

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

Kenji Fernandes Wakabayashi	-	22011039
João Guilherme Gumiero de Micheli	-	22011008
João Flavio do Rio Tozzi	-	25026998
Kesley Silva Damascena	-	22011008

São Paulo

2025

Sumário

1 INTRODUÇÃO	3
2. DOCUMENTO DE ABERTURA DO PROJETOS.....	4
2.1 – Project Charter.....	4
2.2 – Histórias do Usuário	7
3. DESIGN SPRINT – Ideação e prototipação do desafio.....	8
3.1 Desafio	8
3.2 Entender Mapear	8
3.3 Ideação – desenho da solução (trilha do usuário)	8
3.4 Prototipagem.....	8
4. REQUISITOS DE SISTEMA	8
4.1 REQUISITOS FUNCIONAIS DE SOFTWARE	8
4.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE	11
5. CASOS DE USO	14
6. DIAGRAMA DE CLASSE.....	14
7. ARQUITETURA DO SISTEMA	14
8. REFERÊNCIAS BIBLIOGRÁFICAS	14

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

Estamos construindo um sistema de interação de usuários para com uma O sistema visa fornecer uma interface para interação com uma casa inteligente, incluindo controle de temperatura, segurança com câmeras, controle de energia e controle de luzes. O documento não detalha explicitamente a integração com outros sistemas, mas menciona a possibilidade de integração com assistentes virtuais (Alexa, Google Assistant, etc.) e serviços de terceiros (previsão do tempo, notícias, etc.). O sistema atende aos objetivos de proporcionar controle e monitoramento residencial, economia de energia e automação de tarefas domésticas.

Glossário

- IOT: Internet das Coisas. Dispositivos físicos que se conectam e trocam dados com outros dispositivos e sistemas pela Internet.
- Painel de Controle: Interface do usuário para monitorar e controlar os dispositivos da casa inteligente.
- Sensores: Dispositivos que detectam e medem grandezas físicas ou químicas, convertendo-as em sinais elétricos.
- Interface Intuitiva: Interface fácil de usar e entender, com navegação clara e design amigável.
- Design Responsivo: Capacidade da interface de se adaptar a diferentes tamanhos de tela e dispositivos.
- Autenticação Segura: Processo de verificação da identidade do usuário para garantir acesso autorizado ao sistema.
- Criptografia: Processo de codificação de dados para proteger a confidencialidade das informações.
- PTZ: Pan-Tilt-Zoom. Capacidade de câmeras de segurança de mover a lente horizontalmente, verticalmente e ajustar o zoom.
- Dimmer: Dispositivo que permite ajustar a intensidade da luz.

- Alexa, Google Assistant: Assistentes virtuais que podem ser integrados ao sistema para controle por voz.
- Zigbee, Z-Wave: Protocolos de comunicação sem fio usados em dispositivos de automação residencial.

Definição de requisitos de usuário

Requisitos Funcionais: Gráfico de Controle de Temperatura:

Exibir um gráfico em tempo real da temperatura ambiente. Permitir o ajuste da temperatura desejada.

Mostrar o histórico de temperaturas por período (diário, semanal, mensal). Exibir a temperatura de diferentes cômodos, caso existam sensores individuais.

Sistema de Segurança com Câmeras:

Exibir feeds de vídeo ao vivo das câmeras de segurança.

Permitir a visualização de gravações anteriores.

Fornecer alertas de detecção de movimento ou som.

Permitir o controle de câmeras PTZ (Pan-Tilt-Zoom), se aplicável.

Possibilidade de notificações de alerta no celular.

Gráfico de Controle de Energia:

Exibir o consumo de energia em tempo real da casa.

Mostrar o consumo de energia de cada aparelho conectado.

Gerar relatórios de consumo por período (diário, semanal, mensal). Permitir o controle remoto de aparelhos (ligar/desligar).

Possibilidade de identificação de aparelhos com consumo excessivo.

Controle de Luzes:

Exibir o status das luzes de cada área da casa (ligado/desligado). Permitir o controle individual ou em grupo das luzes.

Permitir o ajuste da intensidade da luz (dimmer), se aplicável. Criar cenários de iluminação personalizados (ex: "modo filme", "modo

leitura").

Possibilidade de criação de rotinas, como por exemplo, acender as luzes ao anoitecer.

Interface do Usuário:

Interface intuitiva e fácil de usar.

- Design responsivo para diferentes dispositivos (computadores, tablets, smartphones).

Personalização da interface (layout, cores, etc.).

Possibilidade de controle por comando de voz.

Possibilidade de criação de perfis de usuários.

Segurança:

- Autenticação segura de usuários.

Criptografia de dados sensíveis.

Atualizações de segurança regulares.

Possibilidade de autenticação de dois fatores. Requisitos Não Funcionais: Desempenho:

Respostas rápidas e sem atrasos.

Baixo consumo de recursos do sistema.

Disponibilidade:

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	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	Exibir um gráfico em tempo real da temperatura ambiente.
Descrição	O sistema deve informar por meio de um gráfico em tempo real as alterações e situação atual da

	temperatura do ambiente, informando graus de temperatura em Celsius e Fahrenheit, horário e dia.
Entradas	Temperatura detectada pelo sensor, e horário.
Fonte	Sensor de temperatura.
Saídas	Temperatura local e horário registrado.
Ação	Salvar informações de temperatura e horário no banco de dados e expor no gráfico

RFS02	
Função	Permitir o ajuste da temperatura desejada.
Descrição	O sistema deve permitir a alteração da temperatura desejada pelo usuário, possuindo limite de máximo e mínimo correspondente a capacidade do ar-condicionado e/ou aquecedores.
Entradas	Temperatura desejada informada na dashboard pelo usuário.
Fonte	Dashboard.
Saídas	Temperatura desejada e horário do momento da alteração.
Ação	Configurar ar-condicionado e/ou aquecedor para a temperatura desejada do usuário. Salvar temperatura e horário do ajuste feito pelo usuário.

RFS03	
Função	Mostrar o histórico de temperaturas por período (diário, semanal, mensal).
Descrição	O sistema deve registrar e mostrar as temperaturas em graus Celsius e Fahrenheit registradas nos períodos diários, semanais e mensais, junto de seus horários e alterações realizadas pelos usuários.
Entradas	Temperaturas do ambiente capturadas pelo sensor e seu horário.
Fonte	Dashboard e Sensor de Temperatura.
Saídas	Histórico de temperaturas do ambiente junto do horário, dia e mês e alterações realizadas pelo usuário.
Ação	Exibir ao usuário as temperaturas capturadas pelo sensor em determinado período, junto do horário, dia e mês, e alterações realizadas pelo usuário.

RFS04	
Função	Exibir a temperatura de diferentes cômodos
Descrição	O sistema deve permitir que o usuário visualize a temperatura atual de diferentes cômodos da residência.
Entradas	O usuário pode selecionar um cômodo específico para visualizar sua temperatura. Caso nenhuma seleção seja

	feita, o sistema pode exibir a temperatura de todos os cômodos monitorados. A entrada pode ser feita através de uma interface gráfica com botões, lista suspensa ou outra forma de seleção.
Fonte	Sensores de temperatura instalados nos cômodos.
Saídas	A temperatura atual de cada cômodo deve ser exibida na interface do usuário. A temperatura deve ser apresentada de forma clara e legível, incluindo a unidade de medida e horário.
Ação	As leituras de temperatura recebidas devem ser armazenadas no banco de dados, e exibidas na interface, associadas ao cômodo correspondente e ao horário da leitura.

RFS05	
Função	Exibir feeds de vídeo ao vivo das câmeras de segurança
Descrição	O sistema deve permitir que o usuário visualize os feeds de vídeo em tempo real das câmeras de segurança instaladas em diferentes locais.
Entradas	O usuário pode selecionar uma ou mais câmeras específicas para visualizar seus feeds de vídeo ao vivo. A entrada deve ser feita através de uma interface gráfica com miniaturas das câmeras, uma lista ou um mapa com a localização das câmeras.
Fonte	Câmeras de Segurança IP
Saídas	O fluxo de vídeo em tempo real de cada câmera selecionada deve ser exibido na interface do usuário, incluindo o nome ou localização da câmera referente. O vídeo deve ser reproduzido de forma contínua com o menor atraso possível. A interface deve fornecer controles básicos de visualização, como pausar/reproduzir, ajustar o volume (se a câmera tiver áudio), ou alternar para tela cheia.
Ação	O sistema deve estabelecer e manter conexões com os fluxos de vídeo em tempo real de cada câmera de segurança configurada. Isso pode envolver a utilização de protocolos de streaming de vídeo. O sistema deve receber os dados de vídeo transmitidos pelas câmeras e processá-los para exibição na interface do usuário.

RFS06	
Função	Permitir a visualização de gravações anteriores
Descrição	O sistema deve permitir que o usuário navegue e visualize gravações de vídeo capturadas pelas câmeras de segurança em datas e horários anteriores.

Entradas	O usuário deve especificar a câmera e o período de tempo desejado para visualização. Isso pode ser feito através da seleção de uma data e, opcionalmente, um horário de início e fim. O usuário deve ter controles para navegar pelas gravações.
Fonte	Servidores ou banco de dados onde as gravações de vídeo das câmeras são armazenadas.
Saídas	O vídeo gravado para a câmera e período selecionados deve ser exibido na interface do usuário.
Ação	O sistema deve se conectar ao sistema de armazenamento de vídeo para buscar as gravações disponíveis para a câmera selecionada. O sistema deve decodificar o arquivo de vídeo gravado e exibi-lo na interface do usuário, permitindo ao usuário pausar, reproduzir, avançar, retroceder e navegar pela timeline da gravação.

4.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

RFS01	
Função	Respostas rápidas e sem atrasos.
Descrição	O sistema deve fornecer respostas e executar ações de forma rápida e eficiente, minimizando a percepção de atraso para o usuário.
Entradas	Qualquer interação do usuário que demande uma resposta do sistema.
Fonte	Servidores, rede, banco de dados, câmeras de segurança e sensores.
Saídas	A interface do usuário deve atualizar-se rapidamente em resposta às ações do usuário, fornecendo feedback visual quase instantâneo. As funcionalidades do sistema devem ser executadas de forma rápida, sem que o usuário precise esperar por longos períodos.
Ação	Implementar técnicas de otimização em todos os níveis do sistema, incluindo código, consultas ao banco de dados, comunicação de rede e processamento de vídeo.

RFS02	
Função	Baixo consumo de recursos do sistema.

Descrição	O sistema deve operar de forma eficiente, utilizando o mínimo possível de recursos computacionais, como CPU, memória RAM, espaço de armazenamento e largura de banda de rede, sem comprometer a funcionalidade e o desempenho dentro dos limites aceitáveis.
Entradas	Número de Usuários Conectados, Câmeras Ativas, Sensores Ativos, Nível de Atividade do Usuário e Período de Tempo de Operação.
Fonte	A eficiência do código, algoritmos utilizados e gerenciamento de memória.
Saídas	<p>O processamento do sistema não deve sobrecarregar a unidade central de processamento dos servidores ou dispositivos do usuário.</p> <p>O sistema deve utilizar a memória de forma otimizada, evitando vazamentos de memória e o consumo excessivo que possa levar à lentidão ou instabilidade.</p> <p>As gravações de vídeo e outros dados do sistema devem ser armazenados de forma eficiente, utilizando compressão e políticas de retenção adequadas para minimizar o espaço necessário.</p> <p>A transmissão de feeds de vídeo e outros dados pela rede deve ser otimizada para reduzir o consumo de largura de banda, especialmente em cenários com múltiplas câmeras ou acesso remoto.</p>
Ação	<p>Escrever código eficiente, evitando operações desnecessárias e utilizando estruturas de dados adequadas.</p> <p>Utilizar codecs de vídeo eficientes para reduzir o tamanho dos arquivos de gravação e a largura de banda necessária para a transmissão.</p> <p>Implementar mecanismos para liberar recursos não utilizados (memória, conexões) de forma oportuna.</p> <p>Transmitir apenas os feeds de vídeo que estão sendo visualizados ativamente pelo usuário.</p>

RFS03	
Função	Sistema disponível 24 horas por dia, 7 dias por semana.
Descrição	O sistema deve estar operacional e acessível aos usuários de forma contínua, ininterrupta, 24 horas por dia e 7 dias por semana, sem períodos de inatividade planejados ou não planejados que impeçam o acesso às suas funcionalidades principais.
Entradas	<p>Usuários podem tentar acessar o sistema a qualquer momento, em qualquer dia da semana.</p> <p>O sistema de segurança requer vigilância contínua, tornando a indisponibilidade inaceitável.</p>

Fonte	Servidores robustos, Conexões de rede estáveis e redundantes, Sistema de banco de dados de alta disponibilidade.
Saídas	Os usuários podem fazer login e interagir com o sistema a qualquer momento. Os feeds de vídeo das câmeras estão sempre acessíveis para visualização.
Ação	Configurar sistemas para detectar falhas e transferir automaticamente as operações para sistemas de backup sem intervenção manual. Realizar backups regulares dos dados do sistema para permitir a recuperação em caso de falhas catastróficas.

RFS04	
Função	Tempo de inatividade mínimo.
Descrição	O sistema deve ser projetado e mantido de forma a minimizar o tempo total em que ele fica inacessível ou não funcional para os usuários.
Entradas	Problemas com servidores, câmeras, dispositivos de armazenamento, rede, bugs, erros de programação, problemas de configuração, problemas de conectividade local ou com a internet, quedas de energia, atualizações de software, substituição de hardware, testes de recuperação de desastres.
Fonte	Qualidade e confiabilidade dos códigos e componentes de hardware. Capacidade de detectar e diagnosticar problemas proativamente.
Saídas	A frequência com que o sistema fica indisponível. O tempo médio que o sistema permanece inativo por incidente.
Ação	Configurar sistemas para transferir automaticamente as operações para backups em caso de falha.

RFS05	
Função	Sistema livre de erros e falhas.
Descrição	O sistema deve operar de forma consistente e confiável, executando suas funcionalidades conforme o esperado e sem apresentar erros, falhas ou comportamentos inesperados que possam comprometer a integridade dos dados, a funcionalidade ou a experiência do usuário.
Entradas	Todas as formas de interação previstas com o sistema.
Fonte	Código bem escrito, claro, modular e aderente a boas práticas de programação.
Saídas	Todas as funcionalidades do sistema operam conforme o especificado nos requisitos.

Ação	Adotar padrões de codificação que minimizem a introdução de vulnerabilidades e erros.
-------------	---

RFS06	
Função	Recuperação automática de falhas
Descrição	O sistema deve ser capaz de detectar automaticamente a ocorrência de falhas em seus componentes (hardware ou software) e iniciar processos de recuperação predefinidos sem a necessidade de intervenção manual imediata, visando restaurar a operação normal com o mínimo de interrupção possível.
Entradas	Detecção de mau funcionamento em servidores, câmeras, sensores, armazenamento, rede, etc. Identificação de erros críticos, exceções não tratadas, processos travados. Detecção de interrupção na comunicação de rede com componentes internos ou externos.
Fonte	Software e hardware dedicados a monitorar a saúde e o status dos componentes do sistema.
Saídas	O componente ou serviço com falha retorna ao estado operacional normal. O sistema principal continua funcionando, mesmo que alguns componentes tenham falhado e se recuperado.
Ação	Configurar sistemas de monitoramento para verificar todos os componentes críticos.

5. CASOS DE USO

Apresentar 3 casos de uso do sistema

6. DIAGRAMA DE CLASSE

7. ARQUITETURA DO SISTEMA

8. REFERÊNCIAS BIBLIOGRÁFICAS

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