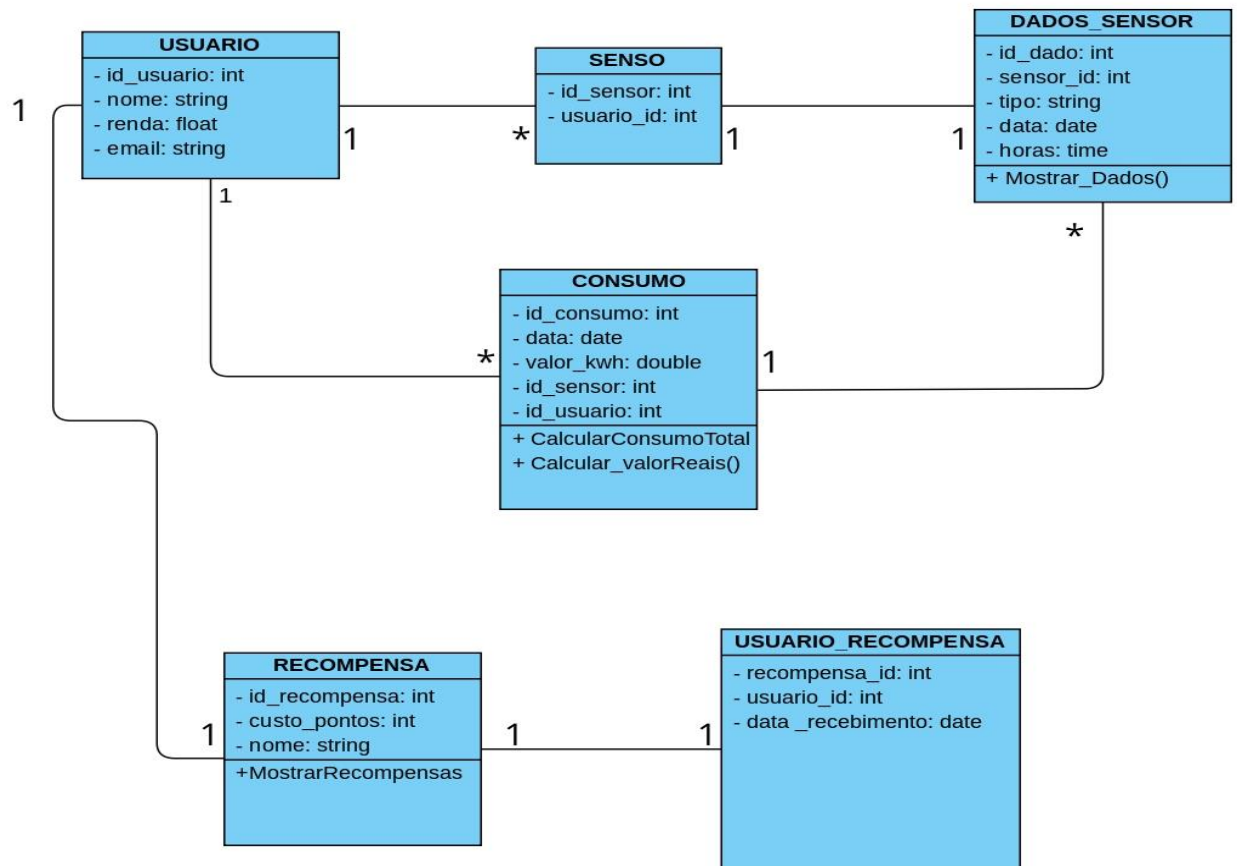
Apresentar 3 casos de uso do sistema

Casas Inteligentes

Bairros Inteligentes

Controle de energia

6. DIAGRAMA DE CLASSE



7. ARQUITETURA DO SISTEMA

1. Camada de Interface (Windows Forms):

- Application desktop desenvolvido em Windows Forms usando C# e .NET Framework.
- Responsável por fornecer uma interface gráfica amigável ao usuário, exibindo dados de energia em tempo real, gráficos, indicadores e opções de controle.

2. Camada de Lógica de Negócio:

- Implementada no código C# dentro do aplicativo Windows Forms.
- Processa os dados recebidos, realiza cálculos ou análises necessárias, e gerencia a comunicação entre a interface e a origem de dados.

3. Comunicação com o Hardware ou Sensor de Energia:

- O sistema pode se comunicar com sensores ou dispositivos IoT via

protocolos específicos (por exemplo, MQTT, MQTT-SH, HTTP, ou serial).

- Essa camada coleta os dados de consumo energético, enviando-os periodicamente ou em tempo real.

4. Camada de Comunicação com o Servidor Web:

- O aplicativo Windows Forms envia ou recebe dados de um servidor web local ou remoto, utilizando APIs REST ou WebSockets para atualização em tempo real.
- Essa comunicação pode ser feita usando componentes HTTPClient, WebSocket, ou bibliotecas específicas em C#.

8. REFERÊNCIAS BIBLIOGRÁFICAS

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