# Histórico de Versões

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** | **Revisor** | **Aprovado por** |
| 13/01/2025 | 1.1 | Elaboração | Antônia Vitória | Antônia Vitória | Antônia Vitória |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Responsáveis

## Equipe

**Coordenador**

Rennan Maia

**Desenvolvimento**

Antônia Andrade

## Cliente

**Gestor do sistema**

A definir.

# Documento de Visão de Projeto

## Objetivo

O propósito deste documento é coletar, analisar e definir as necessidades de alto-nível e características do projeto de software, focando nas potencialidades requeridas pelos afetados e usuários-alvo, e como estes requisitos serão abordados no projeto de software.

A visão do projeto documenta o ambiente geral de processos a ser desenvolvido para o sistema durante o projeto, fornecendo a todos os envolvidos uma descrição compreensível deste e de suas macros funcionalidades.

O Documento de Visão de Projeto documenta apenas as necessidades e funcionalidades do sistema que serão atendidas no projeto de software.

## Descrição do Projeto

O projeto visa desenvolver um sistema para monitorar e exibir dados de consumo de energia elétrica em um ambiente residencial. Os dados serão coletados usando sensores conectados a um microcontrolador e armazenados em um banco de dados local. A interface web mostrará o consumo em tempo real e relatórios básicos.

## Envolvimento

## Abrangência

O SiGEE (Sistema de Gerenciamento de Energia Elétrica) será um sistema web hospedado em um servidor local. Inicialmente, será acessado através de um navegador web no endereço fornecido pelo servidor local.

O sistema fará uso de uma API que será responsável por permitir que o sistema se comunique de forma descentralizada com diferentes componentes, como: Arduino/ESP32 (envia dados de consumo de energia para a API), Banco de Dados MySQL (armazena os dados recebidos pela API) e Interface Web (Dashboard), que consome os dados da API para exibir gráficos e relatórios.

O sistema terá dois tipos de usuários: o usuário padrão, capaz de visualizar o consumo de energia em tempo real através do dashboard web e acessar relatórios de consumo armazenados. E o administrador, que irá gerenciar dispositivos conectados ao sistema, realizar a configuração inicial do sistema e monitorar o armazenamento e desempenho do sistema.

## Papel dos atores

|  |  |
| --- | --- |
| **Descrição** | **Padrão** |
| **Papel** | Tomar decisões baseadas nas informações apresentadas no dashboard. |
| **Insumos ao sistema** | Consultar o consumo de energia em tempo real e acessar relatórios históricos. |
| **Representante** | A definir. |

|  |  |
| --- | --- |
| **Descrição** | **Responsável** |
| **Papel** | Garantir que o sistema esteja funcionando corretamente e aplicar manutenções quando necessário. |
| **Insumos ao sistema** | Gerenciar os dispositivos conectados ao sistema, configurar a API e monitorar o sistema. |
| **Representante** | A definir. |

## Necessidades e Funcionalidades

|  |  |  |  |
| --- | --- | --- | --- |
| **Necessidade** | **ID Func.** | **Funcionalidade Correspondente** | **Prioridade** |
| Monitorar o consumo de energia em tempo real | **F1** | Coletar dados de consumo usando o Arduino e o sensor de corrente | Alta |
| Armazenar os dados coletados | **F2** | Salvar as informações no banco de dados MySQL | Alta |
| Exibir os dados de consumo em uma interface web | **F3** | Criar um dashboard simples com HTML, CSS e JavaScript | Alta |
| Simular dados para testes iniciais | **F4** | Gerar valores de consumo fictícios | Média |
| Integrar hardware com o banco de dados | **F5** | Utilizar uma API para envio e consulta de dados | Alta |
| Permitir o gerenciamento de dispositivos | **F6** | Criar um módulo de cadastro e consulta de dispositivos no sistema | Média |
| Oferecer relatórios históricos de consumo | **F7** | Implementar gráficos e tabelas de consumo ao longo do tempo | Média |

## ****Tecnologias Utilizadas****

## Modelo de banco de dados

### ****Tabela: consumo****

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** |
| id | INT | Identificador único (PK). |
| Dispositivo | VARCHAR | Identificação do dispositivo. |
| Consumo | FLOAT | Dados de consumo em kWh. |
| Data\_hora | TIMESTAMP | Data e hora da coleta. |

## ****Tabela: dispositivo****

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** |
| id | INT | Identificador único (PK). |
| Nome | VARCHAR | Nome do dispositivo. |
| Localizacao | VARCHAR | Local onde o dispositivo está instalado. |

## Restrições tecnológicas

A seguir, encontra-se as principais restrições tecnológicas do Sistema de Gerenciamento de Energia Elétrica (SiGEE):

|  |  |  |  |
| --- | --- | --- | --- |
| **Categoria** | **Restrição Tecnológica** | **Impacto** | **Solução Possível** |
| **Hardware** | Limitação de memória e capacidade de processamento do Arduino | Pode afetar o desempenho e a quantidade de dados coletados | Usar um microcontrolador mais potente, como o ESP32 |
| **Conexão** | Dependência de uma conexão Wi-Fi estável para enviar dados à API | Falhas de conexão podem impedir a coleta de dados em tempo real | Implementar um buffer local para armazenar dados offline |
| **Banco de Dados** | Limitação no volume de dados armazenados no MySQL (especialmente em servidores locais como XAMPP) | Pode afetar a escalabilidade do sistema | Migrar para um servidor de banco de dados na nuvem |
| **API** | Uso de APIs gratuitas pode ter limites de requisições ou tempo de resposta lento | Pode afetar a integração em tempo real | Considerar APIs premium ou criar uma API própria |
| **Interface Web** | Limitação de funcionalidades em uma interface simples com HTML, CSS e JavaScript puro | Pode dificultar a criação de dashboards interativos avançados | Usar frameworks como React ou Vue.js futuramente |
| **Sensores** | Precisão do sensor de corrente pode variar dependendo das condições de instalação | Dados de consumo podem não ser 100% precisos | Calibrar os sensores corretamente e realizar ajustes periodicamente |
| **Segurança** | Falta de criptografia nas comunicações entre o Arduino, API e banco de dados | Pode deixar os dados vulneráveis a ataques | Implementar HTTPS e autenticação nas requisições |
| **Escalabilidade** | Arquitetura inicial projetada para poucos dispositivos | Limita a expansão do sistema | Modelar o sistema para permitir múltiplos dispositivos |
| **Compatibilidade** | Dependência de componentes específicos (Arduino, sensor SCT-013) | Pode dificultar substituições ou atualizações de hardware | Usar componentes compatíveis com outras plataformas |

## Critérios de aceitação do sistema

Aqui estão os **critérios de aceitação** do sistema SiGEE:

* 1. **Critérios de Aceitação Funcional**

|  |  |  |
| --- | --- | --- |
| **ID** | **Critério de Aceitação** | **Descrição** |
| **C1** | O sistema deve ser capaz de coletar dados simulados e reais de consumo de energia | O sistema deve iniciar com dados simulados e depois coletar dados reais via sensor |
| **C2** | A interface web deve exibir os dados em um dashboard simples | Os dados devem ser apresentados em forma de gráfico ou tabela atualizada em tempo real |
| **C3** | Os dados devem ser armazenados corretamente no banco de dados | Todos os dados enviados pelo Arduino ou simulados devem ser persistidos no MySQL |
| **C4** | A integração com a API deve funcionar de forma eficiente | O sistema deve ser capaz de enviar e receber dados através de uma API sem erros |
| **C5** | O sistema deve permitir acesso via navegador | A interface web deve ser acessível através de um endereço local |

## ****Critérios de Aceitação Técnicos****

|  |  |  |
| --- | --- | --- |
| **ID** | **Critério de Aceitação** | **Descrição** |
| **C6** | O hardware (Arduino e sensor) deve ser funcional | O circuito deve estar montado corretamente e funcionando para coleta de dados |
| **C7** | O banco de dados deve suportar consultas rápidas | As consultas feitas pela interface web não devem exceder 2 segundos de tempo de resposta |
| **C8** | O sistema deve ser modular e expansível | Deve ser possível adicionar novos sensores e dispositivos ao sistema futuramente |
| **C9** | O sistema deve ser seguro | O sistema deve garantir que os dados trafeguem de forma segura utilizando HTTPS ou outro método de proteção |
| **C10** | O sistema deve ser documentado | Todo o projeto deve ter documentação explicando seu funcionamento e configuração |

## ****Critérios de Aceitação de Interface****

|  |  |  |
| --- | --- | --- |
| **ID** | **Critério de Aceitação** | **Descrição** |
| **C11** | A interface deve ser responsiva | O dashboard deve ser exibido corretamente em diferentes tamanhos de tela |
| **C12** | O design da interface deve ser simples e intuitivo | O usuário deve ser capaz de interpretar os dados de consumo sem dificuldades |

## Escopo não incluído no sistema

Aqui estão os principais itens fora do escopo do sistema para ajudar a evitar complexidade desnecessária e permitir que o projeto mantenha seu foco no objetivo principal, garantindo que seja viável de ser desenvolvido dentro do tempo e dos recursos disponíveis.

|  |  |
| --- | --- |
| **Item Fora do Escopo** | **Justificativa** |
| **Controle remoto de dispositivos** | O sistema não terá, neste estágio inicial, a funcionalidade de ligar/desligar dispositivos remotamente. |
| **Monitoramento individual de dispositivos** | O sistema coletará dados de consumo de um único ponto geral (sensor de corrente), não monitorando cada dispositivo separadamente. |
| **Notificações e alertas automáticos** | Não haverá envio de alertas por e-mail, SMS ou notificações push em caso de consumo anormal. |
| **Cálculo de custo financeiro de energia** | O sistema não fará cálculo direto do custo financeiro do consumo de energia baseado na tarifa elétrica. |
| **Integração com sistemas de automação** | Não haverá integração com sistemas de automação residencial ou industrial (ex.: controle de luzes, termostatos). |
| **Segurança avançada (criptografia, autenticação)** | O sistema não incluirá mecanismos avançados de segurança, como criptografia de dados ou autenticação de usuários. |
| **Relatórios avançados e preditivos** | O sistema, em sua versão básica, não terá funcionalidades de análise preditiva ou relatórios avançados. |
| **Aplicativo mobile** | Não será desenvolvido um aplicativo para dispositivos móveis nesta versão inicial do projeto. |
| **Monitoramento em múltiplos ambientes** | O sistema será projetado para monitorar o consumo em um único local (residência ou laboratório), sem suporte para múltiplos ambientes. |
| **Interface multiusuário com diferentes níveis de acesso** | O sistema será acessível por um único tipo de usuário, sem controle de permissões ou diferentes níveis de acesso. |

## Cronograma

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Semana** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Definição do Escopo** |  |  |  |  |  |  |  |  |  |  |
| **Design da Interface Web** |  |  |  |  |  |  |  |  |  |  |
| **Banco de Dados** |  |  |  |  |  |  |  |  |  |  |
| **Programação do Arduino** |  |  |  |  |  |  |  |  |  |  |
| **Integração** |  |  |  |  |  |  |  |  |  |  |
| **Testes e Ajustes** |  |  |  |  |  |  |  |  |  |  |
| **Documentação e Apresentação** |  |  |  |  |  |  |  |  |  |  |