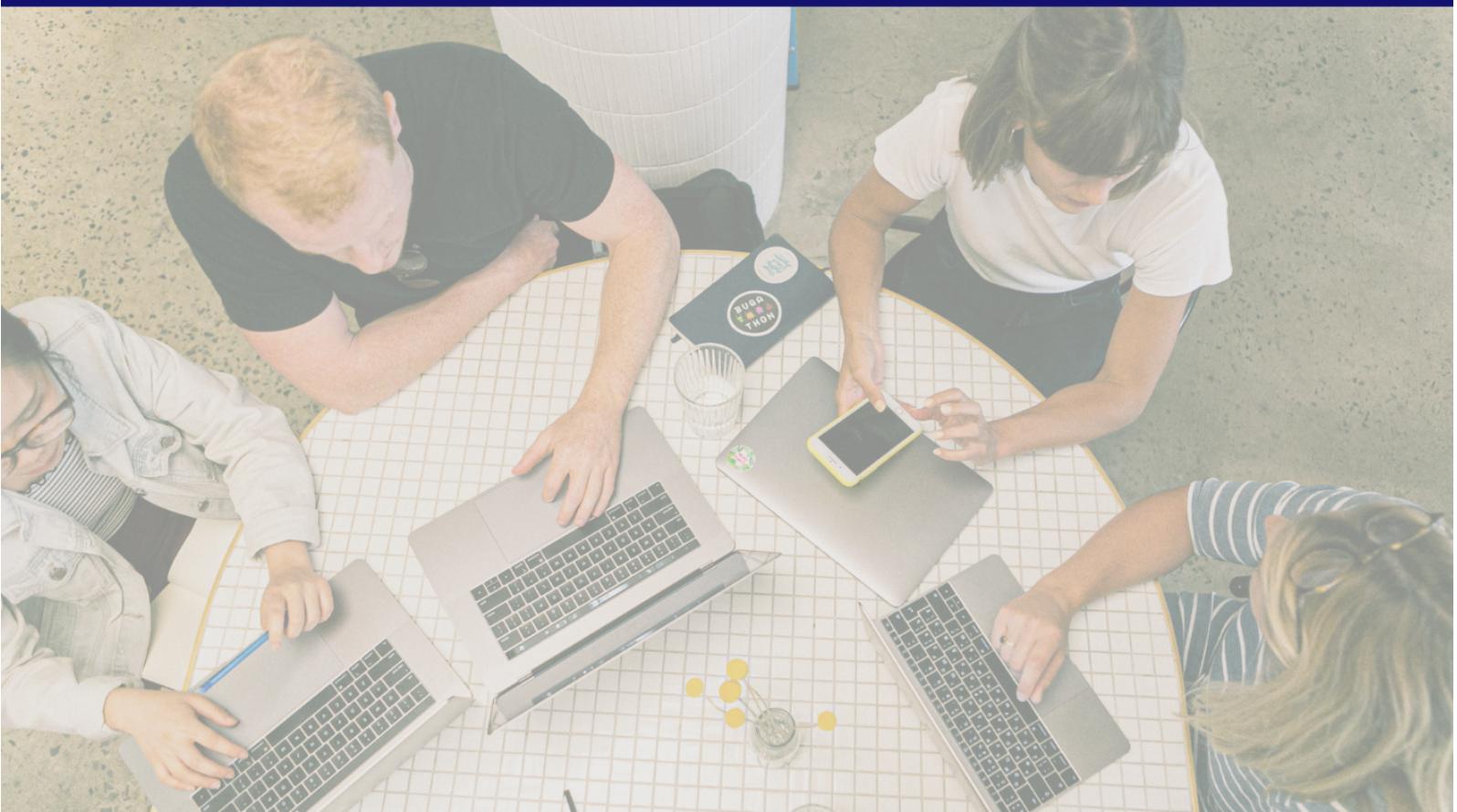




**UNifeob**  
| ESCOLA DE NEGÓCIOS

**2023**

# PROJETO INTEGRADO



UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**  
**CIÊNCIA DA COMPUTAÇÃO**

**PROJETO INTEGRADO**  
Sistema de Gerenciamento de Energia

**UNIFEOB**

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2023

UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**  
**CIÊNCIA DA COMPUTAÇÃO**

**PROJETO INTEGRADO**

Sistema de Gerenciamento de Energia

**UNIFEOB**

MÓDULO COMPUTAÇÃO EM NUVEM

Estrutura de Dados – Prof. Mauro Glória

Linguagem e Técnicas de Programação – Prof. Marcelo Ciacco de Almeida

Tópicos Avançados de Banco de Dados – Prof. Max Streicher Vallim

Computação em Nuvem – Prof. Rodrigo Marudi de Oliveira

Projeto de Computação em Nuvem – Profa. Mariângela Martimbianco Santos

Estudantes:

Davi Pinto Santana Rodrigues, 22000584

Fernando Pagliarini Furlanetto, 22000293

João Otávio Rehder Mendes, 22000508

Kauã da Silva Lima Amorim, 22000589

Kayque Silva Zanelli, 22001114

Lucas de Oliveira Barreiro, 22000100

Victor Castilho, 22001035

Vinicius Miguel Quintino, 22001127

SÃO JOÃO DA BOA VISTA, SP  
NOVEMBRO 2023

# SUMÁRIO

1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	5
3. PROJETO INTEGRADO	6
3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS	6
3.1.1 MODELO LÓGICO	6
3.1.2 MODELO FÍSICO	7
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	10
3.2.1 PROTOTIPAÇÃO	10
3.2.2 FRONT-END	15
3.2.2.1 HTML	15
3.2.3 BACK-END	18
3.2.4 CRUD	19
3.3 COMPUTAÇÃO EM NUVEM	24
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	24
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	24
3.3.3 VANTAGENS DA CLOUD COMPUTING	24
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	24
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	25
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	25
3.3.7 GOOGLE CLOUD ou AWS	26
3.4 ESTRUTURA DE DADOS	26
3.4.1 LEVANTAMENTO DE REQUISITOS	26
3.4.2 VALIDAÇÃO DOS REQUISITOS	27
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	42
3.5.1 ENFRENTANDO ESTEREÓTIPOS	42
3.5.2 ESTUDANTES NA PRÁTICA	44
4. CONCLUSÃO	46
REFERÊNCIAS	47
ANEXOS	49

# 1. INTRODUÇÃO

A proposta desse projeto é a criação de software de gerenciamento de um sistema de energia pública funcionando a partir da captação de energia solar. A ideia deste projeto é atender ao objetivo onze dos dezessete Objetivos de Desenvolvimento Sustentável da ONU, onde o objetivo onze visa ter cidades e comunidades sustentáveis. Para isso foi realizado um estudo a partir de diversos modelos físicos do que seria um poste funcionando com a captação de energia solar e utilizando a mesma durante a noite, dentre os modelos pesquisados, foram utilizados modelos que possuem funcionalidades como sensor de aproximação e capacidade de diminuir a quantidade de luz emitida visando menor consumo. Também foi realizado um estudo em relação aos benefícios que este projeto traria as comunidades e gestores de energia, além de uma grande economia para as prefeituras ou órgãos de gestão de energia tanto na implementação quanto suporte e gastos com manutenção, este projeto iria ajudar na diminuição do uso de métodos antigos de captação de energia como a hidrelétrica e a termelétrica, métodos esses que geram um grande desmatamento para criação além do alto custo. O software desenvolvido a partir de protótipos pesquisados funciona com um banco de dados que recebe as informações como energia captada e energia utilizada, e disponibiliza isso a um sistema que permite a manipulação destas informações além da manipulação das estações de energia. Para criação desta documentação foram utilizados os conhecimentos desenvolvidos em sala de aula nos módulos de Computação em Nuvem, Tópico Avançado de Banco de Dados, Linguagem técnica de programação e Estrutura de dados, além das informações e conhecimentos adquiridos durante o desenvolvimento do projeto.

## **2. DESCRIÇÃO DA EMPRESA**

A empresa alvo do projeto realizado têm razão social UNIFEOB – Centro Universitário da Fundação de Ensino Octávio Bastos, mais conhecido pelo seu nome fantasia FEOB com CNPJ 59.764.555/0001-52 de São João da Boa Vista – SP localizado na Av. Dr. Otávio da Silva Bastos, 2439 - Jardim Nova São João.

Sua principal atividade é a formação de profissionais para o mercado de trabalho a partir de cursos em diversas áreas de conhecimento, suas demais atividades são projetos de extensão visando um maior aprofundamento do aluno em um determinado tema, atividades extracurriculares, como estágios, campanhas e serviços à sociedade.

A universidade vem se destacando cada vez mais e liderando o ranking de melhores universidades da região, gerando um número cada vez maior de matrículas e formando cada vez mais melhores profissionais.

## **3. PROJETO INTEGRADO**

Nesta etapa do PI serão apresentados os conteúdos que cada unidade de estudo utilizará para realizar o projeto, assim como a forma que serão aplicados na empresa escolhida para a realização do projeto.

### **3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS**

Banco de Dados é uma área de conhecimento que engloba a coleta e armazenamento de dados para auxiliar na criação de um histórico de movimentações de dados nas organizações. Por meio do uso de ferramentas e técnicas específicas, o Banco de Dados permite armazenar dados brutos em informações relevantes e acionáveis. Essas informações são coletadas e armazenadas para serem acessadas em qualquer momento com a finalidade de contribuir para o desempenho e o sucesso empresarial. Ele desempenha um papel fundamental na gestão eficiente das organizações, proporcionando uma visão holística e embasada para a tomada de decisões informadas. O professor Max Streicher Vallim foi responsável pelo módulo de Tópicos Avançados de Banco de Dados, que tem como objetivo acompanhar e orientar o processo da criação dos bancos de dados, fornecendo conhecimentos fundamentais para a realização do projeto.

#### **3.1.1 MODELO LÓGICO**

O Modelo Lógico é a etapa onde será mapeado o conceito dos modelos de entidade-relacionamento em objetivos de Bancos de Dados. Nesta fase foram criados os modelos internos de bancos de dados, com detalhes sobre tabelas, relacionamentos, regras, metadados das colunas (tipo, tamanho, obrigatoriedade, ...), visões, etc.

Ao final, o resultado de um projeto lógico é um esquema do banco de dados parecido com o modelo conceitual, porém com mais detalhes de banco de dados e não apenas conceitos.

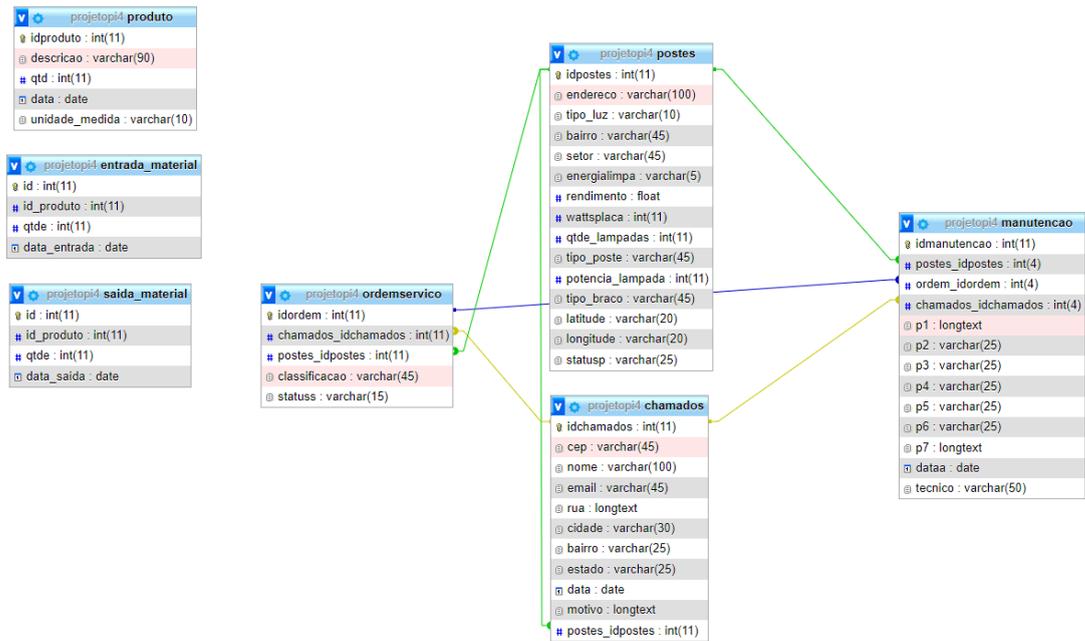


Figura 1 - Modelo lógico.

### 3.1.2 MODELO FÍSICO

O Modelo Físico é a parte final do projeto de Banco de Dados, nesta etapa foram definidos os detalhes técnicos da implementação do banco de dados, por exemplo a forma como os dados serão armazenados, os scripts para a criação dos objetos no banco de dados (tabelas, visões, colunas, funções, triggers e stored procedures), permissão de acesso de usuário, etc. Esta etapa é fortemente ligada ao SGBD que será utilizado.

A partir dessa visão, construímos nossa base de dados a partir de nosso modelo lógico, onde adicionamos alguns triggers e stored procedures para realização de funções autônomas em nosso banco de dados.

Realizamos a construção de 7 tabelas em nosso projeto

1. Tabela “produtos”: Realiza os cadastros de matérias primas do sistema além de atuar como estoque de materiais.
2. Tabela “entrada\_material”: A partir da tabela produto realiza a soma de novas entradas no de matéria prima através de um stored procedure e triggers de inserção, update ou delete.
3. Tabela “saida\_material”: Esta tabela executa através de um stored procedure e triggers de inserção, update ou delete a saída de matérias primas de nosso estoque.
4. Tabela “chamados”: Esta tabela funciona como um “balde” as solicitações de clientes chegam nesta tabela contendo informações da ocorrência.

5. Tabela “ordemservico”: Esta tabela recebe os chamados citados acima após uma classificação, a partir disso a ocorrência é encaminhada para a equipe de manutenção.
6. Tabela “postes”: A tabela poste realiza o cadastro dos pontos de luzes da cidade, assim que os dados são inseridos no banco de dados, essa tabela é solicitada pelo sistema para referenciamento dos pontos de luzes no mapa.
7. Tabela “manutencao”: Esta tabela funciona como um check-list de serviço, onde após a abertura da ordem de serviço, o setor responsável responde a um questionário a respeito do serviço prestado.

### 3.1.3 STORED PROCEDURES

Stored Procedures é classificado como um conjunto de instruções SQL pré-compiladas armazenadas em um banco de dados. Essas instruções são chamadas por um identificador e podem aceitar um período passado durante a execução. Os procedimentos armazenados são usados principalmente para encapsular a lógica de negócios e as operações de dados no banco de dados

A figura a seguir, se trata da stored procedure de atualização de estoque.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `SP_AtualizaEstoque`(`id_est` int, `qtde` int, datae Date)
2 BEGIN
3     declare contador int(11);
4
5     SELECT count(*) into contador FROM produto WHERE idproduto = id_est;
6
7     IF contador > 0 THEN
8         UPDATE produto SET qtd=qtd + qtde, data=datae
9         WHERE idproduto = id_est;
10    ELSE
11        INSERT INTO produto (idproduto, qtd, data ) values (id_est, qtde, datae);
12    END IF;
13 END
```

Figura 2 - Stored Procedure.

### 3.1.4 TRIGGERS

Nosso projeto utilizou de Triggers, um conjunto de informações configuradas para serem “disparadas”, automaticamente a partir de uma modificação em alguma tabela determinada, como exemplo em nosso PI, após uma entrada, saída e atualização, os triggers são encontrados nas tabelas “entradas\_material” e “saida\_material” demonstrados abaixo:

```

-- Criar Triggers - Entrada de Produtos Apos Inserção --

DELIMITER //
CREATE TRIGGER `TRG_EntradaMaterial_AI` AFTER INSERT ON `entrada_material`
FOR EACH ROW
) BEGIN
    CALL SP_AtualizaEstoque (new.id_produto, new.qtde, new.data_entrada);
- END //
DELIMITER ;

-- Criar Triggers - Entrada de Produtos Apos Atualização --

DELIMITER //
CREATE TRIGGER `TRG_EntradaMaterial_AU` AFTER UPDATE ON `entrada_material`
FOR EACH ROW
) BEGIN
    CALL SP_AtualizaEstoque (new.id_produto, new.qtde - old.qtde, new.data_entrada);
- END //
DELIMITER ;

-- Criar Triggers - Entrada de Produtos Apos Exclusão --

DELIMITER //
CREATE TRIGGER `TRG_EntradaMaterial_AD` AFTER DELETE ON `entrada_material`
FOR EACH ROW
) BEGIN
    CALL SP_AtualizaEstoque (old.id_produto, old.qtde * -1, old.data_entrada);
- END //
DELIMITER ;

```

Figura 3 - Trigger “entrada\_material”.

```

-- Criar Triggers - Saida de Produtos Apos Inserção --

DELIMITER //
CREATE TRIGGER `TRG_SaidaMaterial_AI` AFTER INSERT ON `saida_material`
FOR EACH ROW
) BEGIN
    CALL SP_AtualizaEstoque (new.id_produto, new.qtde * -1, new.data_saida);
- END //
DELIMITER ;

-- Criar Triggers - Saida de Produtos Apos Atualização --

DELIMITER //
CREATE TRIGGER `TRG_SaidaMaterial_AU` AFTER UPDATE ON `saida_material`
FOR EACH ROW
) BEGIN
    CALL SP_AtualizaEstoque (new.id_produto, old.qtde - new.qtde, new.data_saida);
- END //
DELIMITER ;

-- Criar Triggers - Saida de Produtos Apos Exclusão --

DELIMITER //
CREATE TRIGGER `TRG_SaidaMaterial_AD` AFTER DELETE ON `saida_material`
FOR EACH ROW
) BEGIN
    CALL SP_AtualizaEstoque (old.id_produto, old.qtde, old.data_saida);
- END //
DELIMITER ;

```

Figura 4 - Trigger “saida\_material”.

## 3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

A disciplina de Linguagem Técnicas de Programação tem como objetivo a demonstração de metodologias para desenvolvimento de sistemas, desde o protótipo, até a finalização do produto físico.

### 3.2.1 PROTOTIPAÇÃO

Para realização da prototipação, foi utilizada a ferramenta de prototipação online chamada: Figma, onde o grupo idealizou como seriam as janelas e funcionalidades de maneira inicial do sistema. Realizamos na plataforma o esboço de idealização, inicialmente o

grupo optou pela utilização somente de cores neutras, com foco na estruturação de layout, nessa etapa do projeto, tipografia, cores, nomes e logotipo ainda não tinham sido projetados de maneira concreta e definitiva, deixando abertura para futuras modificações, de acordo com o desenvolvimento.

Nas imagens apresentadas abaixo, é apresentado como ficou a versão de prototipação do sistema:

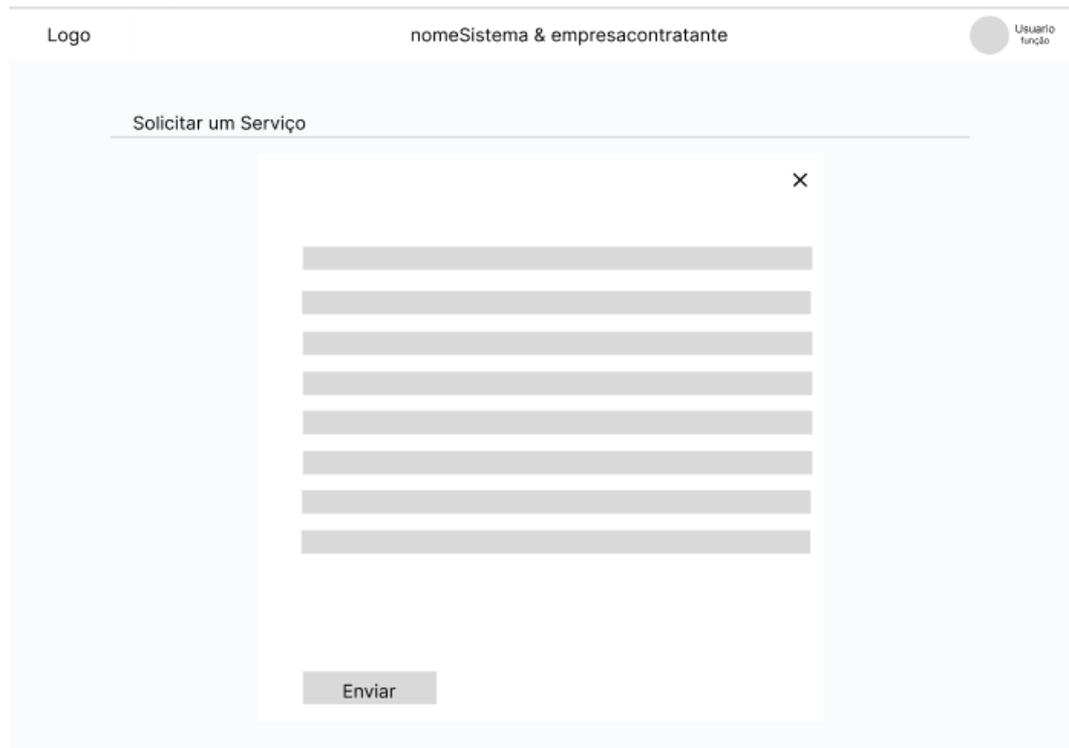


Figura 1 - Solicitação de chamado.

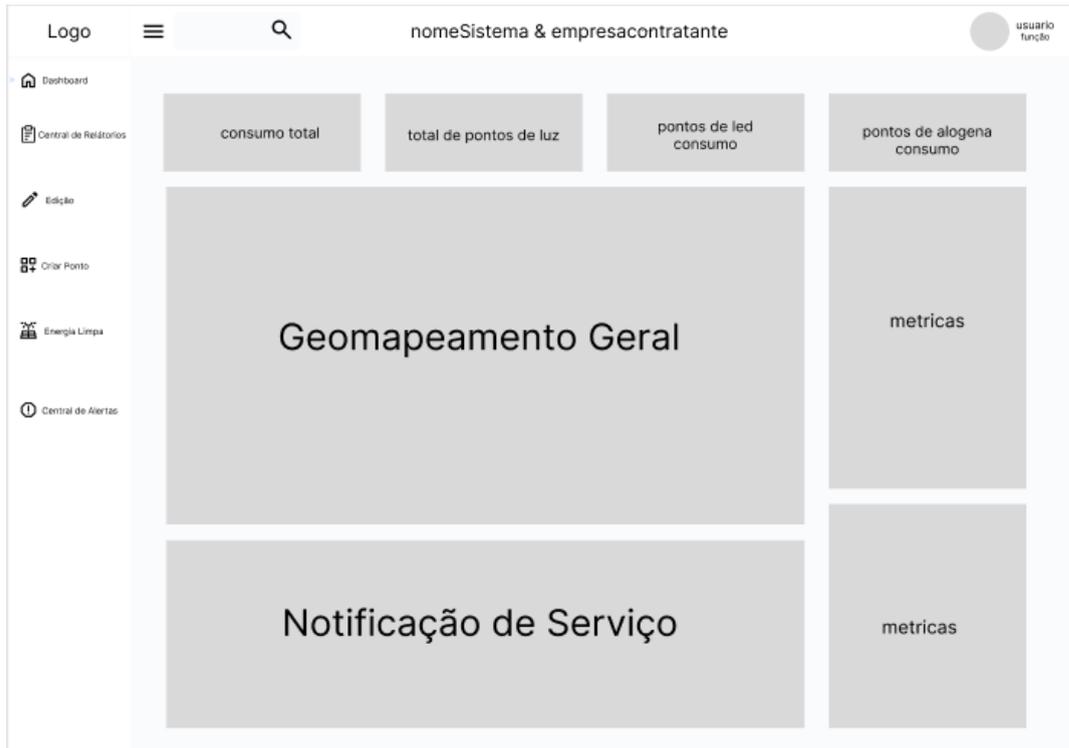


Figura 2 - Dashboard principal.

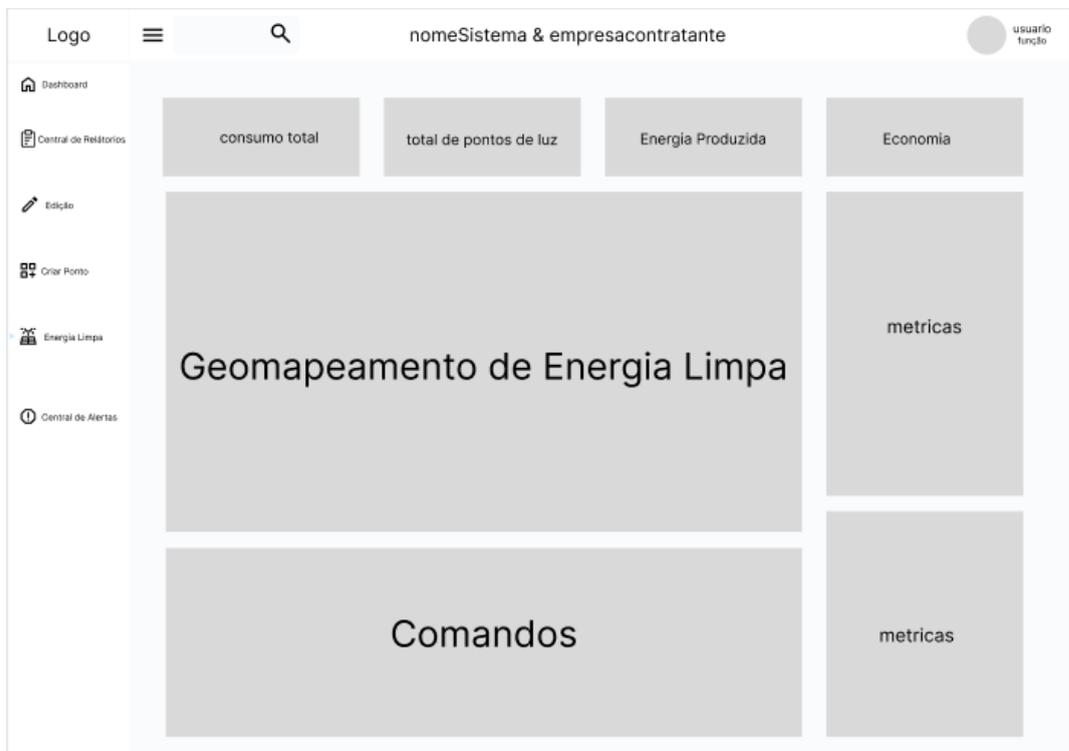


Figura 3 - Dashboard de Energia Limpa.

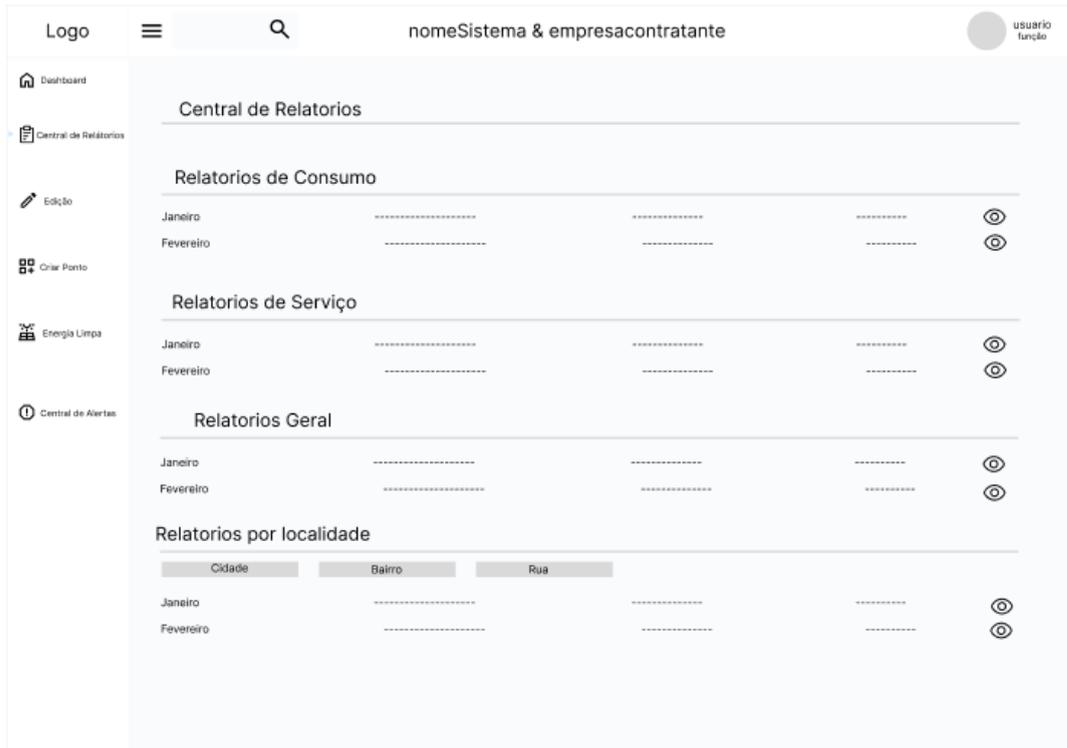


Figura 4 - Central de Relatórios.

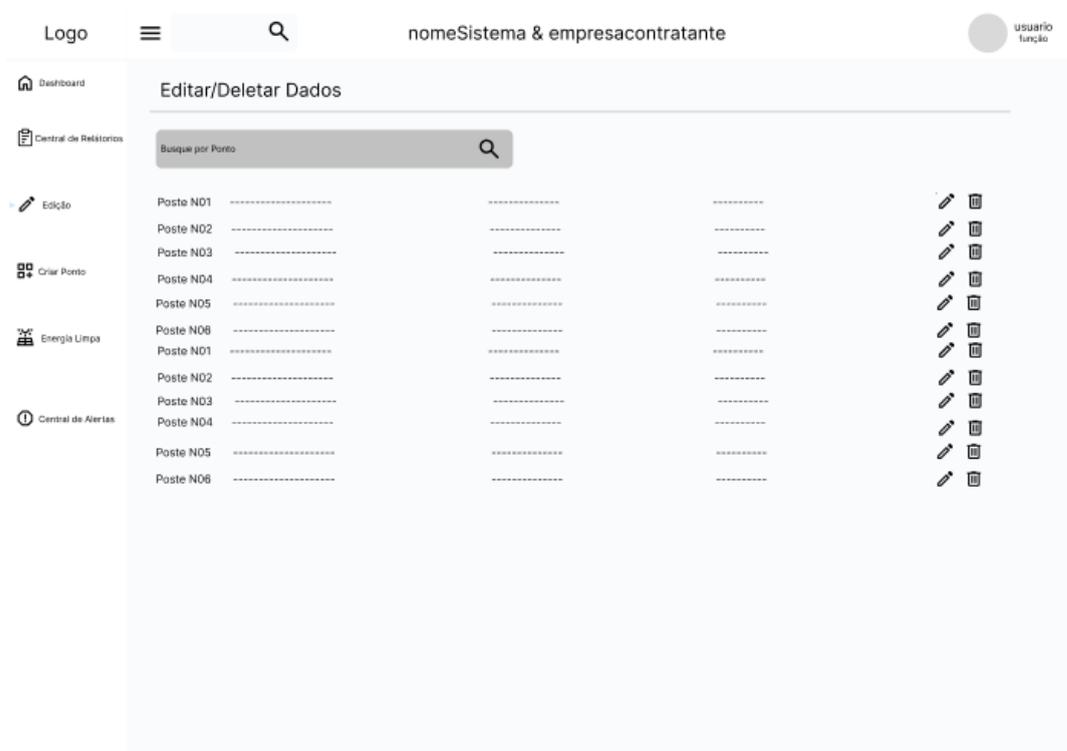


Figura 5 - Janela de Edição e Exclusão.

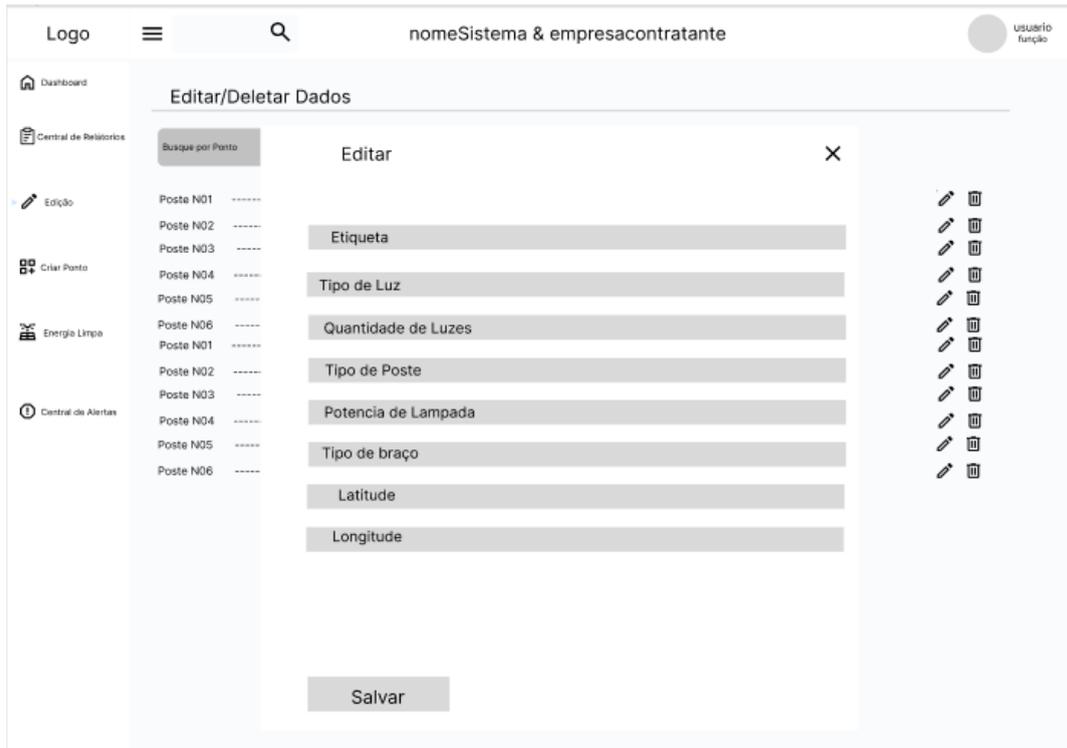


Figura 5 - Edição de dados.

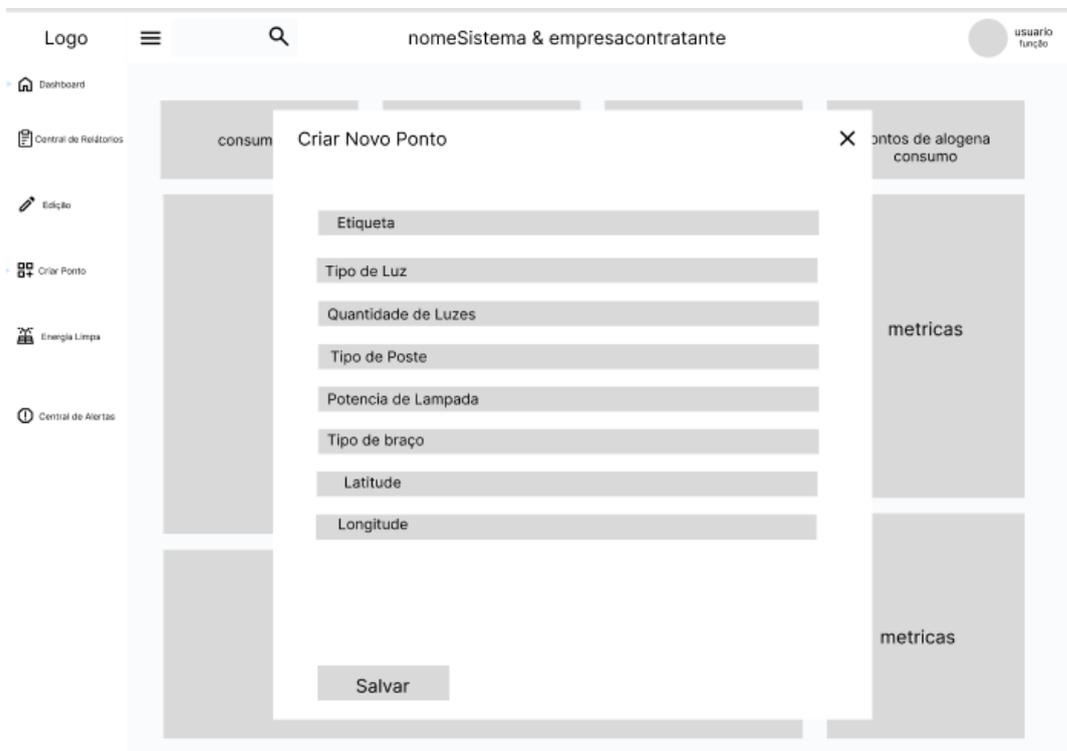


Figura 6 - Solicitação de chamado.



Figura 7 - **Central de Alertas.**

O desenvolvimento do protótipo foi de alta importância para o desenvolvimento do projeto, visto que grande parte das janelas desenvolvidas foram baseadas nele.

### **3.2.2 FRONT-END**

#### **3.2.2.1 HTML**

“O HTML é uma linguagem de marcação: um conjunto de indicadores (chamados de elementos) que define o texto e os objetos do documento e que pode ser incorporado ao seu processador de texto favorito.” (MARCON; NEVES, 1999, p.8).

A partir do uso de hipertexto, essa linguagem de marcação pode agregar blocos de informações, como conteúdo multimídia, que serão acessados por meio de hiperlinks ou links. Este conceito pode maximizar o acesso. Uma pequena quantidade de informações interconectadas, que juntas formam uma grande rede de informações, a Internet segundo a W3C ESCRITÓRIO BRASIL (2016).

De acordo com Silva (2007), o HTML é uma linguagem utilizada para gerar páginas web. Essas páginas são copiadas pelo navegador instalado no dispositivo. Esses navegadores podem ler a linguagem HTML e processar o conteúdo da página a partir daí, como texto, tabelas, Formulários de entrada de dados, entre outras coisas, como mostra a figura abaixo,

onde se tem um a estrutura de códigos do HTML e leitura do código feito através de um navegador instalado em uma máquina local.

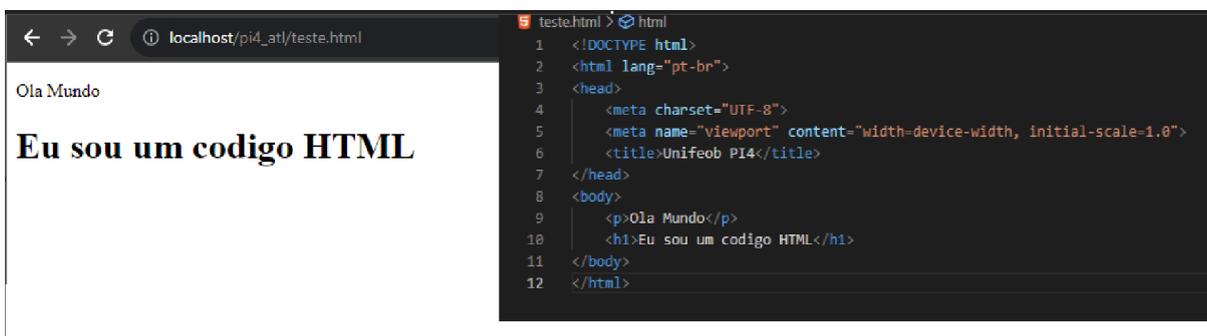


Figura 8 - Estrutura de código HTML.

A criação da página é feita por meio de tags, que informam ao navegador como exibir o conteúdo. Esses são chamados de tags, através dos quais se pode informar se a estrutura é um parágrafo, ou tabela. O número de tags é grande e há cerca de 116 representações de estrutura na 5ª edição, o objetivo principal desta versão é tornar o HTML mais semântico, ou seja, dar mais objetividade e significado na leitura do código. Seguido de FERREIRA (2011), que diz, o HTML5 também oferece mais recursos para que o Cascading Style Sheet (CSS) e o Javascript possam completar seu trabalho com mais eficiência.

### 3.2.2.2 CSS

Segundo GRANNEL (2007) diz que, o CSS (Cascading Style Sheet) é usado para definir o estilo da forma como a página HTML é exibida na tela. Ou seja, pode garantir a uniformidade do formato das páginas do site, aparência elegante. Foi criado para separar a estrutura do documento da aparência.

Como referenciado acima, o CSS (Cascading Style Sheet), tem como principal objetivo deixar o design de um site com um perfil mais elegante, como é mostrado na figura 3, onde se tem a comparação entre a utilização do HTML e CSS.



## Figura 8 - Comparação entre HTML e CSS.

Silva (2011) mencionou que na terceira edição (denominada CSS3), o CSS trouxe cor, alinhamento e espaçamento dos elementos e estilização das imagens entre eles. No entanto, sua maior conquista é o design responsivo.

### 3.2.2.3 Java Script

JavaScript é uma linguagem criada pela Sun MicroSystem Inc. Em parceria com a Netscape, embutida em um código HTML. O que distingue o JavaScript é sua capacidade de interação plena com a página HTML, permitindo que novos pedaços de código HTML ou mesmo JavaScript sejam adicionados dinamicamente, além de possuir a maioria das estruturas típicas de uma linguagem moderna. (SOARES, 2006, p. 48).

O JavaScript é uma linguagem que permite adicionar lógica em páginas escritas no formato HTML. Os parágrafos da lógica do JavaScript podem ser "soltos" ou podem estar ligados à ocorrência de eventos. Parágrafos soltos são executados na ordem em que aparecem na página, enquanto parágrafos que estão vinculados a eventos são executados apenas quando o evento ocorre. Para inserir uma programação de linguagem JavaScript na linguagem HTML, se faz necessário que o início e o fim do conjunto JavaScript sejam marcados com a palavra "Script", de acordo com Goodman (2001), igual o exemplo representado na figura 9 abaixo.

```
teste.html > html
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Unifeob PI4</title>
7 </head>
8 <body>
9   <p>Ola Mundo</p>
10  <h1>Eu sou um codigo HTML</h1>
11
12  <script>
13
14
15    //Script Aqui
16
17  </script>
18 </body>
19 </html>
```

Figura 9 - Abertura e Fechamento de um código JavaScript

Um script JavaScript permite que sejam executadas funções que antes, obrigatoriamente, deviam ser processadas pelo servidor, como validação de campos, abertura

de novas janelas, criação e manipulação de camadas, cálculos simples e complexos etc. (SOARES, 2006, p. 33).

#### **3.2.2.4 Bootstrap**

O framework Bootstrap é conhecido por seu design para auxiliar no desenvolvimento de front-end, tornando o processo mais rápido e fácil. A estrutura oferece suporte a HTML, CSS e JS. Permite a criação de itens de resposta para uso em dispositivos móveis.

Bootstrap é uma plataforma que fornece configurações padrão para elementos de tela da web. O seu principal objetivo é facilitar para os desenvolvedores que não possuem um conhecimento concreto sobre a área as técnicas de design, sendo assim os usuários possam criar páginas de forma única, visualmente bonitas sem muitos transtornos segundo a BOOTSTRAP (2013).

COCHRAN (2012) diz que umas das grandes vantagens de se utilizar a ferramenta de código aberto, ou seja, todos os usuários podem ter acesso é seu sistema de grade. Ele permite que os desenvolvedores programem seus aplicativos de acordo com os padrões de tela, podendo ser executados e ajustados de forma automática ou com apenas algumas alterações no código, e podem ser visualizados perfeitamente em dispositivos de diferentes tamanhos (como telefones celulares, tablets, monitores de pequeno porte, etc...) de forma totalmente responsive (BOOTSTRAP, ano 2013).

### **3.2.3 BACK-END**

#### **3.2.3.1 PHP**

No ano de 1994, um consultor de empresas, chamado Rasmus Lerdorf, enviava o endereço eletrônico de sua página que continha seu currículo para os mais diversos potenciais empregadores, curioso em saber quem eram esses que estavam consultando sua página, desenvolveu um script, feito com a linguagem Perl, que colhia informações sobre os seus visitantes e média o número de acessos que sua página pessoal tinha. ANSELMO (2002, p. 5)

PHP (Hypertext Preprocessor) é uma linguagem de programação para a web com scripts de código aberto, misturados com HTML (Hypertext Markup Language). O PHP é executado no lado do servidor, o que permite que o site seja executado de forma dinâmica e garante maior segurança da informação.

A figura representada abaixo tem como intuito ilustrar a explicação acima, demonstrando a estrutura de um código PHP.

```

teste.php > html
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Unifeob PI4</title>
7  </head>
8  <body>
9      <p>Ola Mundo</p>
10     <h1>Eu sou um codigo HTML</h1>
11
12     <script>
13
14         //Script Aqui
15
16     </script>
17
18
19
20 <?php
21
22 echo 'Sou texto em PHP';
23
24
25 ?>
26
27 </body>
28 </html>

```

Figura 10 - Estrutura de um código PHP

De acordo com PHP.NET (2014) O PHP é a solução para usar em sua página pessoal, mas a demanda aumentou com os anos e uma nova versão foi lançada, que continha novas funções, como interpretador de comandos SQL (linguagem de consulta Estruturado), orientação a objetos, herança, polimorfismo etc.

Hoje essa linguagem é utilizada em todo o mundo desde criações de páginas pessoais até na utilização de grandes indústrias, já que ela possui extensões com compatibilidade com o Banco de Dados MySQL.

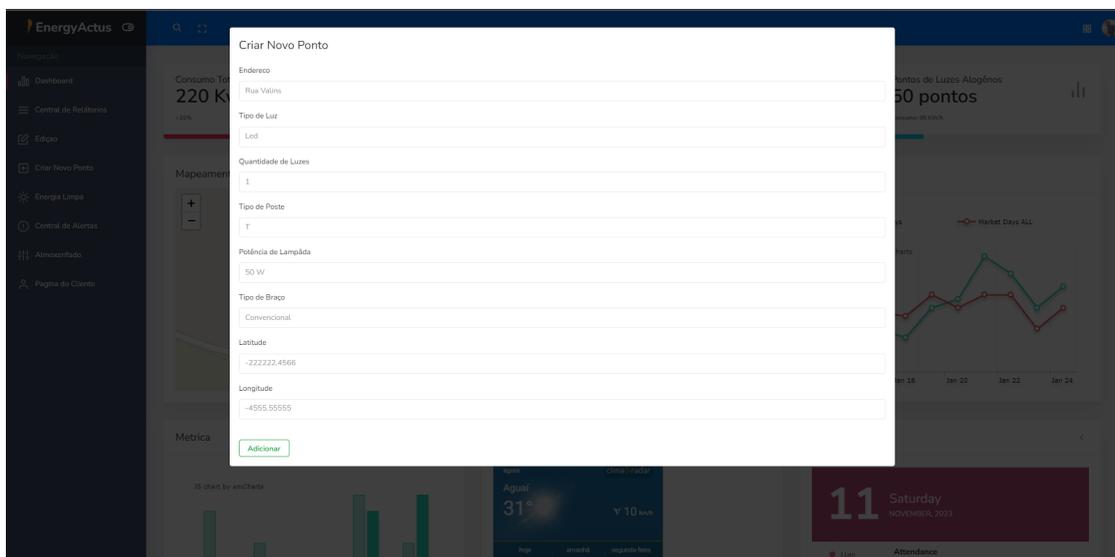
Algumas das razões para se usar PHP e MySQL são: PHP nasceu para a Internet, sua compatibilidade com servidores web é simples e a curva de aprendizado é suave se comparados com outras linguagens, PHP e MySQL são gratuitos e de fácil acesso encontrar serviços de hospedagem que oferecem PHP e MySQL. Os serviços de hospedagem que oferecem esses serviços são mais baratos do que serviços semelhantes de outras tecnologias, e o MySQL é de simples manuseio e rápido, mesmo com grandes quantidades de dados, de acordo com Bento (2014)

### 3.2.4 CRUD

Utilizamos no sistema o método CRUD que significa: Create, Read, Update e Delete, onde o usuário cadastra novos pontos de iluminação (Create), os pontos de iluminação são enviados a diversas partes do sistema, como referenciamento no mapa, dados de consumo e

que são dados de leituras (Read), além de possuir uma edição (Update) e exclusão de qualquer dado cadastrado (Delete).

As figuras 11 e 12 representa a janela de criação e script de criação consecutivamente:7



The image shows a screenshot of the EnergyActus dashboard with a modal window titled "Criar Novo Ponto" (Create New Point) open. The modal contains the following fields and values:

- Endereço: Rua Valina
- Tipo de Luz: Led
- Quantidade de Luzes: 1
- Tipo de Poste: T
- Potência de Lâmpada: 50 W
- Tipo de Braço: Convencional
- Latitude: -22222.4566
- Longitude: -4555.5555

At the bottom of the modal is a green button labeled "Adicionar". The background dashboard shows various charts and data, including a "Consumo Total" of 220 K, a "Mapa" section, and a "Métrica" section with a bar chart. A weather widget shows 31°C and a date widget shows "11 Saturday NOVEMBER, 2023".

**Figura 11 - Janela de criação de novos pontos.**

```

1 <?php
2 require_once '../config/db.php';
3 class Entrada {
4     private $idproduto;
5     private $qtd;
6     private $data;
7     public function setIdproduto($idproduto) {
8         $this->idproduto = $idproduto;
9     }
10    public function __getIdproduto(){
11        return $this->idproduto;
12    }
13    public function setQtd($qtd) {
14        $this->qtd = $qtd;
15    }
16    public function __getQtd(){
17        return $this->qtd;
18    }
19    public function setData($data) {
20        $this->data = $data;
21    }
22    public function __getData(){
23        return $this->data;
24    }
25    public function insert($idproduto, $qtd, $data) {
26        $sql = " INSERT INTO entrada_material (id_produto, qtde, data_entrada )
27        VALUES ('.$idproduto.', '$qtd.', '$data.' ) ";
28        $db = new db();
29        $conn = $db->conectar();
30        if ($conn->query($sql) === TRUE) {
31            $id = $conn->insert_id;
32            $conn->close();
33            header("Location: ../views/almoixerifado/viewsentrada.php?id=". $id);
34        }else{
35            echo "Error: " . $sql . "<br>" . $conn->error;
36            exit();
37        }
38    }
39 }

```

Figura 12 - Script de criação.

As figuras 13 e 14 representa a janela de edição e script de edição consecutivamente:

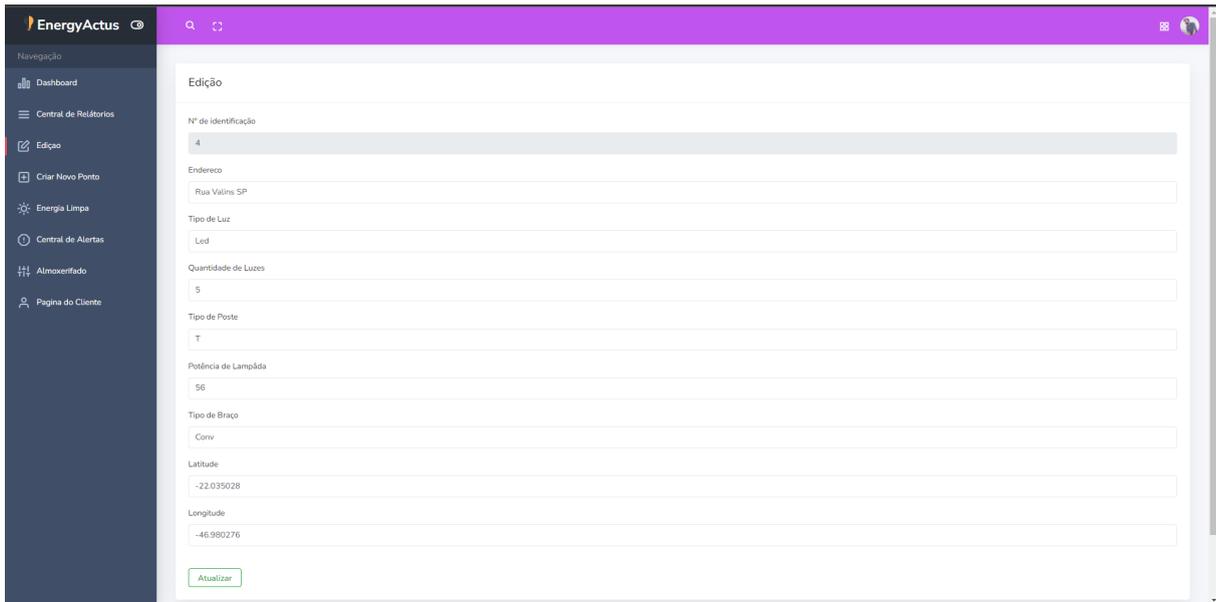


Figura 13 - Janela de edição.

```
1 <?php
2
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $dbname = "projetopi4";
7
8 // Create connection
9 $conn = new mysqli($servername, $username, $password, $dbname);
10
11 // Check connection
12 if ($conn->connect_error) {
13     die("Connection failed: " . $conn->connect_error);
14 }
15
16 $idproduto = filter_input(INPUT_POST, 'idproduto', FILTER_SANITIZE_NUMBER_INT);
17 $descricao = filter_input(INPUT_POST, 'descricao');
18 $qtd = filter_input(INPUT_POST, 'qtd');
19 $data = filter_input(INPUT_POST, 'data');
20 $unidade_medida = filter_input(INPUT_POST, 'unidade_medida');
21
22 // Prepare the SQL statement
23 $sql = "UPDATE produto SET descricao=?, qtd=?, data=?, unidade_medida=? WHERE idproduto=?";
24
25 $stmt = $conn->prepare($sql);
26
27 if ($stmt === false) {
28     die("Error in the prepared statement: " . $conn->error);
29 }
30
31 // Bind parameters and execute the statement
32 $stmt->bind_param("sssd", $descricao, $qtd, $data, $unidade_medida, $idproduto);
33
34 if ($stmt->execute()) {
35     echo "Record updated successfully";
36
37     header('location: almoxeirado.php');
38 } else {
39     echo "Error updating record: " . $stmt->error;
40 }
41
42 $stmt->close();
43 $conn->close();
44 >>
```

Figura 14 - Script de edição.

As figuras 15 e 16 representa a aba de leitura, onde nele possui os ícones edição e exclusão e em seguida o script de deletar:

Edição/Exclusão									
Nº de identificação	Localidade	Tipo de Luz	Quantidade de Luzes	Tipo de Poste	Potência de Lâmpada	Tipo de Braço	Latitude	Longitude	Edição/Exclusão
4	Rua Valins SP	Led	5	T	56	Conv	-22.035028	-46.980276	 

Figura 15 - Janela de leitura de edição e exclusão.

```
views > almoixerifado > exclusao.php > ...
1  <?php
2
3  require_once '../config/db.php';
4
5
6  $db = new db();
7
8  $conexao = $db->conectar();
9
10 $idproduto = $_GET['idproduto'];
11
12 $sql = " DELETE FROM produto WHERE idproduto=".$idproduto;
13
14 print_r( $_GET["idproduto"]);
15
16 $result = $conexao->query($sql);
17
18
19 header('location: almoixerifado.php')
20 ?>
```

Figura 16 - Script de deletar.

```
229 <div class="table-responsive">
230 <table class="table table-hover mb-0">
231 <thead>
232 <tr>
233 <th>Codigo de Produto</th>
234 <th>Descrição</th>
235 <th>Quantidade</th>
236 <th>Data de Lançamento</th>
237 <th>Edição/Exclusão</th>
238 </tr>
239 </thead>
240 <tbody>
241
242 <?php
243
244 require_once '../config/db.php';
245 $db = new db();
246
247 $conexao = $db->conectar();
248
249 $sql = "SELECT * from produto";
250 $result = $conexao->query($sql);
251
252 if ($result->num_rows > 0) {
253 // output data of each row
254 while($row = $result->fetch_assoc()) {
255 //print_r($row);
256
257 echo'
258 <tr>
259 <td>
260 <div class="d-inline-block align-middle">
261 <div class="d-inline-block">
262 <div class="d-inline-block">
263 <div class="d-inline-block">
264 <div class="d-inline-block">
265 </div>
266 </td>
267 <td>
268 <input type="hidden" name="idproduto" idpostes=".$row['idproduto'].'" placeholder="Username" value=".$row['idproduto'].'" readonly >
269 <td>.$row['descricao'].</td>
270 <td>.$row['qtd'].<td>.$row['unidade_medida'].</td>
271 <td>.$row['data'].</td>
272 <td><a href="telaedicao.php?idproduto=".$row['idproduto'].'">i class="ik ik-edit f-16 mr-15 text-green"></i><a href="exclusao.php?idp
273
274
275
276 ?>
```

Figura 17 - Leitura de Dados.

## **3.3 COMPUTAÇÃO EM NUVEM**

### **3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING**

O objetivo do projeto de cloud computing é criar e manter um ambiente de armazenamento em nuvem onde seja possível armazenar todas informações necessárias para o funcionamento do sistema na internet, tornando possível o acesso a essas informações e arquivos a partir de uma internet pública ou rede privada.

### **3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO**

A computação em nuvem, permite permanecer uma conexão direta entre o servidor e o banco de dados, permitindo que o projeto funcione corretamente, puxando e entregando todos dados que serão utilizados de forma simples e rápida.

### **3.3.3 VANTAGENS DA CLOUD COMPUTING**

A computação em nuvem permite uma maior facilidade de utilização, uma vez que o usuário não necessita de um servidor ou máquina física, o banco de dados pode ser utilizado a qualquer momento, além de ser mais flexível, podendo aumentar ou diminuir sua capacidade de armazenamento baseado no quão será utilizado no momento. Como exemplo disto se possuir um grande fluxo de usuários ou informações, o banco pode se expandir momentaneamente para que seja possível o funcionamento sem nenhum problema de latência, e caso não esteja em alta demanda, o banco pode diminuir seu armazenamento, gerando uma maior economia e maior velocidade nas consultas.

### **3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING**

No projeto, utilizaremos o modelo Software as a Service (SaaS), pois ele elimina a necessidade de instalação local, proporcionando facilidade de acesso e atualizações automáticas. Além disso, o balanceamento de carga é essencial para melhorar o desempenho, tendo um papel crucial na entrega consistente das aplicações na nuvem, evitando sobrecargas e garantindo uma utilização eficiente dos recursos.

Os principais componentes incluem data center virtualizado, rede em nuvem, serviços de armazenamento, máquinas virtuais, APIs, segurança em nuvem, serviços de

balanceamento de carga e monitoramento automatizado, esses elementos garantem flexibilidade, segurança e eficiência operacional no projeto da empresa.

Em um projeto que se utiliza cloud computing, a escolha entre modelos de aplicação (SaaS, PaaS, IaaS) afeta a arquitetura. O balanceamento de carga otimiza o desempenho, enquanto componentes como rede, máquinas virtuais e armazenamento garantem escalabilidade. APIs facilitam a integração, e a segurança em nuvem é crucial. O monitoramento automatizado melhora a eficiência. A integração eficiente desses elementos atende às necessidades da empresa, garantindo o sucesso do projeto em cloud.

### **3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)**

Após analisar todos requisitos, a AWS foi escolhida como provedor de cloud computing, visto que é um serviço muito grande e reconhecido do mercado, e possui diversas vantagens quando comparado ao seus concorrentes.

O serviço AWS é um dos melhores em quesito variedade de serviços, facilitando atender corretamente as expectativas e necessidades do cliente. Além de ter uma ótima capacidade de expansão e escalabilidade, permitindo que um sistema possa começar e caso evolua, seu banco de dados irá se adaptar também.

O plano de armazenamento e serviço da AWS são totalmente moldável às necessidades do sistema, sendo assim, é cobrado apenas o que se utiliza, permitindo uma maior economia em momentos de baixo fluxo, e uma maior performance em momentos de expansão.

### **3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING**

O balanceamento de carga na nuvem é vital para otimizar recursos, garantir disponibilidade e manter o desempenho equilibrado. Ele distribui a carga de trabalho entre servidores, monitora constantemente o desempenho, toma decisões dinâmicas e promove redundância para aumentar a disponibilidade.

- **Anatomia de Cloud Computing:**

IaaS (Infraestrutura como Serviço): Oferece recursos virtuais.

PaaS (Plataforma como Serviço): Fornece uma plataforma completa para desenvolvimento.

SaaS (Software como Serviço): Aplicações hospedadas na nuvem acessadas via navegador.

Nuvens Públicas, Privadas e Híbridas: Variando em compartilhamento e dedicação de recursos.

- **Paradigmas Tecnológicos Subjacentes:**

Virtualização: Criação de máquinas virtuais para isolar aplicativos.

Escalabilidade Horizontal e Vertical: Aumento de instâncias ou desempenho conforme demanda.

Automação e Orquestração: Simplificação e coordenação de processos.

Serviços Web e APIs: Facilitam a comunicação entre componentes distribuídos.

### **3.3.7 GOOGLE CLOUD ou AWS**

O serviço escolhido foi o da AWS, pois é um dos melhores em quesito variedade de serviços, facilitando atender corretamente as expectativas e necessidades do cliente. Além de ter uma ótima capacidade de expansão e escalabilidade, permitindo que um sistema possa começar e caso evolua, seu banco de dados irá se adaptar também. Tendo isso em vista, vamos utilizar o serviço de nuvem da AWS para hospedar nosso site para que ele seja acessível de qualquer lugar e a qualquer hora do dia.

comparativo de custo

<https://calculator.aws/#/addService>

## **3.4 ESTRUTURA DE DADOS**

O módulo de Estruturas de Dados auxiliar durante o processo de planejamento e execução do projeto, onde o mesmo ensina os métodos adotados durante a definição dos requisitos de sistema, além de guiar a equipe durante a execução do projeto, como na escolha da linguagem a ser adotada, os métodos e estruturas que serão incrementados, além da junção de todos esses fatores em um software funcional e simples para utilização.

### **3.4.1 LEVANTAMENTO DE REQUISITOS**

O processo de levantamento de requisitos pode ser classificado como as funções que o sistema deve realizar, e quais as restrições que os mesmo tem, a concepção do levantamento

de requisitos contém duas etapas, o levantamento de requisitos funcionais que classificam as funcionalidades presentes na elaboração do software, Já os requisitos não funcionais, definem algo técnico do sistemas, são os requisitos ligados ao uso da aplicação podendo ser relacionada a desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologia envolvidas que na qual seria, "Como o sistema irá ser construído?".

### 3.4.2 VALIDAÇÃO DOS REQUISITOS

Durante o processo inicial do projeto onde foi realizado o planejamento, dividido funções e determinando entregas, foram levantados os seguintes requisitos funcionais e não funcionais.

<b>RF01</b>	
Função	Realizar o Gerenciamento de iluminação pública.
Descrição	O sistema deve ter uma tela onde seja possível acompanhar o status de cada um dos pontos.
Entradas	-
Origem	-
Saídas	Exibição de status dos pontos
Destinos	Banco de dados do sistema
Ação	Receber status do ponto de luz
Requer	Sincronização dos pontos com mapas para visualização
Precondição	-
Pós-condição	-
Efeitos	-

<b>RF02</b>	
Função	Apresentar Dashboard com dados sobre os postes.
Descrição	O sistema deve ter uma dashboard onde fornece dados sobre os postes de energia.
Entradas	Informações coletadas pelos postes
Origem	Informações coletadas por sensores
Saídas	Dashboards
Destinos	Banco de dados do sistema
Ação	Preencher dashboards
Requer	Sincronização dos sensores com banco de dados
Precondição	-
Pós-condição	-
Efeitos	-

<b>RF03</b>	
Função	Possuir CRUD de cadastros dos postes
Descrição	O sistema deve ter um CRUD de cadastros dos postes bem como informações/observações
Entradas	Cadastros
Origem	Informações fornecidas pelo operante
Saídas	-
Destinos	Banco de dados do sistema
Ação	Permitir inclusão, update e exclusão de cadastros
Requer	CRUD bem organizado
Precondição	-
Pós-condição	-
Efeitos	Maior facilidade durante o registro de pontos

<b>RF04</b>	
Função	Possuir auto sustentabilidade
Descrição	O sistema deve conseguir se gerenciar automaticamente sem a necessidade de um técnico sempre o manuseando.
Entradas	-
Origem	-
Saídas	-
Destinos	-
Ação	Se auto gerenciar.
Requer	Integração de APIs e funcionalidades.
Precondição	-
Pós-condição	-
Efeitos	Maior facilidade para o operador do sistema.

<b>RF05</b>	
Função	Ativação dos postes em um horário exato
Descrição	O sistema deve ser capaz de iniciar todos os pontos de energia em um determinado horário.
Entradas	Horário de ativação.
Origem	Horário estabelecido entre fornecedor de energia e equipe desenvolvedora.
Saídas	Ativação dos pontos de energia.
Destinos	-
Ação	Ativar todos os pontos de energia.
Requer	Horário pré determinado.
Precondição	Possui um horário definido para ativação.
Pós-condição	-
Efeitos	Maior facilidade durante o registro de pontos.

<b>RF06</b>	
Função	Dashboard que exibe os gastos de energia
Descrição	Apresentar as informações como gastos de energia em uma dashboard de fácil compreensão.
Entradas	Gastos de energia.
Origem	Sensores de captação presente nos postes.
Saídas	Dados que serão apresentados.
Destinos	Banco de dados do projeto.
Ação	Apresentar dados.
Requer	Dados para serem apresentados.
Precondição	-
Pós-condição	-
Efeitos	Informações mais claras e melhor visualização das vantagens do sistema.

<b>RF07</b>	
Função	Apresentar bom funcionamento, para os clientes poderem realizar chamados
Descrição	Ser simples e intuitivo para que a comunidade possa realizar a abertura de chamados.
Entradas	Chamados.
Origem	Chamados abertos pela comunidade.
Saídas	Chamados para que os técnicos possam atender.
Destinos	Banco de dados do sistema
Ação	Permitir a criação de chamados pela comunidade.
Requer	Simplicidade e usabilidade.
Precondição	-
Pós-condição	-
Efeitos	Maior facilidade para uso da comunidade.

<b>RF08</b>	
Função	Exibir um acompanhamento em tempo real dos pontos de energia
Descrição	Possui uma tela que apresenta o status atual dos pontos de energia.
Entradas	Status dos pontos de energia.
Origem	Pontos de energia.
Saídas	Apresentar isso em uma dashboard.
Destinos	Dashboard.
Ação	Apresentar o status dos pontos de energia.
Requer	API que permita esse controle em tempo real.
Precondição	-
Pós-condição	-
Efeitos	Maior controle dos pontos de energia.

<b>RF09</b>	
Função	Possuir um mapa que apresente os pontos de energia.
Descrição	Possuir um mapa real que mostre os pontos de energia.
Entradas	Mapa e APIs.
Origem	Mapas a partir de APIs.
Saídas	Apresentar esse mapa nas telas com os pontos de energia.
Destinos	Dashboard.
Ação	Apresentar esse mapa nas telas com os pontos de energia.
Requer	APIs e mapas.
Precondição	-
Pós-condição	-
Efeitos	Maior controle dos pontos de energia.

<b>RF10</b>	
Função	Exibir quantidade de energia armazenada
Descrição	Possui uma dashboard com a quantidade de energia armazenada.
Entradas	Dados captados pelos pontos de energia.
Origem	Sensores presentes nos pontos de energia.
Saídas	Quantidade de energia armazenada.
Destinos	Banco de dados do sistema
Ação	Apresentar a quantidade de energia armazenada.
Requer	Sensores que captam e armazenam a energia solar.
Precondição	-
Pós-condição	-
Efeitos	Informações mais claras e melhor gerenciamento.

<b>RF11</b>	
Função	Possuir uma central de relatórios
Descrição	Possui tela própria apenas para relatórios.
Entradas	Dados.
Origem	Sensores de pontos de energia.
Saídas	Relatórios.
Destinos	Banco de dados do sistema e dashboards.
Ação	Ter uma tela própria somente para relatórios.
Requer	Dados.
Precondição	-
Pós-condição	-
Efeitos	Maior controle e melhor acompanhamento do sistema e suas vantagens.

<b>RF12</b>	
Função	Permitir que o usuário salve relatórios em PDF
Descrição	Salvar relatórios em forma de PDF para que possam ser salvos localmente.
Entradas	Relatórios.
Origem	Sistema.
Saídas	Relatórios em PDF.
Destinos	Software.
Ação	Permitir salvar relatórios em PDF.
Requer	Relatórios gerados pelo sistema.
Precondição	-
Pós-condição	-
Efeitos	Salvar relatórios em PDF.

<b>RF13</b>	
Função	Permitir acesso rápido para todas as funções
Descrição	Ser simples e ágil.
Entradas	Dados limpos e simples.
Origem	Software.
Saídas	Sistema simples e ágil.
Destinos	Software.
Ação	Ser simples e ágil.
Requer	Dados limpos e banco de dados organizado para não gerar lentidão ou travamento no sistema.
Precondição	-
Pós-condição	-
Efeitos	Melhor facilidade de uso.

<b>RF14</b>	
Função	Apresentar gráficos.
Descrição	Ter dashboards com gráficos.
Entradas	Dados.
Origem	Software.
Saídas	Gráficos.
Destinos	Dashboards.
Ação	Gerar gráficos para melhor visualização das informações.
Requer	Dados para gerar gráficos.
Precondição	-
Pós-condição	-
Efeitos	Mais clareza nas informações.

<b>RF15</b>	
Função	Possuir dashboards nítidas e bem divididas
Descrição	Telas bem organizadas e claras.
Entradas	-
Origem	Linguagem e organização adotadas.
Saídas	Telas bem organizadas de fácil compreensão.
Destinos	Dashboards.
Ação	Apresentar telas claras e bem divididas.
Requer	Bom planejamento de telas.
Precondição	-
Pós-condição	-
Efeitos	Maior facilidade para os usuários.

<b>RF16</b>	
Função	Possuir uma tela para fornecer suporte.
Descrição	Ter uma tela de suporte para atender os chamados abertos.
Entradas	-
Origem	-
Saídas	Tela para suporte de chamados.
Destinos	Software.
Ação	Ter uma tela para atendimento de chamados.
Requer	Tela bem organizada para que o técnico possa interagir e atender os chamados.
Precondição	-
Pós-condição	-
Efeitos	Maior agilidade nos atendimentos.

<b>RF17</b>	
Função	Possuir um usuário local
Descrição	O sistema deve possuir um usuário local para uso dos técnicos e responsáveis pelo sistema.
Entradas	Usuário cadastrado.
Origem	Sistema.
Saídas	Usuário local.
Destinos	Usuário no sistema.
Ação	Possuir um usuário local no sistema.
Requer	O sistema requer usuário local.
Precondição	-
Pós-condição	-
Efeitos	Maior velocidade no sistema pois o sistema não deve buscar por diversos usuários no sistema.

<b>RF18</b>	
Função	Não permitir cadastro de novos usuários
Descrição	O sistema deve permitir apenas os usuários cadastrados durante o desenvolvimento.
Entradas	-
Origem	-
Saídas	-
Destinos	Banco de dados do sistema
Ação	Não permitir o cadastro de usuários.
Requer	Bloqueio para criação de usuários.
Precondição	-
Pós-condição	-
Efeitos	O sistema deve possuir apenas usuário local para gerenciamento do mesmo, a comunidade não precisa de usuário para abertura de chamados.

<b>RF19</b>	
Função	Possui um bom funcionamento, simples e autoexplicativo.
Descrição	O sistema deve ser simples e de fácil uso.
Entradas	-
Origem	-
Saídas	-
Destinos	-
Ação	Simplificar o uso no sistema.
Requer	Bom planejamento do sistema para criação das telas.
Precondição	-
Pós-condição	-
Efeitos	Maior facilidade de uso.

<b>RF20</b>	
Função	Permitir a gestão da energia armazenada
Descrição	O sistema deve permitir que o técnico gereencie a energia armazenada.
Entradas	Energia armazenada.
Origem	Sensores e baterias.
Saídas	Possibilidade de gerenciamento dessa energia.
Destinos	Software.
Ação	Permitir gerenciamento de energia.
Requer	Sistema de controle da energia.
Precondição	-
Pós-condição	-
Efeitos	Controle de energia possibilitando como exemplo aumentar a quantidade de iluminação em um determinado ponto.

<b>RNF1</b>	
Função	Possuir verificação em duas etapas.
Descrição	Possuir verificação de dois fatores para entrada no sistema.
Entradas	Verificação de duas etapas.
Origem	-
Saídas	-
Destinos	Banco de dados do sistema
Ação	Realizar dupla autenticação ao entrar no sistema.
Requer	Bom gerenciamento de segurança.
Precondição	-
Pós-condição	-
Efeitos	Mitigação de riscos evitando invasões no sistema.

<b>RNF2</b>	
Função	Ser construído em plataforma web.
Descrição	O sistema deve ser desenvolvido em uma plataforma WEB.
Entradas	-
Origem	-
Saídas	-
Destinos	Software.
Ação	Desenvolver o sistema para WEB.
Requer	Conhecimento de linguagens WEB.
Precondição	-
Pós-condição	-
Efeitos	Sistema mais simples de ser desenvolvido e hospedado.

<b>RNF3</b>	
Função	Possuir tempo de resposta ágil
Descrição	O sistema deve possuir uma resposta ágil.
Entradas	-
Origem	-
Saídas	-
Destinos	Banco de dados do sistema.
Ação	Ter um bom tempo de resposta.
Requer	Banco de dados bem desenhado e boa virtualização para não haver lentidão ou gargalos no sistema.
Precondição	-
Pós-condição	-
Efeitos	Melhor experiência durante o uso.

<b>RNF4</b>	
Função	Possuir responsividade
Descrição	O sistema deve ser responsivo.
Entradas	-
Origem	-
Saídas	-
Destinos	Software.
Ação	Permitir que o sistema seja utilizado em diversos aparelhos.
Requer	Desenvolvimento de sistema responsivo.
Precondição	-
Pós-condição	-
Efeitos	Possibilidade de utilizar o sistema em qualquer dispositivo.

<b>RNF5</b>	
Função	Possuir um banco de dados MySql
Descrição	O banco de dados deve ser desenvolvido em MySql
Entradas	-
Origem	-
Saídas	-
Destinos	Banco de dados do sistema
Ação	Desenvolver o banco de dados em MySql
Requer	Conhecimento de MySql para desenvolvimento.
Precondição	-
Pós-condição	-
Efeitos	Maior facilidade de uso do sistema, atualizações e consultas mais rápidas.

<b>RNF6</b>	
Função	Permitir a utilização de frameworks para design.
Descrição	Permitir a utilização de frameworks para design do sistema.
Entradas	-
Origem	-
Saídas	-
Destinos	Software.
Ação	O sistema deve permitir a utilização de frameworks.
Requer	Bom planejamento para uso dos frameworks.
Precondição	-
Pós-condição	-
Efeitos	Sistema mais claro e simples.

<b>RNF7</b>	
Função	Permitir o uso de APIs
Descrição	Durante o desenvolvimento o sistema deve permitir o uso de APIs.
Entradas	APIs.
Origem	-
Saídas	Novas funcionalidades do sistema.
Destinos	Software.
Ação	Permitir inclusão APIs.
Requer	Possibilidade de integração de APIs.
Precondição	-
Pós-condição	-
Efeitos	Por se tratar de um sistema que lida com mapas reais, e com diversas funcionalidades, o mesmo deve possuir a possibilidade de integração de APIs.

<b>RNF8</b>	
Função	Apresentar boa segurança do sistema
Descrição	O sistema deve ser seguro por se tratar de um gerenciamento público.
Entradas	Métodos de segurança.
Origem	-
Saídas	Maior segurança ao sistema.
Destinos	Software.
Ação	Tornar o sistema mais seguro.
Requer	Métodos pensados de segurança.
Precondição	-
Pós-condição	-
Efeitos	Maior segurança e tranquilidade para os utilizadores do sistema.

<b>RNF9</b>	
Função	Constar a situação dos campos de preenchimento
Descrição	O deve informar ao usuário a situação dos campos de preenchimento.
Entradas	APIs.
Origem	-
Saídas	Informar quando algo não está sendo preenchido corretamente.
Destinos	Sistema.
Ação	Informar caso haja algum erro de preenchimento.
Requer	Uso de APIs.
Precondição	-
Pós-condição	-
Efeitos	Maior facilidade para uso da comunidade durante o preenchimento de chamados.

## **3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS**

A Formação para a Vida é um dos eixos do Projeto Pedagógico de Formação por Competências da UNIFEOB.

Esta parte do projeto está diretamente relacionada com a extensão universitária, ou seja, o objetivo é que seja aplicável e que tenha real utilidade para a sociedade, de um modo geral.

### **3.5.1 ENFRENTANDO ESTEREÓTIPOS**

Está disponível para os estudantes no Classroom, o tema “Enfrentando Estereótipos”.

Nesta parte do Projeto, os estudantes deverão realizar uma síntese dos 4 (quatro) tópicos deste tema, quais sejam:

- **Tópico 1:** Estereótipo e convívio social
  - Estereótipos moldam dinâmicas sociais em vários contextos, desde o local de trabalho até interações diárias. No convívio social, esses estereótipos podem influenciar atitudes e comportamentos de diversas maneiras. Conscientizar-se dessas influências é crucial para promover um convívio social mais inclusivo e equitativo, destacando a importância da desconstrução de estereótipos por meio da educação e do respeito à diversidade. Como exemplo, a ideia de que mulheres são menos competentes em cargos de liderança pode influenciar decisões de promoção, prejudicando a igualdade de oportunidades.
- **Tópico 2:** Estereótipo e representação
  - Estereótipos são simplificações generalizadas sobre grupos de pessoas, enquanto representações referem-se à forma como esses grupos são retratados na sociedade, muitas vezes através da mídia. Ambos desempenham papéis significativos no modo como percebemos e interagimos com diferentes comunidades. Um exemplo é a representação constante de certos grupos étnicos como vilões pode levar a percepções distorcidas e discriminação no convívio social.

- **Tópico 3:** Troco likes: a idealização da vida na internet
  - A expressão "Troco Likes" representa a busca por validação e reconhecimento nas redes sociais, muitas vezes levando à idealização de vidas online. Isso ocorre quando as pessoas compartilham seletivamente aspectos positivos de suas vidas, criando uma imagem distorcida da realidade. A exibição de viagens e experiências luxuosas pode criar a ilusão de que a vida de alguém é constantemente emocionante e próspera. Porém essas representações nem sempre refletem a realidade financeira ou emocional.
- **Tópico 4:** Convivendo com a diferença
  - Conviver com a diferença envolve aceitar e respeitar a diversidade em todos os aspectos da vida. Isso requer uma abertura para compreender perspectivas diversas e valorizar as experiências únicas de cada um. Programas que envolvem jovens e idosos em atividades colaborativas, como mentorias intergeracionais, são exemplos de convivência positiva com a diferença.

A síntese precisa apresentar exemplos práticos dos seus conteúdos, ou seja, de modo que possam ser utilizados ou verificados no dia-a-dia.

### 3.5.2 ESTUDANTES NA PRÁTICA

Após realizar a síntese dos conteúdos, é hora dos estudantes colocarem a “mão na massa”, de produzirem algo da formação para a vida que possa ser utilizado na prática pela sociedade.

A equipe deve elaborar um material como um podcast, um banner, um pequeno vídeo (sugere-se este em razão da facilidade de elaboração e divulgação) comentando sobre o estereótipo das pessoas do interior, que muitas vezes são chamadas de “caipiras”.

O objetivo desse material é contribuir para as pessoas refletirem as questões culturais do Brasil, relacionadas a cada uma de suas regiões.

Cada região tem as suas particularidades onde a identidade de seus moradores acaba, muitas vezes, sendo generalizada por quem não convive diretamente naquela realidade, o que acaba gerando estereótipos atribuídos precocemente a alguns grupos.

As pessoas que moram no interior, por serem mais próximas do agronegócio, podem ter sua conduta rotulada de forma de forma errônea por aqueles que não vivem naquela região, por exemplo, na televisão, os filmes e novelas acabam relacionando pessoas do interior à pessoas que vivem no campo, quando pode haver uma grande diferença entre elas.

Você já ouviu o termo “agroboy”? Então, essa atividade pode ajudar as pessoas a entenderem melhor as questões que envolvem a definição deste estereótipo.

Caso a equipe escolha pelo vídeo, poderá ser gravado de forma bem simples e ser disponibilizado em algum canal do Youtube de seus integrantes, como “Não Listado”. Se a equipe se sentir à vontade, também pode compartilhar esse vídeo nas redes sociais, por exemplo, no Instagram e marcar a Escola de Negócios usando @ednunifeob.

O objetivo é que todos os integrantes da equipe participem desse material e que possam compartilhá-lo para que as pessoas da comunidade onde estejam inseridos possam se identificar melhor com suas origens e se sentir orgulhosas por pertencerem à uma determinada região do país.

Portanto, neste tópico do PI, a equipe deve elaborar um pequeno texto descrevendo o conteúdo desse material e, em seguida, colocar o link público do arquivo para que possa ser verificado e avaliado.

A divulgação e compartilhamento desse material para a comunidade externa será de responsabilidade dos próprios estudantes, conforme o compromisso social de cada pessoa, pois a UNIFEOB apenas usará esse material para avaliação desta atividade.

**OBSERVAÇÃO:** A realização do item 3.5 deste projeto é uma atividade que integra parte do conteúdo da unidade de Formação para a Vida com o Projeto Integrado, portanto, **não exclui a obrigatoriedade** do estudante de realizar os desafios e demais atividades disponibilizados para a unidade de estudo de Formação para a Vida.

## 4. CONCLUSÃO

O projeto apresentado neste documento tem como objetivo a idealização e criação de um software que permita gerenciar um sistema de energia público funcionando a partir da captação de energia solar, atendendo assim objetivo onze dos Dezessete Objetivos da ONU - Desenvolvimento Sustentável, onde o objetivo onze busca mais formas de se utilizar energia sustentável nas cidades e comunidades. A partir do software desenvolvido e documentado, apresentamos uma solução funcional e que atenderia as necessidades do mercado, este software tem como alvo empresas responsáveis pelo gerenciamento público de energia, e o mesmo permite o controle de pontos de energia (postes) que funcionam a partir da captação de energia solar durante o dia, armazenando esta energia e liberando do-a em forma de iluminação durante a noite.

O software desenvolvido de forma WEB permite o acompanhamento destes pontos de energia podendo apresentar informações como a quantidade de energia armazenada, o a quantidade de economia que os pontos geraram, mapas reais com todos os pontos de energia e seu status, além de um sistema de chamados que permite que a comunidade relatar problemas em sua área, e que os técnicos possam realizar interações informando o status do reparo e mantendo a comunidade sempre atualizada em relação aos problemas em atendimento.

Durante o desenvolvimento foram encontradas algumas dificuldades na utilização de novas linguagens por se tratar de algo novo para equipe, tornando necessário uma série de pesquisas para melhor compreensão, além deste ponto também por se tratar de um sistema que lida com mapas reais e pontos reais, foi necessário a integração de várias APIs não somente para o funcionamento do sistema mas também para melhor utilização por parte do usuário.

Desta forma concluímos que o projeto apresentado é funcional e pode ajudar muito no desenvolvimento de comunidades mais sustentáveis, com um menor consumo de energia, um consumo mais limpo e um melhor gerenciamento por parte dos provedores.

## REFERÊNCIAS

Redação Cronapp. 10 frameworks que tornam mais rápido o desenvolvimento de softwares. Disponível em: <<https://blog.cronapp.io/frameworks-para-desenvolvimento-de-softwares/>>.

Amazon AWS. 10 O que é o armazenamento em nuvem?. Disponível em: <[https://aws.amazon.com/pt/what-is/cloud-storage/#:~:text=O%20armazenamento%20em%20nuvem%20é%20um%20modelo%20de%20computação%20em,conexão%20de%20rede%20privada%20dedicada./](https://aws.amazon.com/pt/what-is/cloud-storage/#:~:text=O%20armazenamento%20em%20nuvem%20é%20um%20modelo%20de%20computação%20em,conexão%20de%20rede%20privada%20dedicada./>)>.

DEVMedia. Implementando controle de estoque no MySQL com triggers e procedures. Disponível em: <<https://www.devmedia.com.br/implementando-controle-de-estoque-no-mysql-com-triggers-e-procedures/26352>>.

MAGALHÃES, Tulio. Descubra finalmente o que é o cloud computing e para que serve a computação em nuvem. Disponível em: <<https://rockcontent.com/br/blog/cloud-computing/>>.

Amazon AWS. O que é a computação em nuvem?. Disponível em: <<https://aws.amazon.com/pt/what-is-cloud-computing/>>.

SolarVolt Energia. Energia solar em obras públicas: 4 possibilidades transformadoras. Disponível em: <<https://www.solarvoltenergia.com.br/blog/energia-solar-em-obras-publicas/>>.

PortalSolar. Poste de energia solar: o que é e como funciona?. Disponível em: <<https://www.portalsolar.com.br/poste-energia-solar>>.

alba Energia Solar. QUEM TEM ENERGIA SOLAR PAGA TAXA DE ILUMINAÇÃO PÚBLICA?. Disponível em: <<https://albaenergia.com.br/quem-tem-energia-solar-paga-taxa-de-iluminacao-publica/>>.

Marcon, Antonio Marcos; Neves, Denise. Aplicações E Banco De Dados Para Internet. 2. Ed. São Paulo: Érica, 1999.

W3C ESCRITÓRIO BRASIL. Disponível em: <http://www.w3c.br/>.

FERREIRA, S. Guia Prático de HTML5:. Universo dos Livros, 2011..

GRANNELL, C. (2007). The Essential Guide to CSS and HTML Web Design. New York: APress.

SILVA, M. CSS3: Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. NOVATEC, 2011. ISBN 9788575222898.

Soares, Wallace. Ajax (Asynchronous Javascript And Xml): Guia Prático Para Windows. 2. Ed. São Paulo: Érica, 2006.

Goodman, D. Javascript Bible - Gold Edition. 1ª. ed. New York: Hungry Minds Inc, 2001.

BOOTSTRAP, T. Twitter bootstrap. 2013. Disponível em: <https://getbootstrap.com/>

ANSELMO, Fernando. PHP e MySQL. Florianópolis: Visual Books, 2002.

PHP.NET. História do PHP. Disponível em: [http://php.net/manual/pt\\_BR/history.php.php](http://php.net/manual/pt_BR/history.php.php) .

BENTO, Evaldo Junior. Desenvolvimento web com PHP e MySQL. São Paulo: Casa do Código, 2014.

# ANEXOS

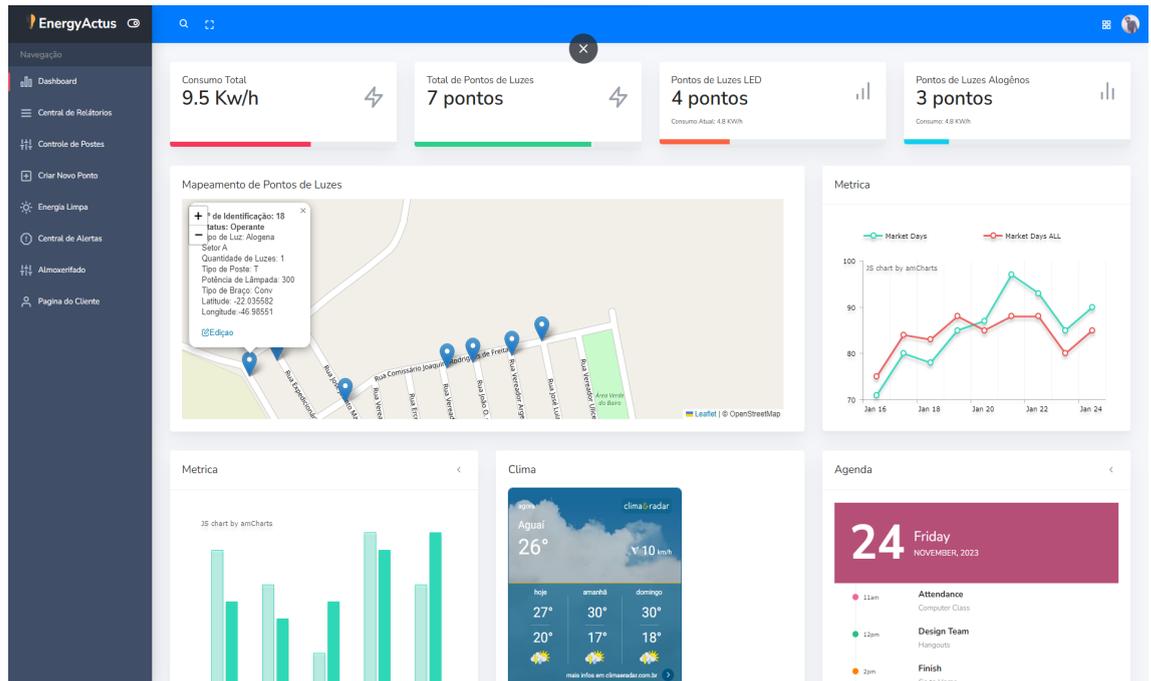
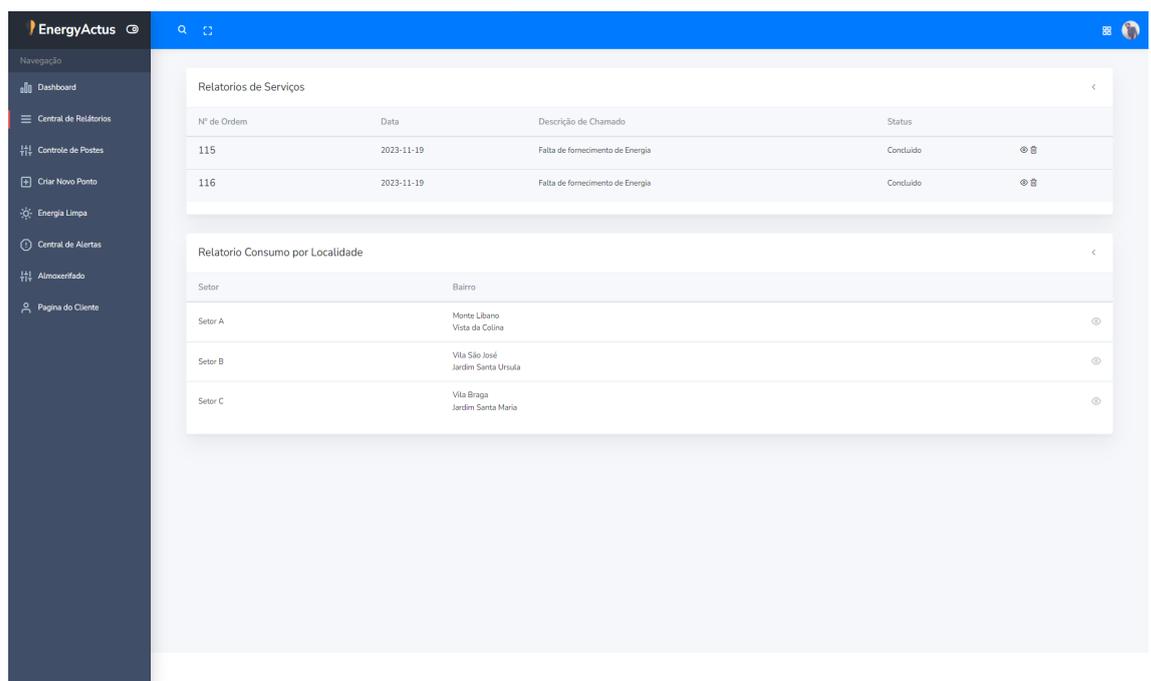


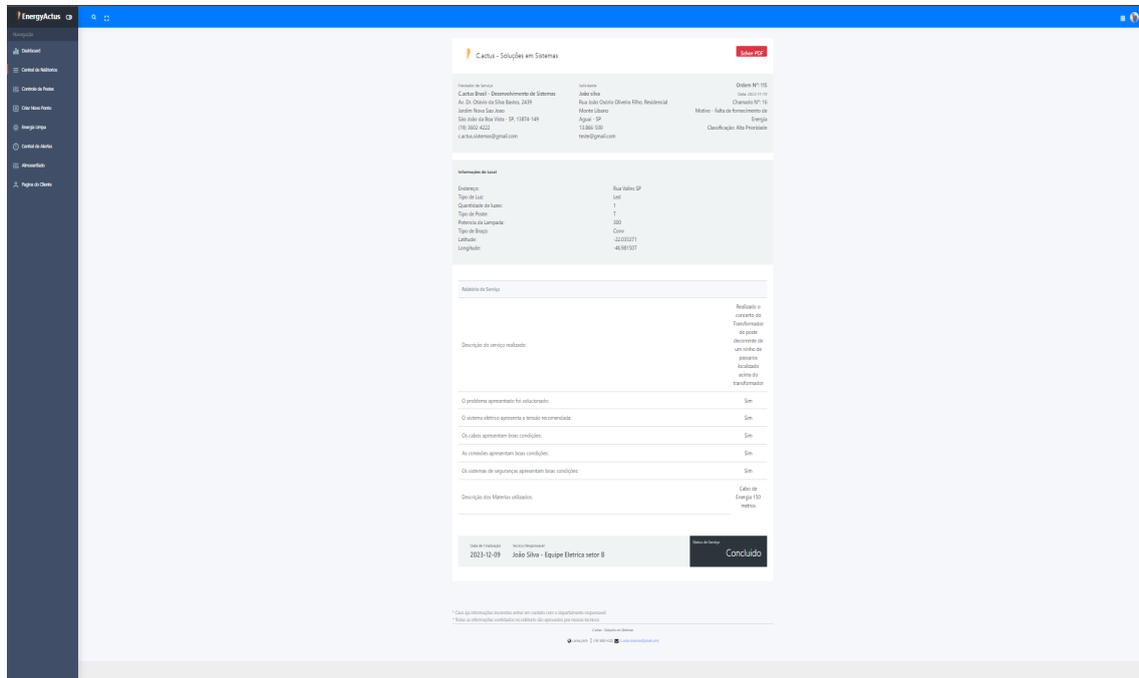
Figura 01 - Dashboard Principal.

Na figura acima, demonstra a dashboard do sistema, nela contemplando o consumo total de energia e a quantidade de ponto de luzes presente na cidade, abaixo representação do geomapeamento dos pontos, seguido de algumas métricas, clima e ordens de serviço em aberto.



## Figura 02 - Central de Relatórios.

Na figura 02, demonstra a central de relatórios onde são classificados por relatórios de serviços e relatórios de consumo por setorização.



## Figura 03 - Visualização de Relatório de Serviço.

Na figura 03, mostra a visualização do relatório serviço, onde demonstra a localização onde foi realizado, os dados de solicitação, o serviço que foi realizado, um checklist de infraestrutura, status de serviço e equipe responsável além disso o usuário tem a opção de salvar no formato PDF

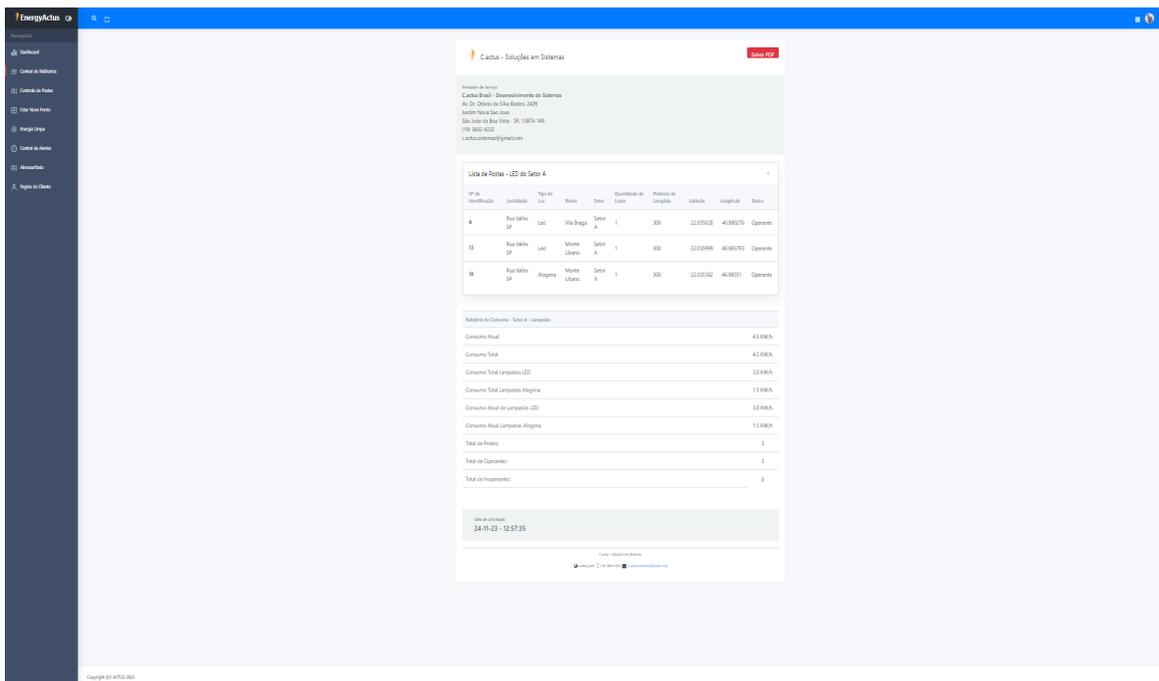


Figura 04 - Visualização de Relatório de Consumo.

Na figura 04, demonstra a o relatório de consumo, onde foi classificado por setor da cidade, o relatório apresenta uma lista de todos os pontos de luz daquele setor, e abaixo apresenta métricas como, consumo total e atual, poste operantes ou inoperantes, consumo individual por tipo de lâmpada.

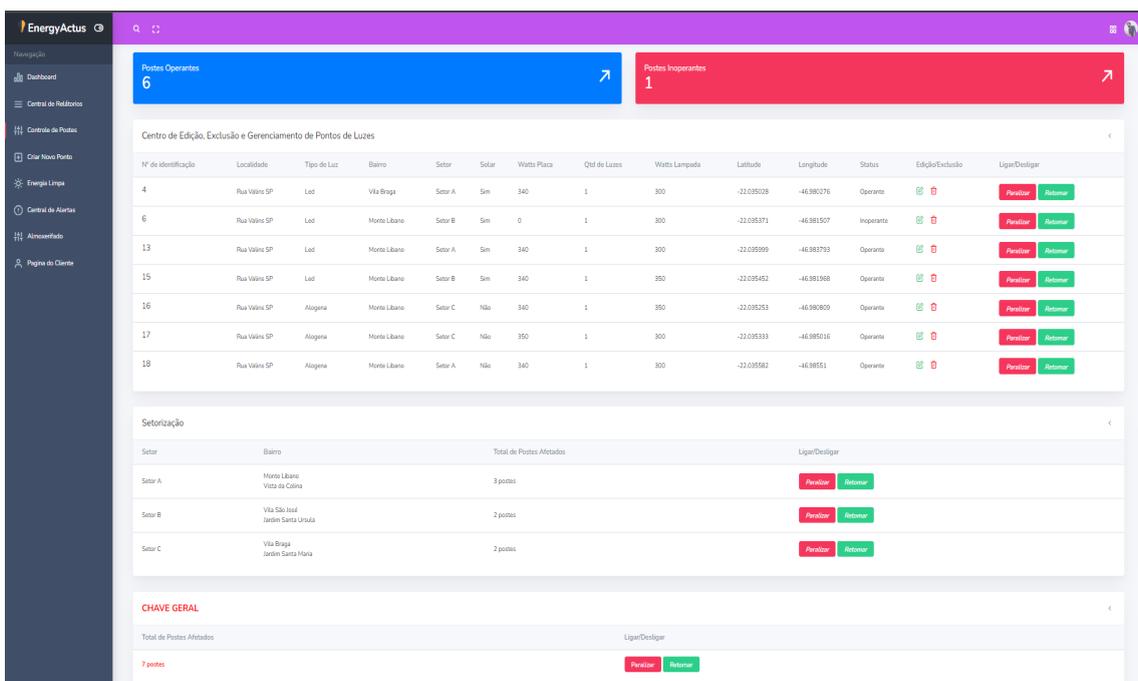


Figura 05 - Central de Controle de Postes(Ligar/Desligar, Editar/Excluir).

Na figura 05, exibe a janela de controle de postes, onde mostra a situação geral de postes operantes e inoperantes, abaixo lista total de todos os postes com opção de edição e exclusão e opções de Paralisar e Retomar a operação do poste.

The screenshot shows the 'Edição' (Edit) form for a light point in the EnergyActus system. The form is titled 'Edição' and contains the following fields:

- Nº de identificação: 4
- Endereço: Rua Valério SP
- Tipo de Luz: Led
- Bairro: Vila Braga
- Sector: Sector A
- Energia Solar?: Sim
- Rendimento da Placa: 0.8
- Potência da Placa (W): 140
- Quantidade de Luzes: 1
- Tipo de Poste: T
- Potência de Lâmpada: 300
- Tipo de Braco: Conv.
- Latitude: -22.079228
- Longitude: -46.902276
- Status: Operante

At the bottom of the form is a green 'Atualizar' (Update) button.

Figura 06 - Janela de Edição de Pontos de Luzes.

Na janela 06 exibe a janela de edição de informações de um ponto de luz.

The screenshot shows the 'Criar Novo Ponto' (Create New Point) form in the EnergyActus system. The form is overlaid on a dashboard that displays a list of light points and their status. The form contains the following fields:

- Endereço: Rua Valério
- Bairro: Morra Libano
- Sector: Sector A
- Tipo de Luz: Led
- Potência Placa Solar?: Sim
- Rendimento da Placa: 0%
- Potência da Placa (W): 150 W
- Quantidade de Luzes: 1
- Tipo de Poste: T
- Potência de Lâmpada: 50 W
- Tipo de Braco: Convencional
- Latitude: -22.0224556
- Longitude: -46.955555
- Status: Operante

At the bottom of the form is a green 'Adicionar' (Add) button. The background dashboard shows a table with columns for 'Postos Operantes', 'Edição/Exclusão', and 'Ligar/Parar'.

Figura 07 - Criar Novo Ponto de Luz(Poste).

Figura 07, mostra o modal de criação de um ponto de luz, nele contemplando todas as informações necessárias para cadastro

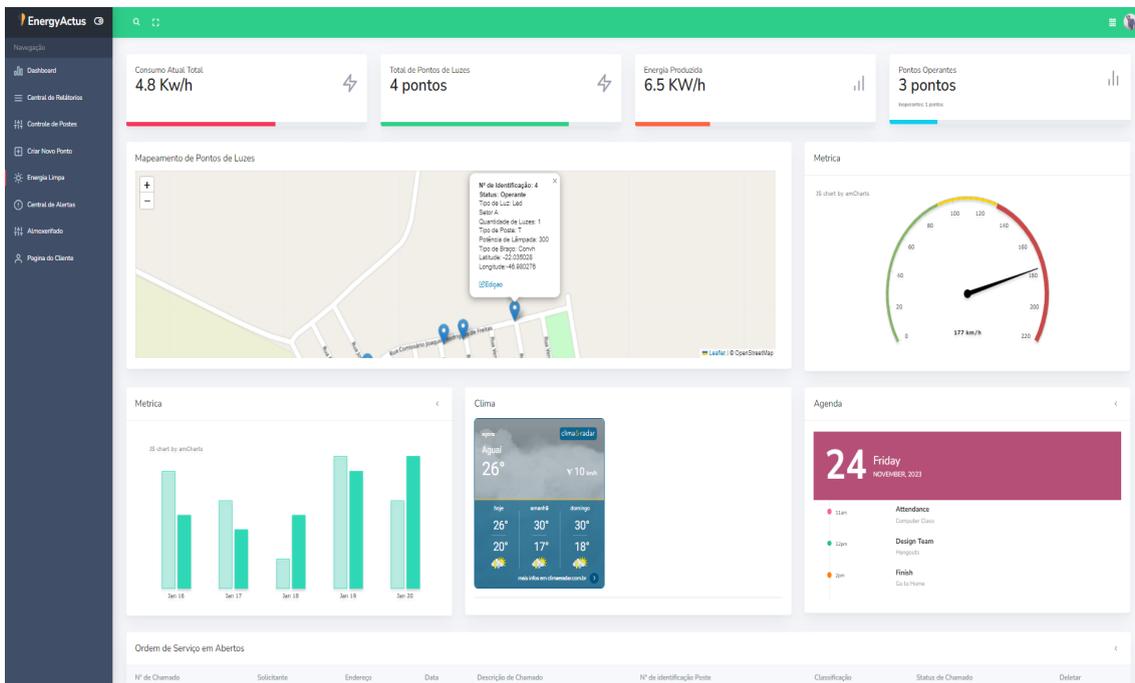
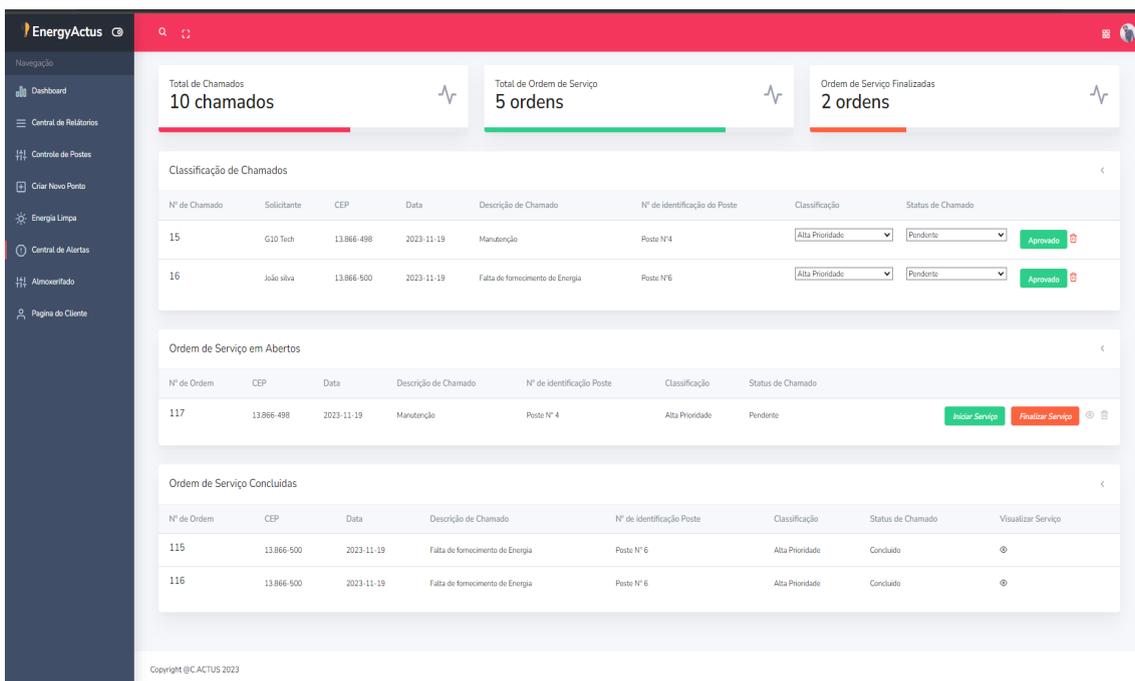


Figura 08 - Dashboard de Energia Limpa (Solar).

Figura 08, mostra a dashboard de energia limpa, uma janela exclusiva para captação e consumo de energia solar, o geomapeamento referenciando onde a postes com placas solares além de possuir indicadores na parte superior da tela com dados de consumo, produção e pontos de luz contemplados.



### Figura 09 - Central de Chamados (Aprovação de Ordem de Serviço).

Figura 09, mostra a seção de central de alertas que ao chegar um novo chamado, a central de alertas realiza a classificação da solicitação e realiza a aprovação ou exclusão se necessário, a partir da aprovação o chamada passa para ordem de serviço em aberto, onde o setor responsável já é acionado e inicia o serviço apertando o “iniciar serviço”, a partir disso o sistema se atualiza automaticamente e coloca de “Pendente” para “Em progresso”, após finalizar a ordem de serviço cai para a seção de concluídas gerando um relatório de prestação de serviço.

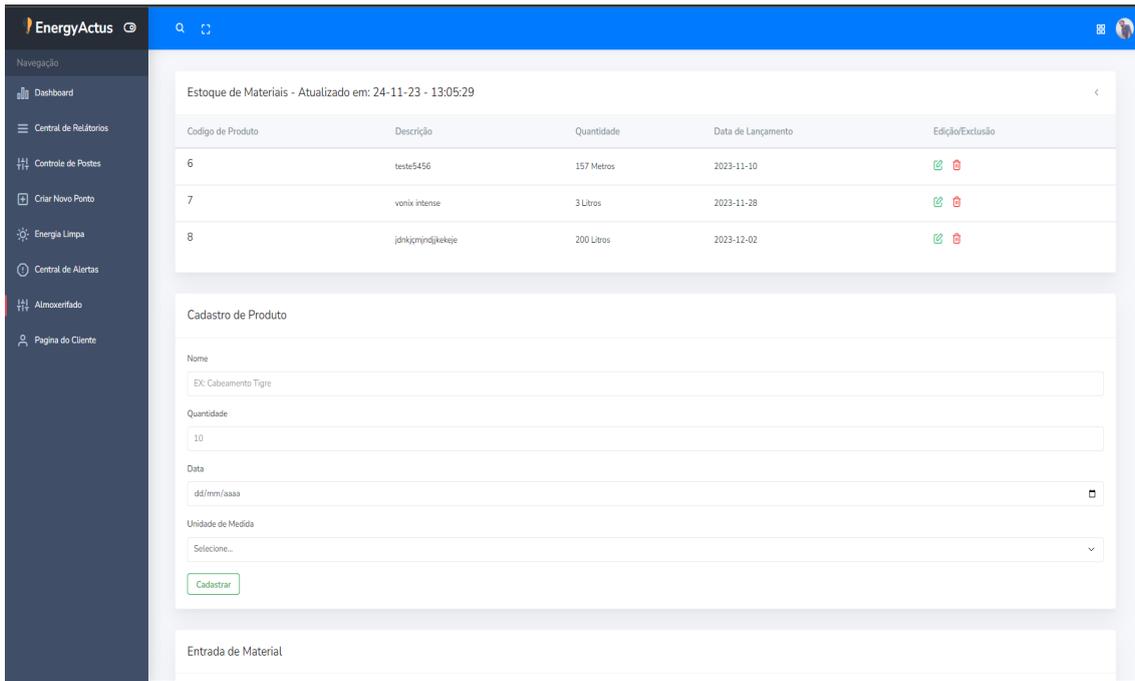
The screenshot shows the 'Relatório de Serviço' (Service Report) form in the EnergyActus system. The form is titled 'Relatório de Serviço' and includes the following fields and sections:

- Preenchimento de relatório de término de serviço**
- Nº de Ordem de Serviço:** 117
- Descrição do serviço realizado:** O serviço realizado...
- O problema apresentado foi solucionado?** Sim
- Visando a Qualidade de nossos serviços, após qualquer ordem de serviço, devem ser realizadas inspeções referentes aos sistemas elétricos do local, sendo assim preencha os campos abaixo de acordo com a situação atual!**
- O sistema elétrico apresenta a tensão recomendada?** Sim
- Os cabos apresentam boas condições?** Sim
- As conexões apresentam boas condições?** Sim
- Os sistemas de segurança apresentam boas condições?** Sim
- Descrição dos Materiais utilizados:** Cabo ethernet - 100 m
- Data de Finalização:** dd/mm/aaaa
- Técnico Responsável pelo serviço:** João Silva - Técnico Chefe

At the bottom of the form, there is a button labeled 'Enviar Relatório'.

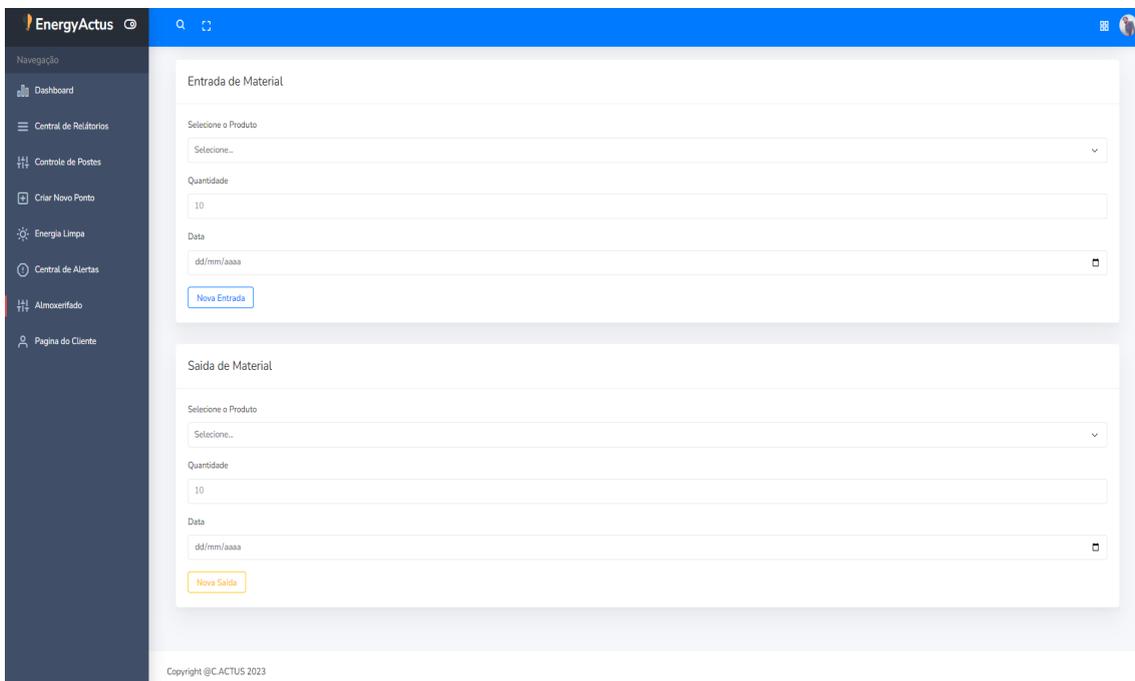
### Figura 10 - Janela de Finalização de Serviço.

Figura 10, mostra a janela de detalhamento do processo com a finalização do serviço, possuindo uma sessão de checklist do serviço e verificação da infraestrutura do local.



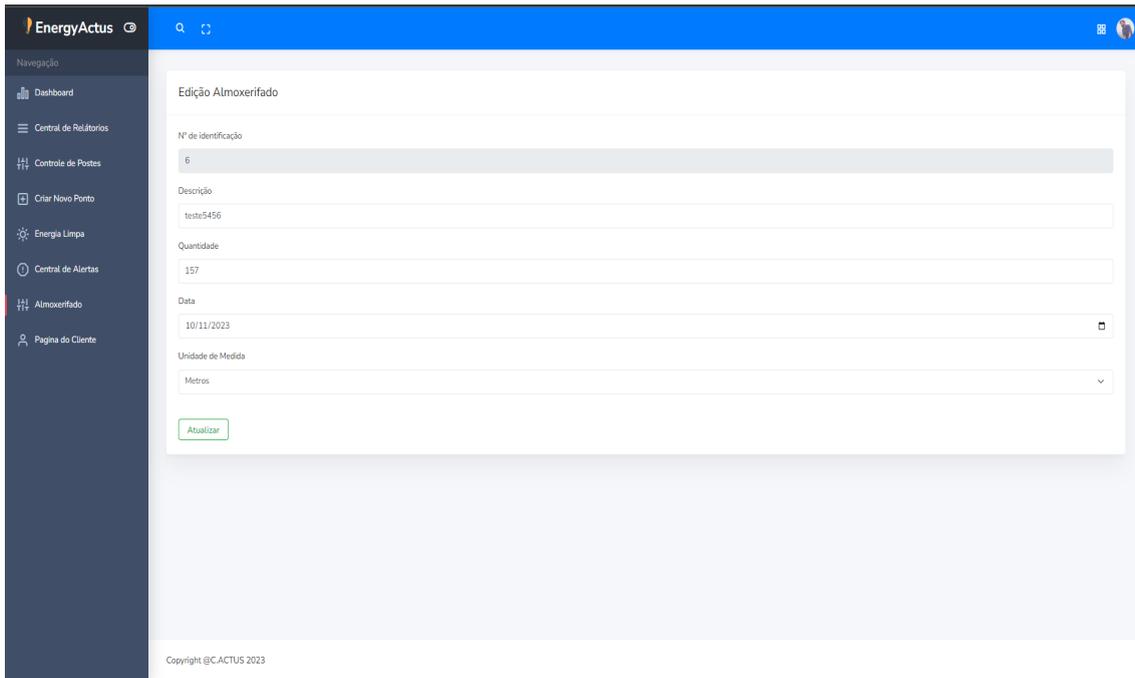
**Figura 11 - Almoxarifado (Parte 1).**

Figura 11, contempla o almoxarifado, onde mostra o estoque de materiais presente contando com edição e exclusão, e logo abaixo o cadastro de um novo produto.



**Figura 12 - Almoxarifado (Parte 2).**

Figura 12, mostra a página de adicionar entradas e saídas de materiais no estoque.



**Figura 13 - Edição Almojarifado.**

Figura 13, mostra a edição de produtos do almojarifado.



**Figura 14 - Janela de Apresentação(Clientes).**

Figura 14, Nosso sistema possui uma página dedicada aos nossos clientes para realizar as solicitações de serviços de uma maneira simples.



E-mail  
você@exemplo.com

---

Rua  
Rua Principal 1234

---

Bairro  
Bairro 1234

---

Cidade

---

Estado

---

Data  
dd/mm/aaaa

O que houve em sua localização?  
Selecione...

Enviar solicitação

c.actus.sistemas@gmail.com

Tel: (19) 99389-0591

© Copyright C.actus 2023

---

### Figura 16 - Janela de Chamados (Parte 2).

Figura 15 e 16, apresenta o formulário apresentado ao cliente para realização de um novo chamado, para maior comodidade e facilitação adicionamos uma API que realiza a busca automática de endereço através do CEP digitado.

# Bem Vindo a C.ACTUS

SOLICITAR UM SERVIÇO

NOSSA HISTÓRIA

NOSSOS SERVIÇOS

## Nossa Historia

Em 2020 nasceu a C.ACTUS com o intuito de ser de promover aos clientes soluções em sistemas dos mais variados tipos.

Somos um grupo que visa fornecer soluções para outras empresas com intenção de impulsionar seu negocio, para isso utilizamos três pontos:

- Usabilidade
- Praticidade
- Responsividade

Acreditamos para atingirmos o sucesso devemos focar em três pontos na criação de um Software. A Usabilidade - programa organizado com informações clara, ícones e campos de fácil compreensão. Praticidade, pensando no cliente de forma em que o mesmo entenda o que é mostrado de forma fácil e objetiva. Responsividade, Ter um controle total de seu software, podendo ter acesso a partir de um celular, tablet, computadores, em que ele sempre se adapte de melhor forma para de sua preferência.

Nossa Missão: Construir sistemas inteligentes para o desenvolvimento de pequenas e medias empresas.

Nossa Visão: Ser reconhecido pela qualidade de construir softwares modernos, praticos e seguros para pequenas e medias empresas.

Nossos Valores: Transparência, Criatividade, Inovação e Personalização.

## Institucional C.actus



**Figura 17 - Janela de Apresentação.**

Figura 17, mostra a página onde os clientes poderão ter acesso e conhecer um pouco da nossa história como empresa.

# Bem Vindo a C.ACTUS

SOLICITAR UM SERVIÇO

NOSSA HISTÓRIA

NOSSOS SERVIÇOS

## O que nós fazemos?

Nós da C.actus desenvolvemos soluções para facilitar atividades das empresas através das Tecnologias com design, praticidade e segurança.

01.

### Sistema Fluxo de Caixa

O grupo C.actus possui um amplo portfólio de aplicações, desde conceitos a protótipos funcionais um deles é o Fluxo de Caixa G10, simples, fácil e seguro, o fluxo de caixa G10 e um dos mais completos a disposição do cliente.

02.

### Sistema E-Commerce

A C.actus Store é uma loja virtual repleta de funcionalidades desde usuários até aos lojistas.

03.

### Sistema STA

Sistema STA é um analisador de temperatura automático para empresas de fabricação de espumas

04.

### Sistema Energy

O Sistema Energy é o mapeamento de pontos de luzes dentro de uma cidade, onde as centrais ou empresas de energia podem utilizar para saber métricas de consumo e defeitos.

© Copyright C.actus 2023

## Figura 18 - Janela de Serviços.

Figura 18, mostra a página onde os clientes poderão ter acesso e conhecer o nosso portfólio de projetos.