

Especificação de Requisitos

DS 2023/2 - Grupo 5

Mateus Cordova, Andrei Bueno, Kalani Sosa

# Sistemas Integrados para Eficiência Energética em Ambientes Educacionais

## 1. Introdução

### 1.1 Propósito do Documento

A crescente preocupação com a eficiência energética nas instituições de ensino destaca a necessidade de abordar questões específicas relacionadas ao consumo de energia em salas de aula. Este projeto visa contribuir para a mitigação desses desafios, desenvolvendo um sistema integrado que permite a medição precisa e o gerenciamento eficiente do uso de energia, temperatura e luminosidade em salas de aula.

### 1.2 Escopo do Projeto

O objetivo principal é a criação de um protótipo de hardware capaz de coletar dados em tempo real sobre diversas condições ambientais em salas de aula, incluindo consumo de energia, temperatura e nível de luminosidade. Esses dados serão acessados por meio de uma API desenvolvida em Python, oferecendo flexibilidade e facilidade de integração com outros sistemas. A coleta contínua de dados em tempo real possibilitará uma resposta rápida a anomalias, garantindo ambientes mais confortáveis e eficientes, a API em Python oferecerá a flexibilidade necessária para integrar o sistema com outras soluções e ampliar suas funcionalidades conforme as demandas evoluem.

O sistema proposto permitirá uma gestão mais eficiente do consumo de energia em salas de aula, contribuindo para a redução dos custos operacionais e para a promoção de práticas sustentáveis.

Este projeto visa enfrentar o desafio da eficiência energética em ambientes educacionais, oferecendo uma solução integrada que combina hardware inovador, uma interface de programação robusta e um painel web intuitivo. A implementação bem-sucedida deste sistema promoverá não apenas a otimização do consumo de energia, mas também a conscientização ambiental e a busca por práticas mais sustentáveis no contexto acadêmico.

## 2. Descrição Geral

### 2.1 Perspectiva do Produto

O sistema proposto para Eficiência Energética em Ambientes Educacionais está integrado a um contexto mais amplo de tecnologias e sistemas educacionais. A principal interface externa é estabelecida com o hardware baseado em Arduino, responsável pela coleta de dados ambientais em salas de aula. A API Python atua como uma ponte de comunicação entre o hardware e o Dashboard, proporcionando uma interconexão eficiente e segura. Além disso, consideramos a perspectiva de futuras integrações com sistemas de gestão escolar e outros dispositivos IoT que possam enriquecer a experiência e funcionalidades do sistema.

### 2.2 Funções do Produto

O produto oferecerá funcionalidades abrangentes, incluindo:

- Coleta precisa e em tempo real de dados relacionados ao consumo de energia, temperatura e luminosidade em salas de aula.
- Transmissão segura desses dados por meio da API Python para o Dashboard.
- Interface intuitiva do Dashboard, apresentando informações em tempo real e histórico por meio de gráficos e tabelas.
- Personalização do Dashboard para atender às preferências dos usuários.
- Sistema de alertas e notificações para eventos críticos.
- Suporte à exportação de dados para análises externas.

### Desenvolvimento do Hardware:

Será projetado e implementado um dispositivo de hardware especializado para coletar dados relevantes, garantindo precisão e confiabilidade nas medições. O foco será na facilidade de instalação e operação, permitindo uma aplicação abrangente em diversos ambientes educacionais.

### API em Python:

Uma interface de programação de aplicativos (API) em Python será desenvolvida para facilitar o acesso e a integração dos dados coletados pelo hardware. Essa API será projetada para oferecer uma comunicação eficiente e segura, permitindo a expansão futura das funcionalidades conforme necessário.

### Panel Web (Dashboard):

Um painel web intuitivo será criado para permitir o gerenciamento centralizado das informações coletadas. Este dashboard proporcionará uma visão abrangente do consumo de energia, temperatura e luminosidade em diferentes salas de aula, além de apresentar gráficos e análises que auxiliem na tomada de decisões relacionadas à eficiência energética.

### 2.3 Características do Usuário

Os usuários finais do sistema incluem administradores escolares, responsáveis pela gestão eficiente de recursos, e professores, interessados no ambiente de ensino adequado para seus alunos. As características demográficas variam desde profissionais da educação com diferentes níveis de expertise técnica até administradores escolares com responsabilidades mais amplas. A interface do Dashboard será projetada para atender a diversas necessidades, desde usuários leigos até aqueles mais familiarizados com tecnologias educacionais.

### 2.4 Restrições

**Limitações Tecnológicas:** O sistema está sujeito às limitações tecnológicas do hardware Arduino e das capacidades de transmissão de dados disponíveis.

**Restrições Regulatórias:** Deve aderir às normas e regulamentações de proteção de dados e privacidade, garantindo conformidade com as leis locais e nacionais.

**Restrições de Negócio:** Limitações orçamentárias e de tempo podem impactar o escopo e as funcionalidades do sistema.

### 2.5 Suposições e Dependências

**Suposições:** Supõe-se que a infraestrutura escolar, como acesso à eletricidade e conectividade à internet, esteja disponível e funcional.

**Dependências Tecnológicas:** A eficácia do sistema depende da estabilidade das tecnologias Arduino, Python e do ambiente de execução do Dashboard.

**Colaboração com a Escola:** Dependência da cooperação ativa da instituição educacional para instalação e integração do hardware nas salas de aula.

Essas considerações são cruciais para garantir o sucesso do projeto, fornecendo uma compreensão abrangente da interação do sistema com o ambiente educacional e os usuários finais.

## 3. Definição do Escopo do Projeto

### 3.1 Objetivos do Projeto

O projeto visa atender às seguintes metas:

- **Desenvolver um Protótipo de Hardware:** Criar um dispositivo baseado em Arduino capaz de medir com precisão o consumo de energia, temperatura e luminosidade em salas de aula, proporcionando dados confiáveis para análise.
- **Implementar uma API Python:** Desenvolver uma interface de programação de aplicativos (API) que permita a comunicação eficiente entre o hardware e o Dashboard, garantindo a transmissão segura e contínua dos dados coletados.
- **Criar um Dashboard Intuitivo:** Desenvolver um painel web intuitivo que permita aos usuários visualizar em tempo real e analisar historicamente os dados de consumo de energia, temperatura e luminosidade, facilitando a tomada de decisões informadas.
- **Oferecer Personalização e Alertas:** Incorporar funcionalidades no Dashboard que permitam aos usuários personalizar a exibição de informações e receber alertas automáticos em caso de anomalias ou eventos críticos.

### 3.2 Funcionalidades Principais

As principais funcionalidades do sistema incluem:

- **Coleta de Dados em Tempo Real:** Permitir a coleta contínua e em tempo real de dados sobre consumo de energia, temperatura e luminosidade nas salas de aula, garantindo uma visão atualizada das condições ambientais.
- **Transmissão Segura de Dados:** Garantir uma transmissão segura e eficiente dos dados coletados do hardware para o Dashboard por meio da API Python, protegendo a integridade das informações.
- **Interface Intuitiva do Dashboard:** Desenvolver uma interface de usuário no Dashboard que seja intuitiva e de fácil compreensão, fornecendo gráficos claros e informações relevantes para os usuários.
- **Personalização da Exibição:** Permitir que os usuários personalizem a exibição do Dashboard de acordo com suas preferências, destacando informações específicas ou ajustando a visualização.
- **Sistema de Alertas Automáticos:** Implementar um sistema de alertas que notifica os usuários sobre eventos críticos ou anomalias, permitindo uma resposta rápida a situações que requerem atenção.

## **Análise de Requisitos/Funcionalidades para o Hardware utilizando Arduino:**

### **1. Precisão nas Medidas:**

- i. O hardware deve ser capaz de realizar medições precisas do consumo de energia, temperatura e nível de luminosidade em salas de aula.
- ii. Garantir a acurácia dos sensores incorporados para assegurar dados confiáveis.

### **2. Coleta em Tempo Real:**

- i. Capacidade de coletar dados em tempo real para proporcionar uma visão atualizada das condições ambientais.
- ii. Frequência de coleta ajustável para atender às necessidades específicas de monitoramento.

### **3. Eficiência Energética:**

- i. O hardware deve ser projetado com eficiência energética, minimizando o próprio consumo de energia para não afetar as medições.

### **4. Conectividade:**

- i. Integração de módulos de comunicação para facilitar a transmissão dos dados coletados.
- ii. Compatibilidade com diferentes meios de transmissão, como Wi-Fi ou Bluetooth, para garantir flexibilidade na conectividade.

### **5. Facilidade de Instalação:**

- i. Projeto que permita uma instalação fácil e rápida em diferentes ambientes educacionais, sem a necessidade de conhecimentos técnicos avançados.

### **6. Durabilidade e Manutenção:**

- i. Garantir a durabilidade do hardware, considerando o ambiente escolar e as possíveis condições adversas.
- ii. Possibilidade de atualizações de firmware para melhorias contínuas e correções de bugs.

### **7. Compatibilidade com Arduino:**

- i. Utilização de placas Arduino para aproveitar a flexibilidade, facilidade de programação e suporte da comunidade.
- ii. Escolha de sensores e módulos compatíveis com a plataforma Arduino.

### **8. Alimentação:**

- i. Opções de alimentação versáteis, como fontes de energia externas ou baterias recarregáveis, para garantir a operação contínua em diferentes cenários.

### **9. Armazenamento de Dados Local:**

- i. Capacidade de armazenar dados localmente para assegurar a disponibilidade de informações em caso de falhas na transmissão ou para posterior análise.

### **10. Segurança:**

- i. Implementação de medidas de segurança para proteger os dados coletados e garantir a integridade do sistema.

### **11. Facilidade de Integração com API:**

- i. Garantir que o hardware seja compatível e se integre de forma eficiente com a API em Python para a transmissão dos dados.

### **12. Custo e Acessibilidade:**

- i. Manter um equilíbrio entre a funcionalidade do hardware e seu custo, garantindo acessibilidade para instituições educacionais com diferentes orçamentos.

Ao atender a esses requisitos, o hardware baseado em Arduino poderá desempenhar um papel fundamental no sucesso do projeto, fornecendo uma solução robusta, eficiente e acessível para a eficiência energética em ambientes educacionais.

### **Análise de Requisitos para a API Python:**

#### **1. Interface de Comunicação:**

- i. Desenvolver uma interface de comunicação eficiente para permitir a transferência de dados entre o hardware Arduino e o sistema de gerenciamento.

#### **2. Formato de Dados:**

- i. Definir um formato padronizado para os dados transmitidos pela API, facilitando o processamento e a interpretação dessas informações no lado do servidor.

#### **3. Segurança da Comunicação:**

- i. Implementar protocolos de segurança robustos para proteger a transmissão de dados, garantindo confidencialidade e integridade.

#### **4. Documentação Clara:**

- i. Elaborar uma documentação clara e abrangente que descreva os endpoints da API, os métodos disponíveis, os parâmetros necessários e os formatos de resposta.

#### **5. Suporte a Métodos HTTP:**

- i. Implementar suporte aos métodos HTTP adequados para as operações, como GET, POST e PUT, conforme necessário para a interação eficiente com o hardware e o painel web.

#### **6. Autenticação:**

- i. Incorporar um sistema de autenticação seguro para garantir que apenas usuários autorizados possam acessar e modificar os dados.

#### **7. Flexibilidade de Integração:**

- i. Projetar a API de forma flexível para permitir futuras integrações com outros sistemas e acomodar possíveis atualizações de hardware.

#### **8. Tratamento de Erros:**

- i. Implementar mecanismos adequados para o tratamento de erros, fornecendo mensagens claras e respostas apropriadas em caso de falhas.

#### **9. Eficiência e Desempenho:**

- i. Otimizar a API para garantir uma comunicação eficiente entre o hardware e o sistema de gerenciamento, evitando atrasos significativos nas atualizações dos dados.

#### **10. Logs e Monitoramento:**

- i. Incorporar funcionalidades de logs e monitoramento para facilitar a identificação e resolução de problemas, além de possibilitar uma análise retroativa.

#### **11. Controle de Acesso:**

- i. Implementar um controle de acesso granular, permitindo diferentes níveis de permissões para usuários, como leitura e escrita, conforme necessário.

#### **12. Testes Automatizados:**

- i. Desenvolver testes automatizados para garantir a estabilidade e a confiabilidade da API, especialmente ao lidar com atualizações e modificações.

#### **13. Suporte a WebSockets (Opcional):**

- i. Caso necessário, considerar a implementação de suporte a WebSockets para comunicação em tempo real, proporcionando uma experiência mais dinâmica no painel web.



Ao atender a esses requisitos, a API Python proporcionará uma base sólida para a comunicação eficiente entre o hardware Arduino e o sistema de gerenciamento, garantindo a integridade e a segurança dos dados coletados.

## **Análise de Requisitos para o Dashboard**

### **1. Interface Intuitiva:**

- i. Desenvolver uma interface de usuário intuitiva e amigável para garantir que o dashboard seja acessível a usuários com diferentes níveis de habilidade técnica.

### **2. Visualização de Dados:**

- i. Apresentar os dados de consumo de energia, temperatura e luminosidade de forma clara e compreensível, por meio de gráficos, tabelas e outros elementos visuais.

### **3. Calculadora e Log Informativo:**

- i. Permitir a personalização de projeções de consumo através de uma calculadora de gastos e um log de atividades.

### **4. Personalização:**

- i. Permitir a personalização da exibição do dashboard, possibilitando que os usuários escolham as informações que desejam visualizar e configurem as preferências de exibição.

### **5. Acesso Móvel:**

- i. Garantir a adaptabilidade do dashboard para dispositivos móveis, permitindo o acesso conveniente e eficiente em smartphones e tablets.

### **6. Atualização em Tempo Real:**

- i. Implementar a atualização em tempo real dos dados, oferecendo aos usuários uma visão instantânea das condições ambientais nas salas de aula.

### **7. Alertas e Notificações:**

- i. Incorporar um sistema de alertas e notificações para informar os usuários sobre anomalias nos dados ou eventos críticos que exigem atenção imediata.

### **8. Histórico de Dados:**

- i. Oferecer a capacidade de acessar e visualizar o histórico de dados para análises retrospectivas e identificação de tendências ao longo do tempo.

### **9. Compatibilidade com Diferentes Navegadores:**

- i. Assegurar que o dashboard seja compatível com os principais navegadores da web para garantir uma experiência consistente para os usuários.

### **10. Segurança de Acesso:**

- i. Implementar um sistema de autenticação seguro para garantir que apenas usuários autorizados possam acessar o dashboard.

### **11. Integração com API:**

- i. Integrar o dashboard de forma eficiente com a API Python para garantir a atualização contínua e precisa dos dados provenientes do hardware.

### **12. Relatórios e Exportação de Dados:**

- i. Incluir a capacidade de gerar relatórios personalizados e exportar dados para facilitar a análise externa e a geração de relatórios específicos.

### **13. Suporte a Múltiplos Idiomas (Opcional):**

- i. Caso aplicável, considerar a inclusão de suporte a múltiplos idiomas para atender a uma audiência internacional ou diversificada.

#### **14. Compatibilidade com Telas Grandes (Opcional):**

- i. Se necessário, projetar o dashboard para ser compatível com telas grandes ou monitores de sala de controle para facilitar o monitoramento em tempo real.

#### **15. Feedback do Usuário:**

- i. Incorporar mecanismos de feedback do usuário para identificar áreas de melhoria e ajustar continuamente o design e a funcionalidade do dashboard.

Ao atender a esses requisitos, o dashboard proporcionará uma experiência eficiente e eficaz para os usuários, permitindo uma gestão informada e proativa das condições ambientais nas salas de aula.

### **3.3 Limites do Sistema**

O sistema tem limites bem definidos:

- **Escopo Educacional:** O sistema concentra-se na eficiência energética em salas de aula e, portanto, não inclui funcionalidades para outros setores não educacionais.
- **Configuração e Manutenção do Hardware:** A instalação e manutenção do hardware estão fora do escopo do sistema, sendo responsabilidade da instituição educacional.
- **Decisões Administrativas:** O sistema fornece dados, mas as decisões administrativas específicas relacionadas a recursos energéticos e práticas sustentáveis permanecem nas mãos da administração escolar.



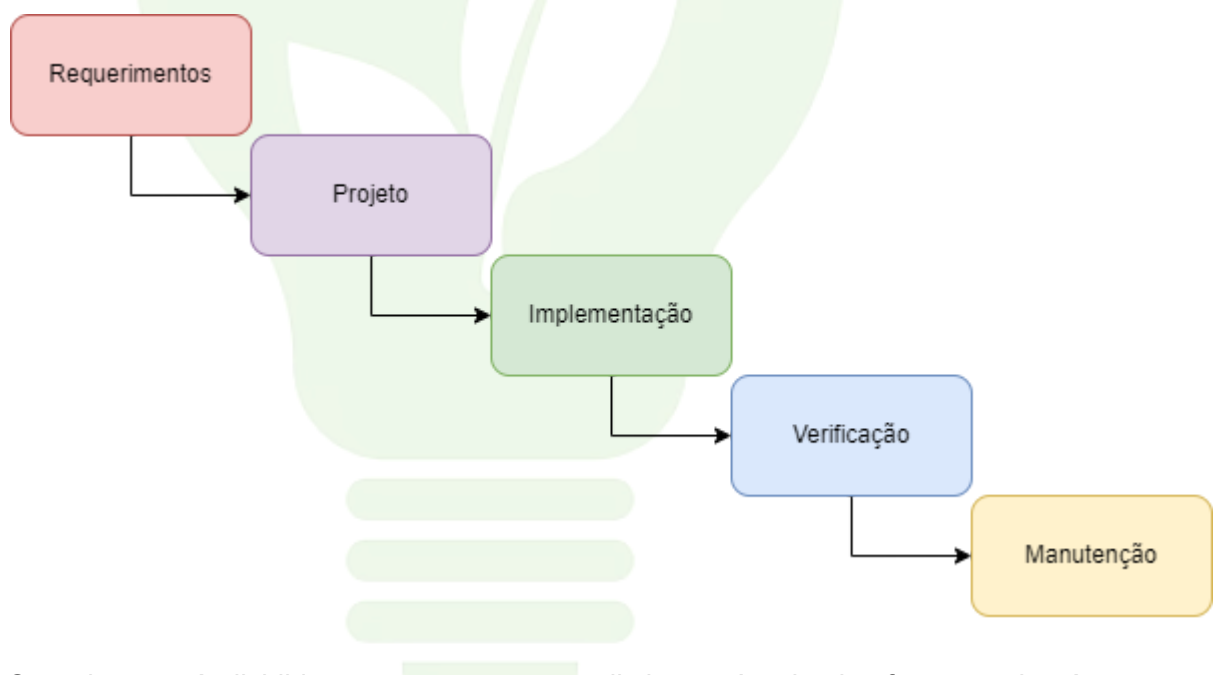
## 4. Escolha da Técnica de Engenharia de Software

### 4.1 Técnica Selecionada

Para a especificação de requisitos deste projeto, será adotada uma abordagem que combina Descrições Textuais, Diagramas de Casos de Uso e User Stories. Essa combinação proporcionará uma compreensão abrangente dos requisitos funcionais e não funcionais, permitindo uma comunicação clara entre a equipe de desenvolvimento, os stakeholders e os usuários finais.

O modelo de projeto a ser utilizado será o waterfall (cascata), a escolha em particular deste modelo se dá a baixa complexidade do projeto, a equipe decidiu por um projeto de baixa complexidade porém que demonstrasse maestria em diversos segmentos no desenvolvimento de software.

Waterfall Model:

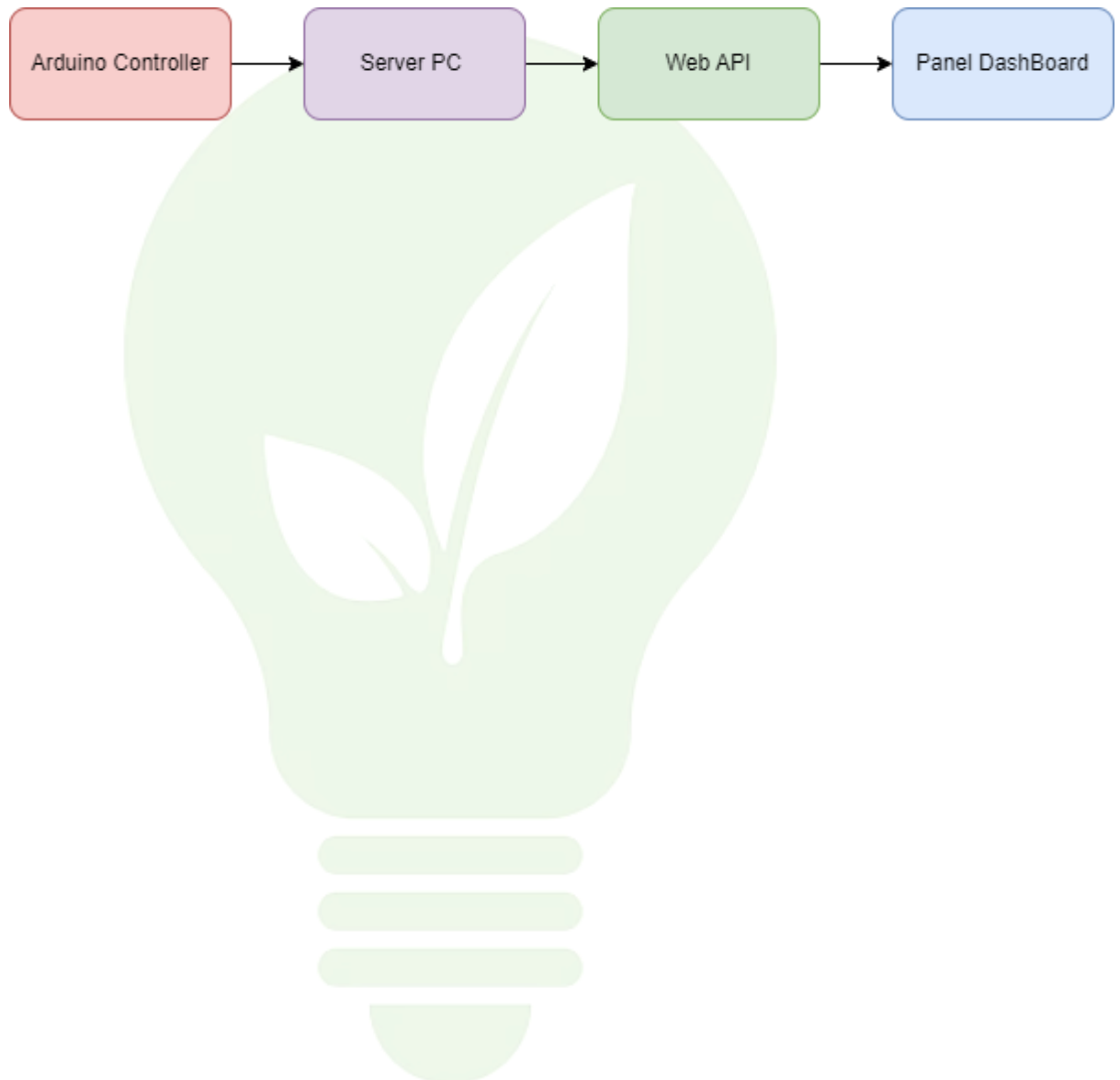


O projeto será dividido em quatro etapas distintas. A primeira fase envolverá a emulação de um Arduino e o desenvolvimento do projeto em hardware, garantindo a replicação das funcionalidades essenciais. Em seguida, será realizada a configuração do ambiente de desenvolvimento e produção, incluindo a implementação dos serviços necessários no servidor.

Na terceira etapa, concentraremos nossos esforços no desenvolvimento da API. Esta API desempenhará um papel crucial ao estabelecer uma conexão direta entre o servidor, o Arduino e o Painel de Controle (Dashboard), permitindo também a integração com outras aplicações.

A última fase do projeto será dedicada ao desenvolvimento do Painel de Controle. Este componente será responsável por facilitar a interação do usuário com os sistemas de monitoramento. Ele permitirá que o usuário final controle o consumo individual de cada sala, além de gerar gráficos e relatórios detalhados. Essa abordagem modular e sequencial visa garantir uma implementação eficiente e bem-sucedida de cada componente do sistema, culminando em uma solução integrada e de fácil utilização para os usuários finais.

Project Model:



## 5. Requisitos Detalhados do Sistema

### 5.1 Requisitos Funcionais

#### Coleta de Dados em Tempo Real

O sistema deve permitir a coleta contínua e em tempo real de dados sobre consumo de energia, temperatura e luminosidade nas salas de aula. Essa funcionalidade deve garantir a precisão das medições e ser configurável para diferentes taxas de amostragem.

### **Transmissão Segura de Dados**

A API Python deve assegurar uma transmissão segura e eficiente dos dados coletados do hardware para o Dashboard.

Utilizar protocolos de segurança padrão para proteger a integridade dos dados durante a transmissão.

### **Interface Intuitiva do Dashboard**

O Dashboard deve fornecer uma interface intuitiva, apresentando gráficos e tabelas claras e compreensíveis.

Permitir a personalização da exibição para atender às preferências dos usuários.

### **Personalização da Exibição**

Os usuários devem poder personalizar a exibição do Dashboard, destacando informações específicas ou ajustando a visualização conforme suas preferências.

### **Sistema de Alertas Automáticos**

Implementar um sistema de alertas que notifica os usuários sobre eventos críticos ou anomalias detectadas nos dados.

Esses alertas devem ser configuráveis, permitindo aos usuários definir seus próprios parâmetros.

## **5.2 Restrições de Design**

As restrições de design são fundamentais para orientar o desenvolvimento do sistema, considerando as limitações impostas pelo cliente, pelo ambiente de operação e pelas escolhas tecnológicas prévias.

### **Limitações Tecnológicas:**

As escolhas tecnológicas, incluindo a utilização do Arduino, impõem restrições de compatibilidade e desempenho. O sistema deve ser desenvolvido levando em consideração essas limitações para garantir a integração eficaz e a operação eficiente.

### **Restrições Regulatórias:**

O sistema deve atender às regulamentações locais e nacionais relacionadas à segurança de dados e privacidade. Restrições impostas por normas específicas devem ser incorporadas ao design para garantir conformidade.

### **Restrições Orçamentárias:**

Limitações orçamentárias definidas pelo cliente devem ser consideradas durante o processo de design, influenciando escolhas de hardware, tecnologias e abordagens de implementação.

**Ambiente de Operação:**

Restrições específicas do ambiente, como condições de temperatura e umidade em salas de aula, devem ser levadas em conta para garantir a durabilidade e confiabilidade do sistema.

**5.3 Atributos do Sistema**

Os atributos do sistema desejados são características fundamentais que contribuem para a eficácia e a utilidade do sistema, indo além das funcionalidades específicas.

**Confiabilidade:**

O sistema deve ser altamente confiável, assegurando a coleta precisa de dados e a transmissão segura, mesmo em condições adversas. A confiabilidade é crucial para manter a integridade das informações monitoradas.

**Manutenibilidade:**

A manutenibilidade do sistema deve ser uma prioridade, garantindo que o código seja modular, bem documentado e de fácil compreensão. Isso facilitará atualizações, correções e melhorias futuras.

**Portabilidade:**

O sistema deve ser projetado para ser facilmente adaptado a diferentes ambientes educacionais, promovendo a portabilidade. Isso facilitará a implementação em diversas instituições de ensino, independentemente de suas características específicas.

**Segurança:**

A segurança dos dados é essencial. O sistema deve implementar medidas robustas para proteger informações sensíveis, atendendo a padrões e regulamentações de segurança relevantes.

**Desempenho Eficiente:**

O sistema deve oferecer um desempenho eficiente, garantindo respostas rápidas a solicitações do usuário e minimizando o impacto no consumo de recursos, como energia e largura de banda.

**Facilidade de Uso:**

A interface do usuário, especialmente no Painel de Controle, deve ser intuitiva e de fácil utilização, garantindo que usuários de diferentes níveis de habilidade possam interagir eficazmente com o sistema.

## 6. Informações de Suporte

**6.1 Anexos (caso tenha)**

Inclusão de qualquer documento de suporte ou informação adicional que possa auxiliar na compreensão dos requisitos, tais como mock-ups, diagramas ou tabelas, links de wikis, documentações.

