



FUNDAÇÃO
HERMÍNIO OMETTO

Sistemas de Informação
Projeto Interdisciplinar II / 4ºA/2025

**Tecnologias Inovadoras para promover Energia Limpa,
Sustentável e Acessível**

**Plataforma de Monitoramento e Otimização de Consumo de
Energia**
NeoSolaris

André Henrique Valério - 102610

Geovana Miuky Ogawa - 117509

João Rafael L. de Jesus - 117322

Nádia Nayara Dias Coelho - 117260

Quéren-Hapuque de D. Alves - 116528

Victor Henrique Bruno - 114587

3^a Fase do Projeto Técnico

1 Proposta do Projeto

1.1 Contexto

A eficiência energética tem sido um dos principais desafios enfrentados pela sociedade atual, onde o consumo excessivo de eletricidade impacta tanto os custos operacionais quanto o meio ambiente. Segundo a Eletrobrás (2002), para gerenciar a energia de uma instalação de forma eficaz, é essencial compreender os sistemas energéticos disponíveis, os hábitos de consumo, os processos de aquisição de energia e a experiência dos usuários e técnicos envolvidos. Nesse sentido, a automação de dispositivos elétricos surge como uma solução eficiente para otimizar o consumo energético.

A falta de controle sobre os gastos, somada ao uso ineficiente de equipamentos e instalações elétricas, tem gerado uma crescente preocupação quanto ao desperdício de energia. Além disso, os custos com eletricidade aumentam com a aplicação das bandeiras tarifárias, que impactam diretamente as faturas de energia, tornando-as ainda mais imprevisíveis e desvantajosas (HERTZ ELÉTRICA, 2025).

O encarecimento das tarifas é um problema particularmente agravado pela escassez hídrica. As usinas hidrelétricas, que representam a maior parte da matriz energética brasileira (64,9%), estão cada vez mais vulneráveis, devido à diminuição dos níveis de reservatórios, especialmente em períodos de seca severa. Como consequência, são aplicadas taxas extras, como a bandeira de escassez hídrica, que aumentam ainda mais o custo para os consumidores (HERTZ ELÉTRICA, 2025).

Esses números revelam uma crescente pressão sobre o sistema elétrico nacional, ao mesmo tempo em que destacam a necessidade urgente de soluções mais eficientes e sustentáveis para gerenciar o consumo de energia.

Nesse contexto, o projeto surge como uma proposta inovadora, focada em soluções tecnológicas para monitoramento, otimização e controle inteligente do consumo de energia elétrica residencial. A implementação desse sistema não apenas reduz os custos com energia, mas também contribui para práticas sustentáveis, alinhadas aos Objetivos de

Desenvolvimento Sustentável (ODS) da ONU. Estudos abordam como essas tecnologias podem otimizar o consumo de energia, reduzir desperdícios e melhorar a eficiência energética geral, como também, destacam a importância da automação e do monitoramento em tempo real para tomadas de decisão mais eficazes (Al-Ali et al, 2017). Assim, integrar soluções tecnológicas a ambientes representa um avanço significativo para a eficiência energética e a sustentabilidade.

1.2 Escopo

O sistema gerencia e monitora ambientes e dispositivos, com foco em eficiência energética. Acessando a página inicial da plataforma, é possível visualizar páginas com o intuito de fornecer informações do que será possível consumir após o cadastro de uma conta no sistema. Após autenticação, o usuário poderá realizar o cadastro, edição e exclusão (CRUD) de ambientes e dispositivos, além de configurar metas de consumo. O sistema permitirá a consulta de todos os ambientes e dispositivos cadastrados, exibindo informações detalhadas sobre cada um. Terá histórico e relatórios, baseados nos dados de consumo energético de todos ambientes e dispositivos, com a possibilidade de filtragem por períodos e comparação entre as entidades cadastradas.

1.3 Requisitos

A especificação dos requisitos visa descrever as funcionalidades essenciais que o sistema deve oferecer aos usuários. Esses requisitos foram definidos com base nos processos identificados durante a modelagem do negócio e estão diretamente alinhados com os objetivos do projeto, como promover economia de energia, facilitar o gerenciamento de ambientes e dispositivos, e permitir o controle eficiente por meio do monitoramento e relatórios de consumo. Abaixo estão listados os requisitos:

- Permitir que o usuário se cadastre para acesso à plataforma.
- Autenticação de usuário.
- Cadastrar ambientes que representem espaços físicos onde os dispositivos estarão inseridos.
- Cadastrar dispositivos que serão monitorados e controlados pelo sistema.
- Exibir o consumo de energia dos dispositivos cadastrados.

- Histórico de consumo total e médio de energia nos ambientes/dispositivos.
- Gerar relatórios/dashboard sobre o consumo diário/mensal/anual e desempenho energético atingido.
- Coletar, armazenar e analisar dados de consumo.
- Permitir que o usuário defina metas de consumo e receba alertas ao se aproximar do limite definido.
- Gráficos interativos para visualização do consumo por período, dispositivo ou ambiente.
- Interface responsiva.

Esses requisitos formam a base funcional da solução proposta, permitindo uma gestão energética inteligente, personalizada e alinhada às demandas atuais dos consumidores.

2 Modelagem BPMN e Diagrama de Caso de Uso

O diagrama BPMN (*Business Process Model and Notation*) desenvolvido representa de forma detalhada o fluxo de funcionamento do sistema desde o acesso inicial do usuário até o monitoramento e análise dos dispositivos cadastrados. Seu objetivo é demonstrar as principais etapas, decisões e interações que ocorrem dentro da aplicação.

O diagrama de caso de uso representa as principais interações entre o usuário e o sistema. Seu objetivo é demonstrar, de forma simples e visual, as funcionalidades disponíveis e como o usuário pode acessá-las dentro da aplicação.

O acesso para o diretório que contém as imagens com o modelo BPMN e diagrama de caso de uso, está na plataforma do GitHub, na URL: [<https://github.com/energyflow-plataform/energyflow/tree/main/modeling>].

2.1 Modelagem BPMN

O processo inicia com o acesso ao site e a realização do login do usuário. A partir disso, o sistema direciona para diferentes painéis de acordo com o nível de acesso: administrador ou usuário comum. O administrador possui privilégios adicionais, como adicionar e configurar dispositivos, gerenciar usuários, editar informações e visualizar estatísticas globais. Já o usuário comum pode visualizar e acompanhar os dispositivos vinculados à sua conta.

Após o login, ocorre o processo de cadastro e configuração de dispositivos, que inclui a definição de um nome e ID, além da verificação de conexão e autenticação. Caso o dispositivo esteja corretamente vinculado, o sistema redireciona o usuário para a tela de dispositivos cadastrados; caso contrário, é exibida uma mensagem de erro e o processo é interrompido.

Na etapa de monitoramento, o sistema coleta e processa dados em tempo real, possibilitando a visualização de métricas de consumo energético diárias, semanais, mensais e anuais. A cada período, são gerados relatórios contendo informações como a potência (kWh).

Além disso, o sistema possui uma funcionalidade de alertas automáticos, que identifica valores fora do padrão de consumo. Quando detectada uma anomalia, é exibido um alerta de alto consumo, direcionando o usuário para a tela de energia, onde é possível analisar os dados detalhadamente.

Por fim, todos os fluxos que não apresentam anomalias retornam para a tela inicial, encerrando o ciclo de monitoramento. Dessa forma, o BPMN permite compreender de maneira estruturada as etapas de autenticação, configuração e acompanhamento energético dentro do sistema, evidenciando a integração entre as funcionalidades administrativas e de usuário.

2.2 Diagrama de Caso de Uso

No diagrama, o ator principal é o Usuário, responsável por interagir diretamente com o sistema e realizar as operações disponíveis. As funcionalidades estão organizadas em diferentes casos de uso, que descrevem as ações que podem ser executadas.

O primeiro caso de uso é o Cadastrar, que permite ao usuário criar uma conta no sistema, fornecendo suas informações básicas para autenticação e acesso futuro. Em seguida, o Login possibilita a entrada do usuário na plataforma por meio de suas credenciais, garantindo segurança e individualização das sessões.

Após o acesso, o usuário pode criar ambientes e dispositivos, funcionalidade essencial para a organização e o gerenciamento de diferentes locais de monitoramento. Na sequência, o caso de uso Configurar Dispositivo permite ajustar parâmetros e vincular os dispositivos à conta do usuário, preparando-os para o monitoramento.

O caso de uso Controlar Dispositivos abrange as ações de visualização, ativação e desativação dos equipamentos, oferecendo controle direto sobre os dispositivos cadastrados. Já o Alterar/Excluir Dados concede ao usuário a capacidade de atualizar

informações ou remover registros desnecessários, mantendo o sistema consistente e atualizado.

Por fim, o Gerar Relatórios possibilita a visualização de dados consolidados sobre o desempenho e o consumo energético dos dispositivos, permitindo uma análise mais detalhada e estratégica do uso dos recursos.

Em resumo, o diagrama de caso de uso demonstra de forma clara as principais funcionalidades do sistema e o papel ativo do usuário na operação e gestão dos dispositivos, evidenciando a interação entre o ser humano e a aplicação tecnológica desenvolvida.

3 Relação de Artefatos de Banco de Dados II

Os artefatos do Banco de Dados do sistema têm como objetivo representar de forma clara e organizada os dados que sustentam o funcionamento da plataforma. Esse processo é essencial para garantir a integridade das informações e facilitar o desenvolvimento das funcionalidades, que vão desde o cadastro de usuários até o controle detalhado dos dispositivos e a análise do consumo energético.

O banco de dados foi projetado para armazenar informações sobre usuários, unidades (empresas, residências ou instalações), ambientes, dispositivos, registros de consumo e dados de localização. As entidades se relacionam entre si de maneira hierárquica, possibilitando o controle de acesso, a organização espacial e a análise detalhada do consumo energético.

3.1. Tabelas

- **Tabela usuário:**

Armazena os dados dos usuários cadastrados no sistema. Cada usuário possui informações pessoais básicas (nome, sobrenome, e-mail e senha), além do tipo de acesso, que pode ser *administrador* ou *comum*. O campo *dt_cadastro* registra o momento da criação do usuário e *ultimo_login* permite acompanhar a última autenticação.

Principais atributos:

- *id*: identificador único do usuário;
- *nome* e *sobrenome*: identificação pessoal;
- *email*: endereço eletrônico único;

- senha: credencial de acesso ao sistema;
- tipo: define o nível de permissão (admin ou comum);
- dt_cadastro e ultimo_login: controle temporal.

- **Tabela unidade**

Representa a unidade monitorada — que pode ser uma empresa, casa, prédio ou instalação. Cada unidade é vinculada a um usuário administrador por meio do campo criado_por e está associada a um endereço completo, utilizando o campo id_cep como chave estrangeira.

Principais atributos:

- id: identificador único da unidade;
- nome: nome da unidade (ex.: “Fábrica Central” ou “Residência A”);
- id_cep: vínculo com o endereço completo;
- cod_acesso: código único utilizado para permitir o vínculo de novos usuários à unidade;
- criado_por: indica o usuário responsável pela criação da unidade;
- dtCadastro: data de registro.

- **Tabela usuario_unidade**

Essa tabela faz o controle de vínculo entre usuários e unidades, funcionando como uma relação N:N. Ela também armazena o status do vínculo, permitindo que os administradores aprovem, rejeitem ou mantenham solicitações pendentes.

Principais atributos:

- id_usuario: referência ao usuário;
- id_unidade: referência à unidade;
- situacao: indica se o vínculo está “pendente”, “aprovado” ou “rejeitado”;
- dt_vinculo: registra o momento do vínculo.

- **Tabela ambiente**

Define os ambientes internos de uma unidade, como "Sala de Reunião", "Laboratório", "Cozinha" etc. Cada ambiente pertence a uma única unidade e pode conter diversos dispositivos.

Principais atributos:

- id: identificador único do ambiente;
- id_unidade: vínculo com a unidade;
- nome: nome do ambiente;
- descricao: informações adicionais sobre o local.

- **Tabela dispositivo**

Contém os equipamentos monitorados dentro de cada ambiente, como motores, geladeiras, ar-condicionado e outros. Armazena dados como nome, tipo, potência e situação atual (ativo/inativo).

Principais atributos:

- id: identificador do dispositivo;
- id_ambiente: ambiente ao qual o dispositivo pertence;
- nome e tipo: identificação do equipamento;
- potencia: consumo energético nominal em watts;
- situacao: define se o dispositivo está em funcionamento.

- **Tabela consumo**

Registra os dados de consumo energético coletados pelos dispositivos em intervalos de tempo pré-definidos. Armazena o consumo mínimo, máximo e médio, otimizando o uso do banco de dados sem sobrecarregá-lo com leituras contínuas.

Principais atributos:

- id_dispositivo: referência ao dispositivo monitorado;
- inicio_intervalo e fim_intervalo: definem o período de coleta;

- consumo_min, consumo_max e consumo_medio: valores registrados no intervalo.

- **Tabelas de Localização (cep, bairro, cidade, estado)**

Essas tabelas representam a estrutura hierárquica de endereços, promovendo **normalização e reutilização dos dados**. Elas garantem que uma unidade possa ser localizada com precisão sem duplicar informações.

- **estado**: contém o nome e a sigla de cada estado.
- **cidade**: vincula-se a um estado.
- **bairro**: vincula-se a uma cidade.
- **cep**: contém o código postal e logradouro, vinculado a um bairro.

3.2. Relações entre as Tabelas

- **Usuário ↔ Unidade**: relação N:N, representada pela tabela usuario_unidade;
- **Unidade ↔ Ambiente**: relação 1:N, uma unidade possui vários ambientes;
- **Ambiente ↔ Dispositivo**: relação 1:N, um ambiente contém vários dispositivos;
- **Dispositivo ↔ Consumo**: relação 1:N, um dispositivo gera múltiplos registros de consumo;
- **Unidade ↔ Localização**: relação 1:1 via id_cep, permitindo detalhamento do endereço.

A seguir, temos o modelo lógico (Figura 1) do banco de dados:

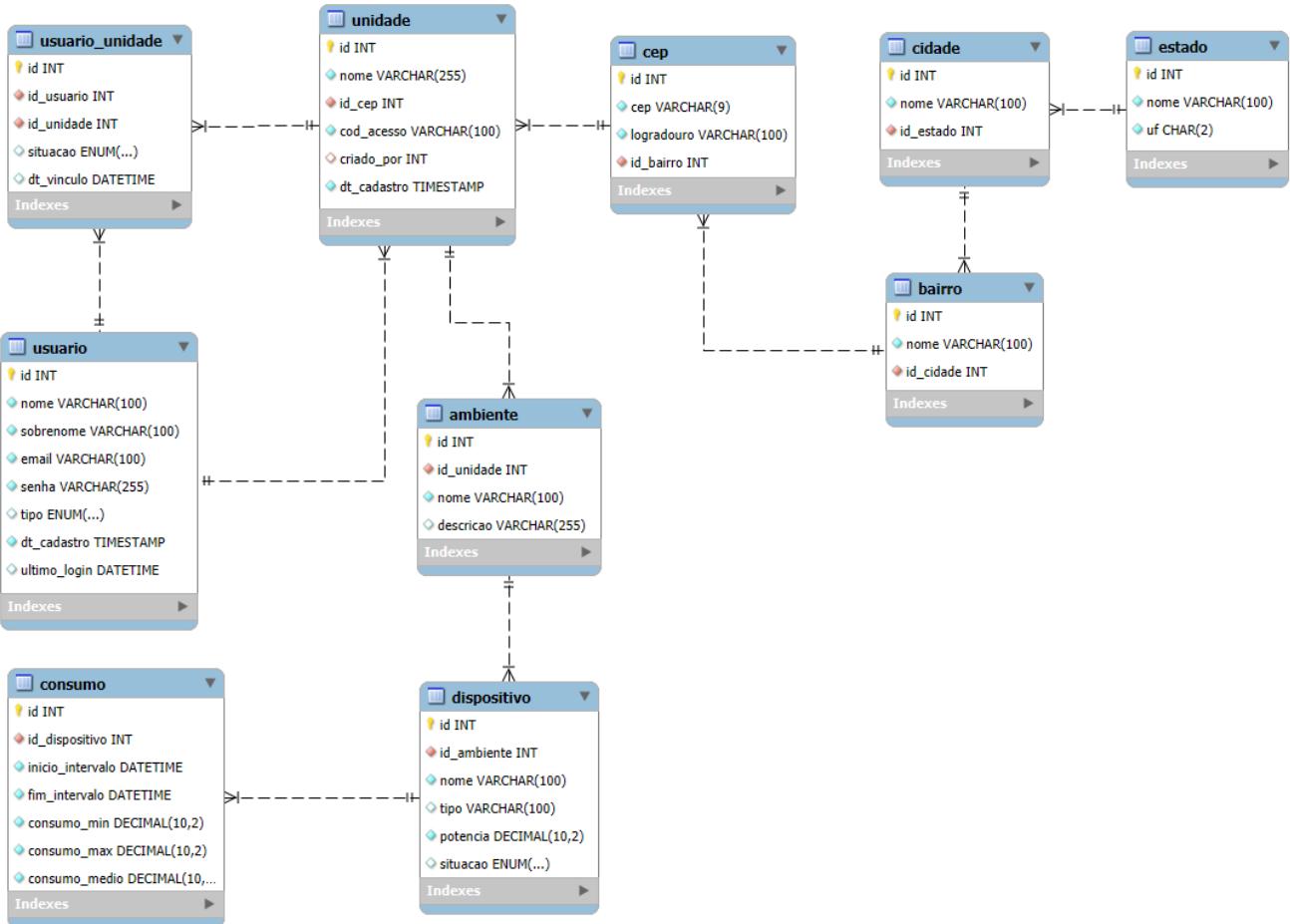


Figura 1 – Modelo Lógico do Banco de Dados.

O acesso para o diretório que contém o arquivo com o script SQL com os *create*, *insert*, *select*, *views* para os relatórios e o modelo lógico, está na plataforma do GitHub, na URL: [<https://github.com/energyflow-plataform/energyflow/tree/main/database>].

4 Relação de Artefatos de Programação para Web I

Nesse contexto, os artefatos voltados à Programação para Web estão focados na página inicial da plataforma, com o objetivo de alinhar a identidade visual da plataforma à proposta central do projeto, voltada ao monitoramento em tempo real.

4.1 Front-end

A página inicial foi desenvolvida para refletir essa característica, apresentando um design dinâmico e interativo. Foram incorporadas animações de gráficos com movimentação contínua, que atualizam periodicamente as informações fictícias de consumo e os valores gastos, reforçando a ideia de acompanhamento constante dos dados e proporcionando uma experiência mais imersiva e informativa ao usuário. A seguir temos a primeira seção da página (Figura 2):

The screenshot shows the homepage of the EnergyFlow project. At the top, there is a navigation bar with the logo 'EnergyFlow' and links for 'Inicio', 'Recursos', 'Benefícios', 'Casos de uso', 'Preços', 'Contato', and a green button 'Acessar Sistema'. The main section features a large heading: 'Monitore sua energia em tempo real, reduza custos e aumente a eficiência'. Below the heading, a subtext states: 'A nossa plataforma fornece dados precisos, alertas automáticos e visualização do consumo de energia em tempo real.' There are two buttons: 'Acessar Sistema' and 'Fale com um especialista'. To the right, a box titled 'Painel de energia' displays a bar chart with five bars of varying heights, representing energy consumption. Below the chart, the text 'Consumo atual' is followed by '2.4 kW' and 'Custo de hoje' followed by 'R\$12.50'. At the bottom left, three performance metrics are listed: '30%' (Redução de custos), '24h' (Monitoramento em tempo real), and '99.9%' (Garantia de disponibilidade).

Figura 2 – Primeira seção da página inicial do Projeto EnergyFlow.

Seguindo para o menu na parte superior da tela, o usuário terá acesso para a página de autenticação (Figura 3) ou cadastro de usuário (Figura 4), mostrado a seguir:

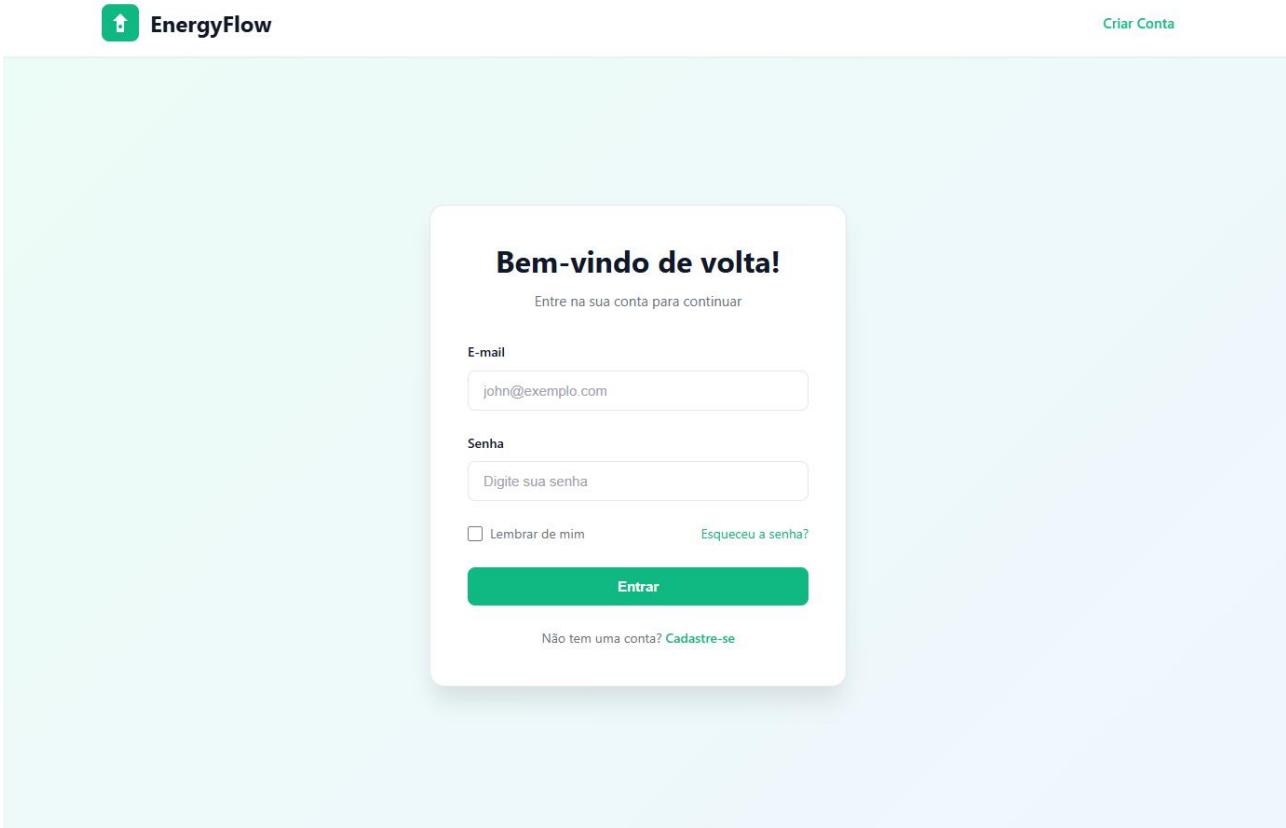


Figura 3 – Página de autenticação de usuário.

The screenshot shows the 'Criar Conta' (Create Account) page of the EnergyFlow website. At the top left is the EnergyFlow logo with a green arrow icon. At the top right is a 'Entrar' (Log In) button. The main title 'Criar Conta' is centered at the top of the form. Below it is a subtitle: 'Comece a monitorar seu consumo energético hoje'. The form fields are arranged in two columns: 'Nome' (Name) and 'Sobrenome' (Last Name), each with a placeholder 'Seu nome' (Your name). Next are 'E-mail' (Email) with placeholder 'john@exemplo.com' and 'Senha' (Password) with placeholder 'Crie uma senha forte' (Create a strong password). A note below the password field states: 'Deve possuir pelo menos 8 caracteres com letras maiúsculas, minúsculas, número e caractere especial' (Must have at least 8 characters with uppercase letters, lowercase letters, numbers and special characters). Below the password field is a 'Confirmar Senha' (Confirm Password) field with placeholder 'Digite a senha novamente' (Enter the password again). A checkbox labeled 'Eu concordo com os Termos de uso e a Política de Privacidade' (I agree with the Terms of Use and Privacy Policy) is present, with a link to the 'Termos de uso' (Terms of Use) and 'Política de Privacidade' (Privacy Policy) in green. At the bottom center is a large green 'Criar Conta' (Create Account) button. Below the button is a link 'Já possui uma conta? Entrar' (Already have an account? Log In).

Figura 4 – Página de cadastro de usuário.

Seguindo para a seção de gerenciamento no menu principal, o usuário poderá acessar as telas responsáveis pelo **cadastro, edição e exclusão** de registros de **Unidades, Ambientes e Dispositivos**, conforme ilustrado nas figuras a seguir (Figuras 5, 6 e 7).

Cadastrar Unidade
Registre uma nova unidade para monitorar

Nome
Ex: Matriz São Paulo

Código de Acesso
Digite o código para acessar a unidade

CEP
Ex: 01001-000

Número
Digite o número do endereço

Cadastrar Unidade

Figura 5 – Página de cadastro de unidade.

Cadastrar Ambiente
Adicione um novo ambiente para monitorar

Nome
Ex: Setor II

Descrição
Descreva o ambiente...

Unidade
Selecione uma unidade...

Cadastrar Ambiente

Figura 6 – Página de cadastro de ambiente.

Figura 7 – Página de cadastro de dispositivo.

Essas páginas compõem a estrutura hierárquica do sistema, onde cada **Ambiente** precisa estar associado a uma **Unidade** previamente cadastrada, e cada **Dispositivo** deve estar vinculado a um **Ambiente** existente. Essa relação garante a integridade dos dados e reflete a organização real dos locais monitorados.

No frontend, durante o processo de cadastro, o sistema apresenta caixas de seleção (*select boxes*) que exibem apenas os registros disponíveis no momento. Por exemplo, ao cadastrar um novo **Ambiente**, o campo de seleção de **Unidade** mostra apenas as unidades previamente cadastradas no banco de dados. Da mesma forma, ao cadastrar um **Dispositivo**, o campo de seleção de **Ambiente** lista apenas os ambientes já existentes, evitando inconsistências e assegurando a correta associação entre os elementos do sistema. Essa abordagem facilita o uso e mantém a coerência entre os diferentes níveis de cadastro, tornando o gerenciamento dos recursos mais intuitivo e organizado para o usuário.

A tela apresentada a seguir (Figura 8) exibe a listagem completa de todos os **recursos cadastrados no sistema**, incluindo **Unidades**, **Ambientes** e **Dispositivos**. Nessa interface, o usuário pode visualizar de forma organizada todas as informações registradas, facilitando o acompanhamento e o gerenciamento dos elementos do sistema.

Unidades Cadastradas

ID	NOME	CÓDIGO DE ACESSO	CEP	AÇÕES
12	Matriz São Paulo	123456	13504822	<button>Editar</button> <button>Excluir</button>

Ambientes Cadastrados

ID	NOME	UNIDADE	AÇÕES
6	Setor II	Matriz São Paulo	<button>Editar</button> <button>Excluir</button>

Dispositivos Cadastrados

ID	NOME	TIPO	AMBENTE	STATUS	AÇÕES
6	Máquina de Produção	Motores	Setor II	Ativo	<button>Editar</button> <button>Excluir</button>

Figura 8 – Visualização dos recursos cadastrados no sistema.

Com a exibição consolidada desses dados, é possível verificar as associações existentes entre as entidades, garantindo uma visão geral da estrutura cadastrada.

Dessa forma, toda a parte de interface do sistema foi desenvolvida com foco na usabilidade e na integração direta com o backend, proporcionando uma navegação simples e intuitiva. As funcionalidades de cadastro, listagem e associação entre os diferentes recursos — Unidades, Ambientes e Dispositivos — garantem uma experiência completa ao usuário, permitindo o gerenciamento eficiente das informações de forma prática e visual. Essa etapa consolida o frontend como uma camada essencial para a interação com o sistema, complementando o funcionamento da aplicação como um todo.

4.2 Back-end

Já o backend do sistema foi desenvolvido utilizando a linguagem Java, com o framework Spring Boot, responsável por gerenciar toda a lógica de funcionamento, comunicação com o banco de dados e controle de acesso dos usuários.

4.2.1 Autenticação e Segurança

Nessa etapa do projeto, foram implementadas as funcionalidades principais de login e cadastro de usuários, garantindo a integração entre a interface do sistema e o banco de dados relacional.

Durante o processo de cadastro, o backend recebe os dados enviados pelo formulário do usuário e realiza uma série de validações para assegurar que todas as informações estejam corretas e completas. Além disso, foi implementado um mecanismo de verificação de e-mail duplicado, evitando o registro de contas já existentes. Após a validação, os dados são inseridos na tabela de usuários do banco de dados.

Para garantir a segurança das informações sensíveis, especialmente das senhas, foi aplicado o Spring Security em conjunto com o algoritmo BCrypt, responsável por criptografar as senhas antes do armazenamento no banco de dados. Esse processo converte a senha em um hash seguro, impedindo sua visualização direta, mesmo por administradores do sistema.

Além da criptografia, foi implementado o JWT (JSON Web Token) como método de autenticação e controle de acesso. Esse token é gerado automaticamente após a validação das credenciais do usuário e contém informações codificadas que identificam a sessão ativa. O token é então enviado ao cliente, que o utiliza em todas as requisições subsequentes a endpoints protegidos, substituindo a necessidade de manter sessões no servidor. Essa abordagem torna a comunicação mais leve, escalável e segura, uma vez que o servidor apenas valida o token em cada requisição, sem armazenar informações de sessão.

O Spring Security, em conjunto com o JWT, também permite a definição de níveis de permissão e restrições de acesso a determinadas rotas, garantindo que apenas usuários autenticados e com privilégios adequados possam acessar áreas específicas do sistema. Dessa forma, o controle de autenticação e autorização foi estruturado de maneira robusta, seguindo as boas práticas de segurança em aplicações web modernas.

A funcionalidade de login foi desenvolvida para validar as credenciais do usuário, gerar o token JWT e estabelecer a autenticação segura. Quando as informações são confirmadas, o sistema redireciona o usuário para a área principal, onde poderá acessar os recursos disponíveis de acordo com seu nível de permissão. Caso os dados estejam incorretos, são exibidas mensagens de erro orientando o usuário a corrigir as informações.

Essa implementação do backend assegura o funcionamento dinâmico e seguro do sistema, promovendo a integração entre o frontend e o banco de dados, além de seguir boas práticas de desenvolvimento web, como validação de entrada, criptografia de senhas, autenticação via token JWT e controle de autorização.

4.2.2 Gerenciamento de Recursos e Banco de Dados Local (MySQL)

Nesta fase do desenvolvimento, o backend foi ampliado para contemplar as operações de CRUD (Create, Read, Update e Delete) dos principais recursos do sistema: unidades, ambientes e dispositivos. Essa expansão teve como objetivo permitir o gerenciamento completo da estrutura monitorada, mantendo a integração entre o frontend e o banco de dados de forma eficiente e organizada.

Durante o desenvolvimento local, foi utilizado o MySQL como sistema gerenciador de banco de dados (SGBD), devido à sua facilidade de configuração, compatibilidade com o Spring Data JPA e bom desempenho para testes e validações. As entidades foram modeladas e estruturadas diretamente no MySQL, o que possibilitou realizar consultas e operações de persistência com segurança e rapidez durante as etapas iniciais do projeto.

Cada entidade foi modelada de acordo com suas relações hierárquicas — as unidades representam os espaços principais, os ambientes pertencem a essas unidades, e os dispositivos estão associados a cada ambiente. Essa modelagem exigiu a definição de chaves estrangeiras e o mapeamento correto entre as tabelas, garantindo a integridade dos dados e facilitando a navegação entre os diferentes níveis do sistema.

No backend, foram criadas rotas específicas para cada operação, utilizando os recursos do Spring Boot, como os controladores REST, os repositórios JPA e os serviços de negócio responsáveis por validar e processar as requisições. Assim, é possível cadastrar novos registros, listar informações existentes, atualizar dados e excluir elementos de forma segura.

No frontend, as funcionalidades foram desenvolvidas com HTML, CSS e JavaScript, garantindo uma interface intuitiva e responsiva para interação com o usuário. Através de requisições HTTP, o sistema se comunica com o backend para exibir e manipular os dados

em tempo real. Cada operação do CRUD foi acompanhada por mensagens de retorno, confirmando o sucesso das ações ou informando eventuais erros, o que contribui para uma experiência de uso mais clara e eficiente.

Além disso, foi incorporada ao projeto a documentação da API utilizando o Swagger, uma ferramenta que permite visualizar e testar todos os endpoints disponíveis de forma interativa. A integração do Swagger ao Spring Boot facilita a compreensão do funcionamento da API, descrevendo os métodos, parâmetros e respostas de cada rota. Essa documentação automatizada tornou o processo de desenvolvimento mais ágil, auxiliando na verificação das funcionalidades implementadas e servindo como base para futuras manutenções e ampliações do sistema.

Essa etapa representou um avanço importante no projeto, pois consolidou a integração completa entre as camadas da aplicação e proporcionou à equipe um aprendizado mais aprofundado sobre manipulação de dados, estruturação de endpoints, documentação de APIs e relacionamento entre entidades em um sistema web moderno.

4.2.3 Testes Automatizados e Validação do Sistema

Outra melhoria importante implementada foi a criação de testes automatizados utilizando o JUnit, com foco na validação das funcionalidades relacionadas aos usuários. Esses testes foram desenvolvidos para garantir o correto funcionamento das operações de login, cadastro e manipulação de dados dos usuários, assegurando que cada método da aplicação se comporte conforme o esperado.

Os testes foram estruturados para verificar desde a integridade dos dados recebidos até as respostas retornadas pelo sistema, cobrindo diferentes cenários — como inserção de informações válidas, tentativas de cadastro duplicado e autenticação com credenciais incorretas. Com isso, foi possível identificar e corrigir possíveis inconsistências antes mesmo da execução completa do sistema, aumentando a confiabilidade do backend.

A utilização do JUnit também contribuiu para a adoção de boas práticas de desenvolvimento orientado a testes (TDD – Test Driven Development), permitindo que o código fosse continuamente validado e mantido de forma mais segura. Dessa forma, o processo de testes automatizados não apenas garantiu a estabilidade do sistema, mas também facilitou futuras manutenções e expansão do projeto.

4.2.4 Deploy e Migração para Produção (Render e Vercel)

Para disponibilizar o sistema de forma acessível e funcional fora do ambiente de desenvolvimento local, foi realizado o processo de deploy tanto do backend quanto do frontend em plataformas de hospedagem em nuvem, garantindo a execução completa da aplicação em ambiente online.

O backend foi hospedado na plataforma Render, um serviço de nuvem que oferece suporte a aplicações desenvolvidas em Spring Boot. Durante o desenvolvimento local, o sistema utilizava o banco MySQL, mas para o ambiente de produção foi adotado o PostgreSQL, devido à sua integração nativa com a plataforma Render e melhor compatibilidade para deploy em nuvem. Essa migração foi feita de forma transparente, mantendo a estrutura relacional das tabelas e as chaves estrangeiras definidas durante o desenvolvimento.

O processo iniciou-se com a preparação do projeto para produção, incluindo ajustes no arquivo application.properties e na configuração das variáveis de ambiente do servidor, responsáveis por armazenar dados sensíveis como as credenciais do banco de dados e a porta de execução. O código-fonte foi versionado no GitHub, permitindo a integração contínua com o Render. Assim, a cada atualização na branch principal, o serviço realiza automaticamente o build e o deploy da aplicação.

O banco de dados relacional PostgreSQL foi configurado diretamente no Render, garantindo compatibilidade e comunicação segura com o backend. Essa integração foi feita por meio de variáveis de ambiente seguras, assegurando a proteção das informações armazenadas. Com o deploy concluído, a API passou a estar acessível por meio de uma URL pública, permitindo que o frontend consumisse os endpoints remotamente e o sistema fosse testado em condições semelhantes às de produção.

Já o frontend foi implantado na plataforma Vercel, escolhida por sua praticidade e integração nativa com projetos baseados em HTML, CSS e JavaScript. O processo de deploy foi realizado a partir do mesmo repositório versionado no GitHub, permitindo que o Vercel fizesse o build automático do projeto e disponibilizasse uma URL pública para acesso ao sistema. Essa abordagem garantiu que o frontend estivesse conectado de forma direta ao backend hospedado no Render, possibilitando a comunicação em tempo real entre as duas camadas da aplicação.

O deploy em nuvem trouxe maior confiabilidade, escalabilidade e facilidade de manutenção ao projeto, além de permitir o acesso remoto ao sistema de qualquer

dispositivo conectado à internet. Essa etapa consolidou o ciclo completo de desenvolvimento e entrega do sistema, tornando-o plenamente funcional, acessível e pronto para futuras atualizações e aprimoramentos.

Para facilitar o acesso e validação do sistema em ambiente de produção, seguem abaixo os links públicos correspondentes às aplicações hospedadas:

- **Frontend (Vercel):** <https://energyflow-plataform.vercel.app>
Interface web do sistema, desenvolvida em HTML, CSS e JavaScript, responsável pela interação com o usuário e comunicação com a API hospedada no backend.
- **Backend (Render):** <https://energyflow-backend-i3xs.onrender.com>
API REST desenvolvida em Spring Boot, responsável pelo processamento das requisições, persistência dos dados e integração com o banco PostgreSQL.

 *Observação:* Por estar hospedada em um plano gratuito na plataforma Render, a API pode entrar em modo de hibernação após um período de inatividade. Assim, ao acessá-la pela primeira vez, pode ser necessário aguardar alguns minutos para que o servidor seja reativado e o serviço volte a responder normalmente.

O acesso para o diretório que contém os arquivos HTML, CSS, JavaScript e os endpoints desenvolvidos em Java com Spring Boot, estão na plataforma do GitHub, na URL: [<https://github.com/energyflow-plataform/energyflow/tree/main/frontend>].

5 Link de Acesso do GitHub

Repositório do projeto:

[<https://github.com/energyflow-plataform/energyflow>]

6 Link de Acesso ao Cronograma do Projeto (Trello)

Cronograma do projeto:

[<https://trello.com/invite/b/68a65d7e2c98c96ee021a04e/ATTlb3f9f63b6fb5a233edb8235325779d02FE35C0EF/pi2s2025neosolaris>]

7 Considerações Finais

Durante esta etapa do desenvolvimento do projeto, foi possível consolidar de forma significativa os conhecimentos adquiridos em desenvolvimento web, integração entre frontend e backend, e boas práticas de engenharia de software. A evolução do sistema, que anteriormente contava apenas com o módulo de autenticação, avançou para a implementação completa das funcionalidades de CRUD das entidades unidades, ambientes e dispositivos, além da documentação da API com Swagger e da inclusão de testes automatizados com JUnit.

O uso do Spring Boot como base do backend possibilitou a criação de uma arquitetura mais organizada e escalável, enquanto o Spring Security, aliado ao JWT (JSON Web Token), trouxe maior segurança à aplicação, garantindo o controle de acesso e a proteção dos dados dos usuários. Essas práticas proporcionaram uma experiência real de desenvolvimento de aplicações corporativas, nas quais a autenticação, a autorização e a confiabilidade do sistema são fatores essenciais.

No frontend, o uso de HTML, CSS e JavaScript permitiu o desenvolvimento de uma interface intuitiva e funcional, reforçando o entendimento sobre consumo de APIs REST e interação dinâmica com o backend. A integração entre as camadas foi fundamental para compreender o fluxo completo de comunicação entre cliente, servidor e banco de dados.

A adoção do Swagger contribuiu para a documentação padronizada da API, facilitando o entendimento e os testes dos endpoints, enquanto a implementação do JUnit assegurou a qualidade do código e a estabilidade das funcionalidades críticas do sistema. Essas ferramentas consolidaram práticas profissionais de desenvolvimento, teste e manutenção de software.

De modo geral, essa etapa representou um avanço expressivo no projeto, permitindo à equipe aprimorar tanto as competências técnicas quanto a capacidade de planejamento, colaboração e resolução de problemas. O resultado é um sistema mais completo, seguro e confiável, que reflete o progresso contínuo do desenvolvimento e o aprendizado obtido ao longo do processo.

Referências Bibliográficas

AL-ALI, A. R; ALIKARAR, Mazin; GUPTA, Ragini; RASHID, Mohammed; ZUALKERMAN, Imran. **A smart home energy management system using IoT and big data analytics approach.**

IEEE Transactions on Consumer Electronics, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8246800>>. Acesso em: 29 mar 2025.

HERTZ ELÉTRICA. **Problema: Alto Consumo De Energia - Soluções E Dicas.** Disponível em: <<https://www.hertzeletrica.com.br/glossario/problema-alto-consumo-de-energia-solucoes-dicas/>>. Acesso em: 3 abr 2025.

JWT Official Documentation. **Introduction to JSON Web Tokens.** Disponível em: <<https://www.jwt.io/introduction#what-is-json-web-token>>. Acesso em: 20 out 2025.

JUnit Official Documentation. **JUnit User Guide.** Disponível em: <<https://docs.junit.org/current/user-guide/>>. Acesso em: 20 de out 2025.

MAGALHÃES, L. C. **Orientações gerais para conservação de energia elétrica em edificações.** ELETROBRAS/PROCEL - 2a edição – 2002.

Render Official Documentation. **Deploying on Render - Understand how deploys work.** Disponível em: <<https://render.com/docs/deploys>>. Acesso em: 20 out 2025.

Spring Security, Spring Official Documentation. **Authorize HttpServletRequests.** Disponível em: <<https://docs.spring.io/spring-security/reference/servlet/authorization/authorize-http-requests.html>>. Acesso em: 20 out 2025.

Spring Security, Spring Official Documentation. **Password Storage.** Disponível em: <<https://docs.spring.io/spring-security/reference/features/authentication/password-storage.html>>. Acesso em: 20 out 2025.

Swagger Official Documentation, OpenAPI Documentation. **What is OpenAPI?** Disponível em: <https://swagger.io/docs/specification/v3_0/about/>. Acesso em: 20 out 2025.

Vercel Official Documentation. **Vercel Documentation - Get Started with Vercel.** Disponível em: <<https://vercel.com/docs>>. Acesso em: 20 out 2025.