**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**ALEX GABRIEL SOARES SOUSA**

**LUCAS ORMENESE ALTIERI**

**VINICIUS DOS SANTOS GIROTTI**

**SUSTENTABILIDADE INDIVIDUAL**

**CAMPINAS**

**2025**

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**ESCOLA POLITÉCNICA**

**ENGENHARIA DE SOFTWARE**

**ALEX GABRIEL SOARES SOUSA**

**LUCAS ORMENESE ALTIERI**

**VINICIUS DOS SANTOS GIROTTI**

**SUSTENTABILIDADE INDIVIDUAL**

Relatório de projeto de sistema, apresentado no componente curricular Projeto Integrador I, do Curso de Engenharia de Software, da Escola Politécnica da Pontifícia Universidade Católica de Campinas.

Orientador: Prof. Dr. André Mendeleck

**CAMPINAS**

**2025**

**SUMÁRIO**

[1. INTRODUÇÃO 4](#_Toc197849323)

[2. JUSTIFICATIVA 5](#_Toc197849324)

[3. OBJETIVOS 6](#_Toc197849325)

[4. ESCOPO 7](#_Toc197849326)

[5. NÃO ESCOPO 22](#_Toc197849327)

[6. REQUISITOS FUNCIONAIS 23](#_Toc197849328)

[7. REQUISITOS NÃO FUNCIONAIS 42](#_Toc197849329)

[8. TESTES 43](#_Toc197849330)

[9. METODOLOGIA APLICADA AO PROJETO 48](#_Toc197849331)

[**10.** **ACOMPANHAMENTO DA GESTÃO DO PROJETO** 51](#_Toc197849332)

[**11.** **PREMISSAS** 52](#_Toc197849333)

[12. RESTRIÇÕES 54](#_Toc197849334)

[13. REFERÊNCIAS 55](#_Toc197849335)

# INTRODUÇÃO

A sustentabilidade ambiental, embora amplamente discutida, enfrenta desafios na tradução para ações cotidianas efetivas. A Organização das Nações Unidas (ONU) estabeleceu os Objetivos de Desenvolvimento Sustentável (ODS) como diretrizes globais, destacando-se o ODS 3, que relaciona saúde e equilíbrio ambiental, e o ODS 12, voltado ao consumo sustentável.

No entanto, relatórios como os do Programa das Nações Unidas para o Meio Ambiente (2021) apontam que a adesão individual a práticas sustentáveis ainda é limitada, frequentemente prejudicada pela falta de orientação personalizada e infraestrutura adequada.

Diante desse cenário, este projeto propõe uma plataforma interativa que combina diagnóstico e classificação de hábitos.

Delimitando-se a quatro eixos críticos – água, energia, resíduos e mobilidade –, a ferramenta utiliza indicadores (ex.: litros de água/dia, taxa de recicláveis) para classificar o usuário e oferecer recomendações práticas, como redução de plástico e planejamento alimentar.

A hipótese central é que a integração entre autoconhecimento, suporte contextualizado e engajamento coletivo pode superar barreiras comportamentais, transformando metas globais em ações locais mensuráveis.

Ao vincular escolhas individuais aos ODS, busca-se preencher uma lacuna identificada: a necessidade de ferramentas que convertam princípios abstratos em impactos tangíveis, fortalecendo a relação entre responsabilidade pessoal e avanços sistêmicos.

# JUSTIFICATIVA

A transição para hábitos sustentáveis é urgente, mas esbarra em desafios práticos. Dados globais indicam que 80% das pessoas reconhecem a importância da sustentabilidade, porém apenas 20% agem consistentemente.

No Brasil, 40% da água é desperdiçada, 15% das emissões de CO₂ vêm do transporte individual, e apenas 4% dos resíduos são reciclados. Esses números expõem a lacuna entre consciência e ação, agravada pela falta de ferramentas que orientem mudanças no dia a dia.

Alinhado aos Objetivos de Desenvolvimento Sustentável (ODS 3 e 12), o projeto procura ajudar essas dificuldades com uma plataforma que diagnostica hábitos, oferecendo soluções adaptáveis.

Estima-se que práticas como planejamento de compras poderiam reduzir o desperdício doméstico em 30%, comprovando que pequenas ações, quando orientadas, geram impacto coletivo. A iniciativa justifica-se ao transformar metas abstratas em passos concretos, democratizando a sustentabilidade como escolha diária, mensurável e acessível.

# OBJETIVOS

**OBJETIVO GERAL:**

Criar uma plataforma interativa que classifique o nível de sustentabilidade individual com base no consumo de água, energia, geração de resíduos e uso de transporte.

Promovendo a adoção de hábitos sustentáveis por meio de diagnósticos personalizados, soluções adaptáveis e acompanhamento em tempo real, alinhado aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU.

**OBJETIVOS ESPECÍFICOS:**

1. Analisar padrões de consumo nas categorias de água, energia, resíduos e mobilidade, utilizando métricas claras como litros de água/dia, kWh/dia, porcentagem de recicláveis, para classificar o usuário em níveis de sustentabilidade (alta, moderada ou baixa).
2. Desenvolver uma plataforma intuitiva que forneça um diagnóstico detalhado dos hábitos do usuário, destacando pontos críticos e oportunidades de melhoria, com interface acessível e visualização de progresso.
3. Oferecer um banco de soluções personalizadas (ex.: tutoriais para redução de plástico, planejamento de compras sustentáveis, alternativas de transporte) adaptadas ao perfil e à realidade de cada usuário.

# ESCOPO

O nosso objetivo ao desenvolver a plataforma é atingir pessoas que buscam adotar hábitos sustentáveis no consumo de recursos. Para isso, será necessário que o usuário forneça ao sistema dados como consumo médio de água (litros/dia), gasto energético mensal (kWh), quantidade de resíduos não recicláveis gerados (kg/semana) e tipo de transporte utilizado (ex.: carro, transporte público).

Com os dados obtidos, o utilizador será classificado em um índice de sustentabilidade (alta, moderada ou baixa).

Com base em métricas da Organização das Nações Unidas (ODS 3 e 12), e receberá recomendações personalizadas — como substituição de plástico por alternativas reutilizáveis ou métodos para reduzir desperdício alimentar.

Além disso, o usuário poderá salvar seu progresso em um banco de dados, acessar seu progresso diário, visualizando seu impacto em tempo real. A plataforma será desenvolvida em Python, integrando análise de dados e orientações práticas para transformar a sustentabilidade em ações mensuráveis e adaptáveis à rotina.

**PONDERAÇÃO**

O projeto seguirá uma metodologia de abrangência em quatro principais campos, sendo eles baseados no consumo de água, luz, meios de locomoção e geração de resíduos. Para tanto sua classificação será descrita da seguinte maneira:

Consumo de Água:

|  |  |
| --- | --- |
| **Classificação** | **Medida** |
| Meio Ambiente Agradece | <100 litro/dia |
| Alta Sustentabilidade | 100 - 150 litros/dia |
| Moderada Sustentabilidade | 150 - 200 litros/dia |
| Baixa Sustentabilidade | >200 litros/dia |

Consumo de Energia:

|  |  |
| --- | --- |
| **Classificação** | **Medida** |
| Meio Ambiente Agradece | <2,5 kWh/dia |
| Alta Sustentabilidade | 2,5 - 5 kWh/dia |
| Moderada Sustentabilidade | 5 - 10 kWh/dia |
| Baixa Sustentabilidade | >10 kWh/dia |

Classificação de Transporte:

|  |  |
| --- | --- |
| **Classificação** | **Medida** |
| Meio Ambiente Agradece | Bicicleta, Caminhada |
| Alta Sustentabilidade | Bicicleta, veículos elétricos |
| Moderada Sustentabilidade | Misto entre bicicleta e veículos móveis privados e públicos |
| Baixa Sustentabilidade | Veículos movidos a fósseis |

Classificação da Geração de Resíduos Não Recicláveis

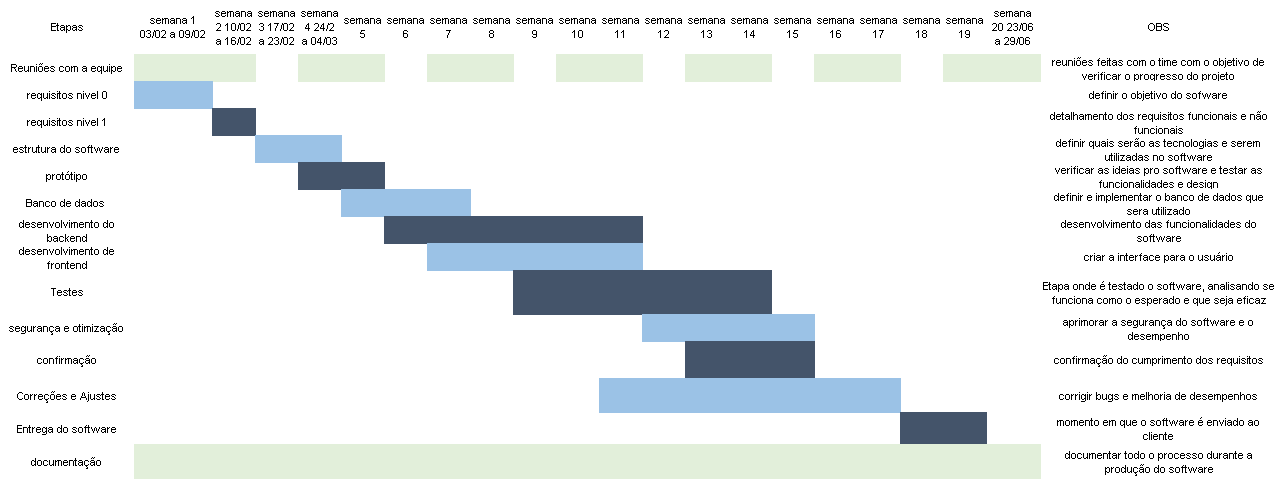
|  |  |
| --- | --- |
| **Classificação** | **Medida** |
| Meio Ambiente Agradece | <20% de geração de resíduos |
| Alta Sustentabilidade | 20 - 50%de geração de resíduos |
| Moderada Sustentabilidade | 50 - 60% de geração de resíduos |
| Baixa Sustentabilidade | >60% de geração de resíduos |

Como serão coletadas essas informações:

Para o monitoramento do consumo hídrico, a coleta de dados será realizada através de hidrômetros. No caso da energia elétrica, utilizaremos os registros dos medidores de consumo (relógios de luz). Quanto à gestão de resíduos, a análise será feita mediante pesagem do lixo orgânico e/ou estimativa do usuário sobre seu próprio.

Essa medição quantitativa permitirá avaliar se os níveis de geração de detritos estão alinhados com padrões sustentáveis estabelecidos. E a forma de meçar a classificação de transporte, ocorrerá de maneira direta, onde o usuário escreverá que transporte foi usado e quantas, vezes, e assim será feita sua classificação.

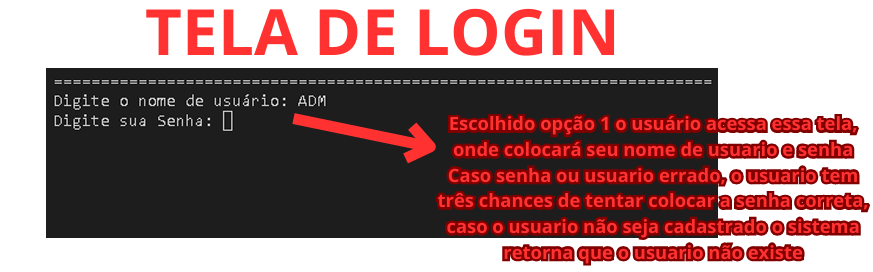
**Cronograma:**



**Diagrama das Telas**

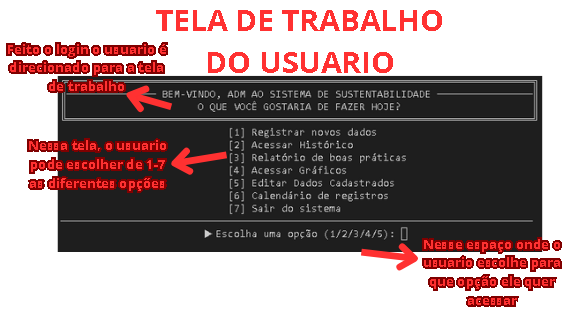
**Tela Inicial:**

****

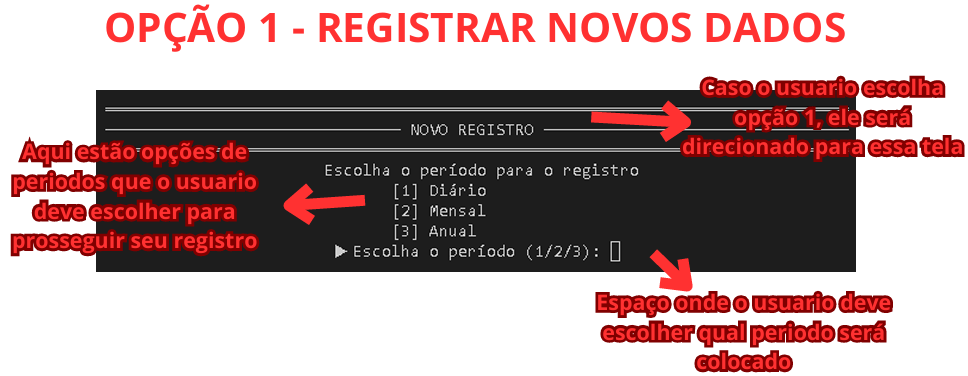
**Tela de Login:** 

**Tela de Cadastro:** 

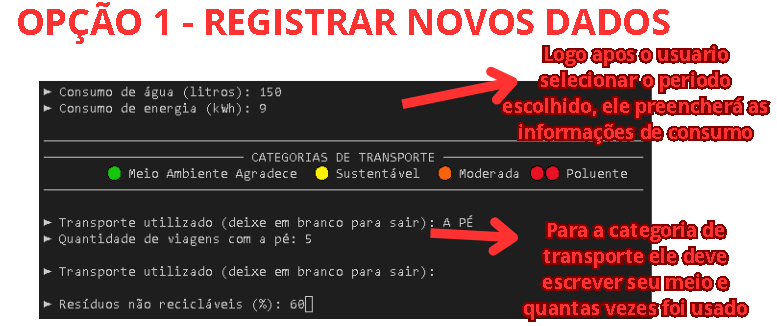
**Tela de Trabalho:**

****

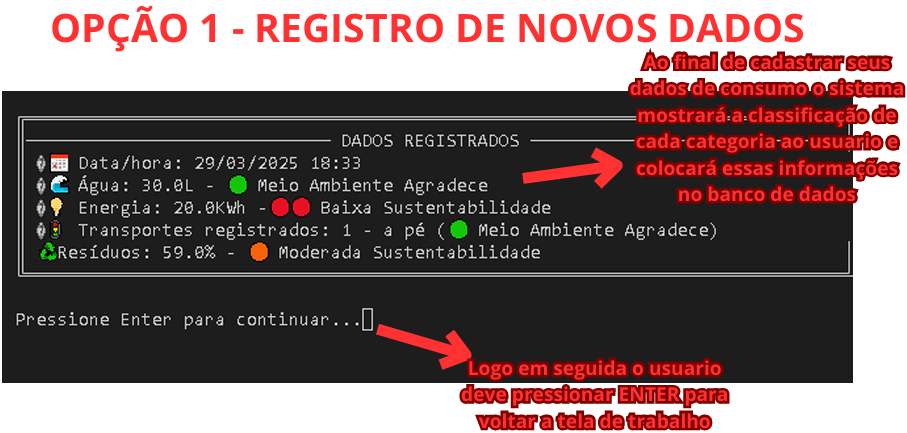
**Tela de Registro de Novos Dados - Período:**

****

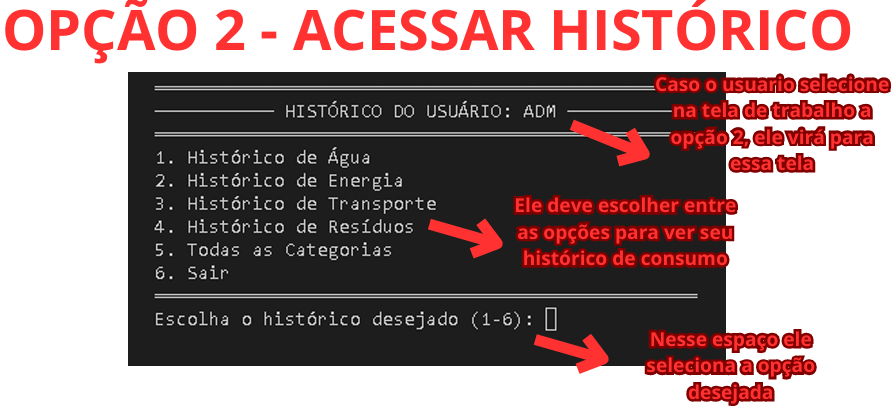
**Tela de Registro de Novos Dados – Colocar Dados:**



**Tela de Registro de Novos Dados – Resultado:**

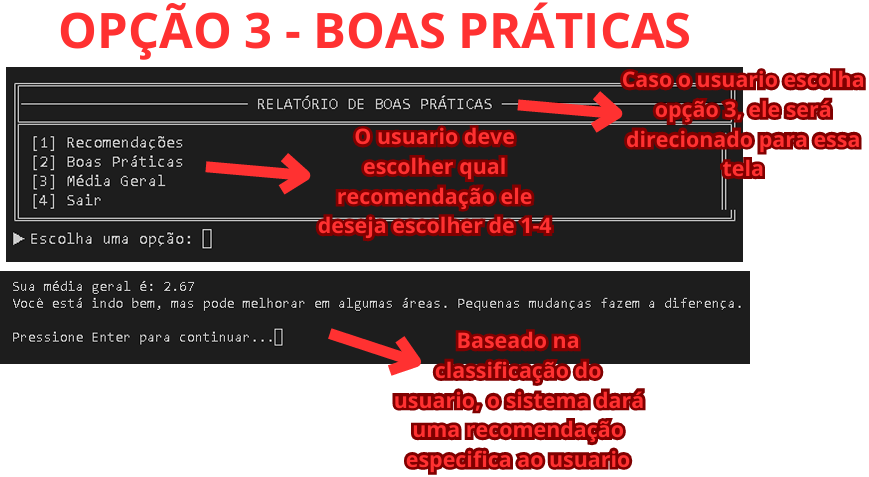
****

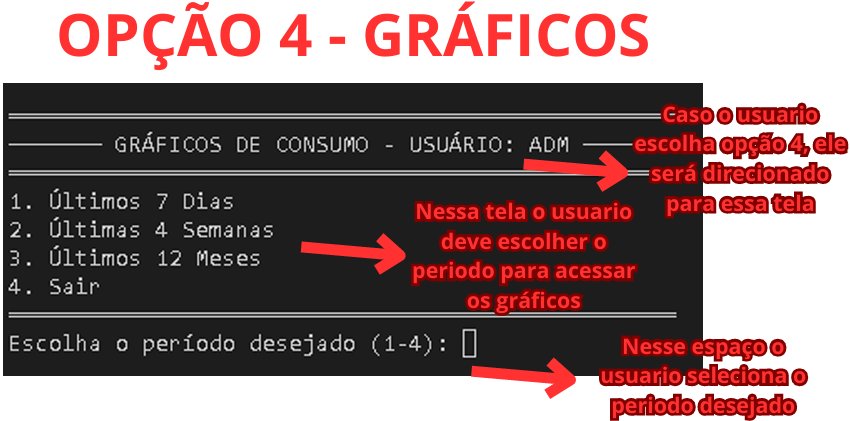
**Tela de Histórico - Menu:**

****

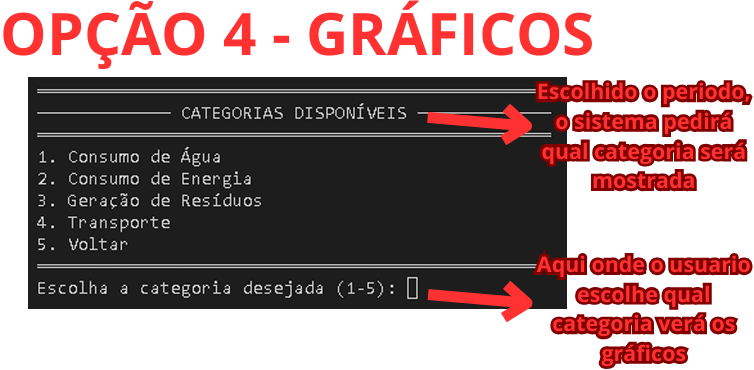
**Tela de Histórico – Histórico:**

****

**Tela de Boas Práticas:Tela de Gráficos – Menu Periodo:**

****

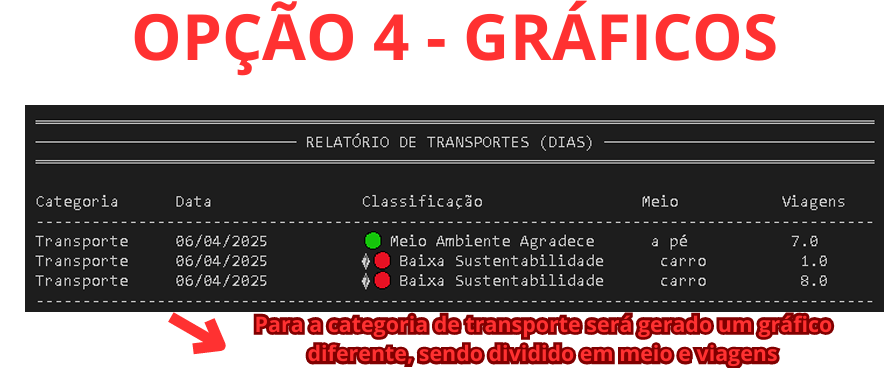
**Tela de Gráficos – Menu Categoria:**

****

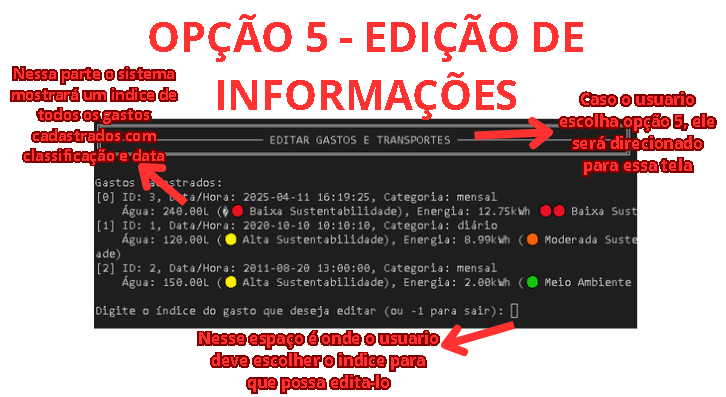
**Tela de Gráficos – Gráficos Gerais:**

****

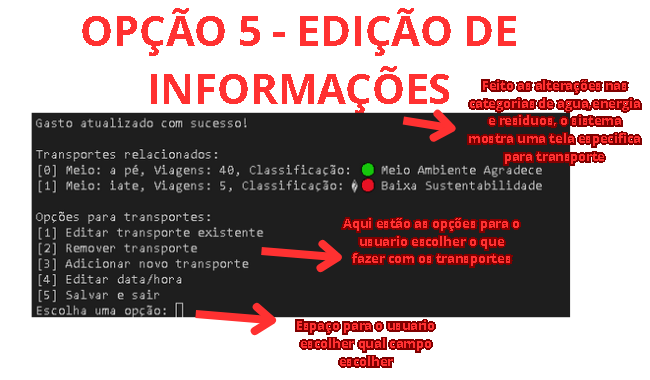
**Tela de Gráficos – Gráfico especifico para transporte:**

****

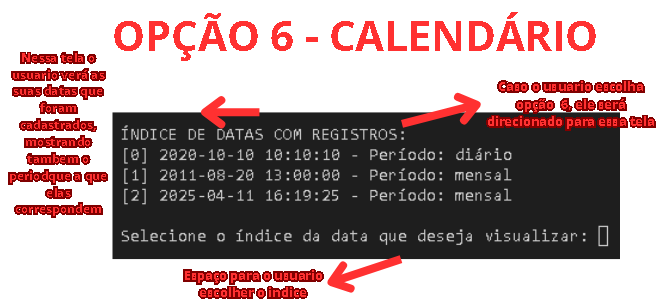
**Tela de Edição de Informações - Geral:**



**Tela de Edição de Informações – Salvamento e especifico para transporte:**

****

**Tela de Calendário – Tela inicial:**

****

**Tela de Calendário – Resultados:**

****

**Banco de Dados**

O banco de dados foi desenvolvido em MySQL Workbench, e integrado de duas formas ao sistema, pelo método de conexão local e por rede, que nesse caso pode variar conforme a circunstância da rede, uma vez que dependerá da organização.

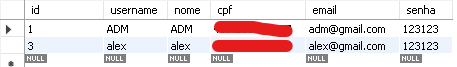
Para tal, existe um parâmetro visado pelo usuário da rede e a senha do mesmo. E deve ser ampliado o método de transcrição por caracteres especiais, uma vez que o sistema se utiliza de meios para fornecer suas classificações. Sendo assim, deve regular as primeiras transcrições do sistema, para que o MySQL reconheça os meios do sistema e o deixe funcionar corretamente.

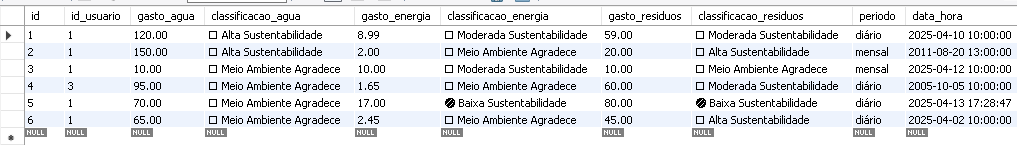
**Tables criadas:**

**Tables criado modelagem visual:**

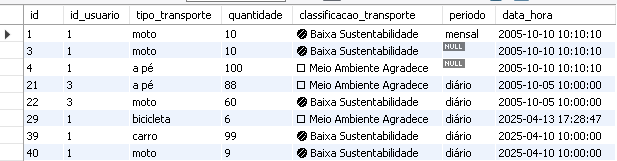
****

**Usuarios:**

****

**Gastos dos usuários:**

**Transportes dos usuários:**

****

Vale ressaltar que parâmetros para fim de cadastro ou registro de novas informações, que são CPF, Username e E-mail, são chaves únicas, então caso o um novo registro seja feito, e esse registro já exista no sistema, o próprio banco de dados comunicará ao FrontEnd que tal ação não pôde ser executado, visto que isso infringe a legitimidade do sistema, com informações e dados duplicados.

**CRIPTOGRAFIA**

Os processos de criptografia e descriptografia conforme implementados no script Python hill\_cipher\_logic.py. Este script utiliza a Cifra de Hill, um algoritmo criptográfico clássico que opera sobre blocos de caracteres através de operações de álgebra linear.

**1. Configuração e Parâmetros Iniciais**

* Antes da execução dos processos de cifragem ou decifragem, o sistema é inicializado com os seguintes parâmetros e componentes essenciais:
* Módulo de Operação (MODULUS): Estabelecido em 256, este valor define o corpo matemático sobre o qual todas as operações aritméticas são realizadas. A escolha de 256 alinha-se com a representação de caracteres no padrão ASCII estendido.
* Tamanho do Bloco (BLOCK\_SIZE): Configurado como 2, indica que o texto a ser processado é dividido em segmentos de dois caracteres. Este parâmetro também determina as dimensões da matriz chave (n x n, onde n é o BLOCK\_SIZE).
* Matriz Chave (KEY\_MATRIX): Uma matriz quadrada de dimensões BLOCK\_SIZE x BLOCK\_SIZE (neste caso, 2x2) é utilizada como o segredo criptográfico. O exemplo no script utiliza a matriz [[3, 3], [2, 5]]. Um requisito fundamental para a funcionalidade da Cifra de Hill é que esta matriz seja invertível módulo MODULUS. O script valida esta condição verificando se o determinante da matriz possui um inverso multiplicativo modular 256.
* Matriz Chave Inversa (KEY\_MATRIX\_INV): Para o processo de descriptografia, é necessária a matriz inversa da KEY\_MATRIX no contexto do módulo definido. O script pré-calcula esta matriz inversa (KEY\_MATRIX\_INV) utilizando a função matrix\_mod\_inverse. Esta função emprega o Algoritmo Estendido de Euclides para determinar o inverso modular do determinante da matriz chave e, subsequentemente, calcula a matriz adjunta para obter a inversa modular da matriz completa.

**2. Processo de Criptografia**

A criptografia de uma mensagem de texto plano é realizada pela função encrypt(plaintext), seguindo as etapas subsequentes:

1. Preparação do Texto (\_prepare\_text):

O texto de entrada (plaintext) é inicialmente convertido para uma representação numérica. Cada caractere é substituído pelo seu valor correspondente na tabela ASCII, através da função ord().

Para garantir que o texto possa ser dividido integralmente em blocos de tamanho BLOCK\_SIZE, um mecanismo de preenchimento (padding) é aplicado. Se a quantidade de caracteres numéricos resultante não for um múltiplo de BLOCK\_SIZE, são adicionados caracteres nulos (valor ASCII 0) ao final da sequência numérica até que o comprimento seja adequado.

2. Operação de Cifragem por Blocos:

A sequência numérica preparada é segmentada em blocos, cada um contendo BLOCK\_SIZE números. Cada bloco, tratado como um vetor linha, é submetido a uma multiplicação matricial pela KEY\_MATRIX.

O resultado desta multiplicação é então reduzido módulo MODULUS (256). Matematicamente, para cada bloco de texto plano $P$ e a matriz chave $K$, o bloco cifrado $C$ é obtido por $C \equiv P \cdot K \pmod{256}$. Os vetores numéricos resultantes de cada bloco cifrado são concatenados para formar a sequência numérica completa do texto cifrado.

3. Formatação da Saída Cifrada (\_format\_output):

A sequência de números que representa o texto cifrado é convertida de volta para caracteres, utilizando a função chr(). Quaisquer caracteres de preenchimento (NUL) que foram adicionados durante a etapa de preparação são removidos do final da mensagem.

O produto final é uma string de texto cifrado. É importante notar que esta string pode conter caracteres não imprimíveis, dependendo dos valores numéricos gerados. O script sugere, como alternativa, o retorno dos dados cifrados como uma lista de inteiros ou uma string codificada em Base64 para mitigar problemas de armazenamento ou transmissão.

**3. Processo de Descriptografia**

A recuperação da mensagem original a partir de um texto cifrado é gerenciada pela função decrypt(ciphertext). O processo espelha a criptografia, utilizando a matriz chave inversa:

1. Preparação do Texto Cifrado (\_prepare\_text):

O texto cifrado (ciphertext) é primeiramente convertido para sua representação numérica ASCII. A função \_prepare\_text é reutilizada para esta conversão, assegurando o tratamento consistente de qualquer padding que possa estar presente, caso a string cifrada tenha sido armazenada diretamente ou se seu comprimento não for um múltiplo exato do BLOCK\_SIZE.

O script contempla a possibilidade de o texto cifrado estar codificado em Base64, indicando a necessidade de decodificação prévia nesse cenário.

2. Operação de Decifragem por Blocos:

A sequência numérica do texto cifrado é dividida em blocos de BLOCK\_SIZE elementos.Cada bloco cifrado, também tratado como um vetor linha, é multiplicado pela KEY\_MATRIX\_INV (a matriz inversa da chave original).

O resultado desta operação matricial é, similarmente, reduzido módulo MODULUS (256). Para cada bloco cifrado $C$ e a matriz chave inversa $K^{-1}$, o bloco de texto plano original $P$ é recuperado por $P \equiv C \cdot K^{-1} \pmod{256}$.

Os vetores numéricos resultantes de cada bloco decifrado são concatenados para reconstruir a sequência numérica do texto original.

3. Formatação da Saída Decifrada (\_format\_output):

A sequência numérica recuperada é convertida novamente para caracteres através da função chr(). Quaisquer caracteres de preenchimento (NUL) são removidos do final da string resultante. O resultado deste processo é a mensagem de texto plano original.

# NÃO ESCOPO

* Anexo de arquivos.
* Alteração de requisitos
* Alteração/manipulação do banco de dados pelo usuário
* Exclusão de dados
* Não serão necessárias informações adicionais
* Tradução para idiomas além do português.
* Integração com redes sociais para compartilhamento automático de dados
* Funcionalidade em modo off-line
* Certificação oficial de sustentabilidade para empresas ou usuários.

# REQUISITOS FUNCIONAIS

**REQUISITO FUNCIONAL 1**: Cadastro do usuário

**Descrição:** O sistema deve permitir que os usuários se cadastrem no software, inserindo dados: E-mail, nome de usuário, senha e CPF. Sendo realizado ao selecionar sua opção no sistema. Apresentando uma opção ao final de efetuar o cadastro ou não.

**O que deve ser atendido:**

* Os dados cadastrados devem ser únicos no banco de dados;
* O sistema deve pedir uma confirmação de cadastro;
* A senha deve conter 6(seis) caracteres ou mais;
* O software deve analisar se o e-mail cadastrado está correto, caso contrário, deve notificar o usuário;
* O software deve verificar o CPF cadastrado
* O sistema deve criptografar e armazenar as informações no banco de dados

**REQUISITO FUNCIONAL 2:** Login

**Descrição:** O sistema deve apresentar ao usuário uma tela onde o será possível realizar o login no software, essa tela deve apresentar dois campos de entrada de dados, um para inserir o nome de usuário, e outro para o usuário inserir sua senha.

**O que deve ser atendido:**

* O sistema deve analisar se o nome de usuário existe no banco de dados, caso contrário, o sistema deve notificar o usuário com uma mensagem de erro;
* O sistema deve analisar se a senha inserida pelo usuário é mesma cadastrada no banco de dados, se não, o usuário deve ser notificado
* Caso 3 (três) tentativas de login ou senha fracassada, o sistema deve bloquear o acesso.
* Se o login for realizado com sucesso, o sistema deve redirecionar o usuário para a tela principal do software.

**REQUISITO FUNCIONAL 3:** Tela de Trabalho

**Descrição:** O sistema após o login efetuado, exibirá essa tela ao usuário. Essa tela permite o usuário escolher e acessar diferentes campos, sendo eles para cadastrar novas informações, acessar histórico, relatório de boas práticas, visualizar gráficos de consumo e sair. Mas como função principal, essa tela deve servir para o usuário adicionar seus dados de consumo.

**O que deve ser atendido:**

1. Exibir uma mensagem de boas-vindas ao usuário;
2. O sistema permite que o usuário escolha entre diferentes períodos para fazer o cadastro de seus gastos;

* Consumo de água (em litros)
* Consumo de energia (em kWh)
* Uso de transporte
* Geração de resíduos não recicláveis (em %)

1. Ao terminar de preencher as informações, o sistema mostrará a classificação daquele usuário e enviará para o banco de dados as informações.

**REQUISITO FUNCIONAL 4:** Tela de Histórico

**Descrição:** Dentro da tela de trabalho, após o usuário escolher a opção de histórico, será exibido essa tela. Para acessar essa tela o usuário precisa ter alguma informação cadastrada, caso não à tenha o sistema, não permitirá o acesso.

**O que deve ser atendido:**

1. Exibir um menu de opções onde o usuário pode escolher qual opção desejar ver seu histórico, sendo a última, todas
2. Deve ser exibido a classificação do usuário e data e hora daquele registro
3. Para a opção de transporte, apresentará o nome do seu meio de transporte e sua classificação

**REQUISITO FUNCIONAL 5:** Tela de Boas Práticas

**Descrição:** Dentro da tela de trabalho, após o usuário escolher a opção de boas práticas, será exibido esse campo. Para acessar essa tela o usuário precisa ter alguma informação cadastrada, caso não à tenha o sistema, não permitirá o acesso.

**O que deve ser atendido:**

1. Será exibido três opções de escolhas ao usuário, sendo recomendações, boas práticas e média geral;
2. Conforme a classificação do usuário o sistema exibirá temas e escolhas dando recomendações, boas práticas e aconselhando.

**REQUISITO FUNCIONAL 6:** Tela de gráficos

**Descrição:** Dentro da tela de trabalho, após o usuário escolher a opção de gráficos, será exibido esse campo. Para acessar essa tela o usuário precisa ter alguma informação cadastrada, caso não à tenha o sistema, não permitirá o acesso.

**O que deve ser atendido:**

1. Será exibido três opções de escolhas ao usuário, sendo divididas em períodos, dos últimos 7 dias, das últimas 4 semanas e dos últimos 12 meses.
2. Conforme o período escolhido, o usuário deve escolher qual categoria ele deseja visualizar
3. Na categoria de transporte será exibido uma tabela mostrando o tipo de transporte e sua classificação, separado por tipo e data
4. Nas categorias água, energia e geração de resíduos não recicláveis, será exibido um gráfico de barras verticais baseado nos dados cadastrados daquele usuário naquela categoria

**REQUISITO FUNCIONAL 7:** Edição de informações

**Descrição:** Dentro da tela de trabalho, após o usuário escolher a opção de edição de dados, será exibido esse campo. Para acessar essa tela o usuário precisa ter alguma informação cadastrada, caso não à tenha no sistema, não permitirá acesso.

**O que deve ser atendido:**

1. Será exibido uma lista de opções que são os índices que o usuário tem cadastro.
2. Conforme a escolha do índice, o usuário pode editar, remover, adiciona e/ou salvar

**REQUISITO FUNCIONAL 8 –** Calendário

**Descrição:** Dentro da tela de trabalho, após o usuário escolher a opção de calendário, será exibido esse campo. Para acessar essa tela o usuário precisa ter alguma informação cadastrada, caso não à tenha no sistema, não permitirá acesso.

**O que deve ser atendido:**

1. Será exibido uma lista de todos as datas/horas que o usuário fez algum registro no programa
2. De acordo com o índice, o usuário escolher qual data selecionar
3. Conforme as datas serão exibidas em verde o dia exato daquele mês, sendo o exato momento em que houveram registros, e em vermelho, os que não houveram

**REQUISITO FUNCIONAL 9 –** BackEnd (MySQL Workbench)

**Descrição:** Dentro de tudo que se é gerado de informações sobre o usuário, são armazenadas no banco de dados, que por sua vez, deve estar conectado em uma host local ou ethernet. O arquivo corresponde ao banco de dados, não tem utilidade sozinho, sendo este basicamente uma ponte entre o sistema e o banco de dados.

**O que deve ser atendido:**

1. Deve ser armazenado todo e qualquer tipo de informação que o usuário vier a cadastrar

**REQUISITO FUNCIONAL 10 –** Criptografia

**Descrição:** A criptografia está sendo tangida na parte que se refere a senha de usuário. Em uma metodologia da cifra de hill, utilizando 256 caracteres.

**O que deve ser atendido:**

1. O sistema deve criptografar a senha de usuário e a descriptografá-la.

**CASOS DE USO:**

**REQUISITO FUNCIONAL 1 –** Cadastro do usuário

**Tela Inicial:**

****

**Tela de Registro:**



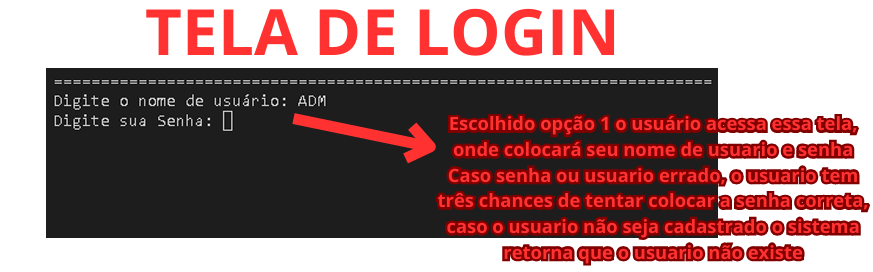
* **Pré-Condições:** Acesso à internet
* **Validações:** Nome de usuário deve ser único; senha deve conter pelo menos 6 caracteres
* **Ator principal:** Usuário
* **Requisitos especiais:** Os dados devem ser criptografados

**Fluxo principal:**

|  |  |
| --- | --- |
| **Ações do usuário** | **Ações do sistema** |
| Acessar a tela de cadastro | Exibir a tela de cadastro |
| Preencher os campos (nome, e-mail, CPF, Senha) | Validar os dados |
| Confirmar dados | Validar informações não existem no sistema |
|  | Enviar dados para o Banco de dados |
|  | Exibir mensagem de sucesso |
|  | Redirecionar para tela de login |

**REQUISITO FUNCIONAL 2 –** Login

**Tela de Login:**



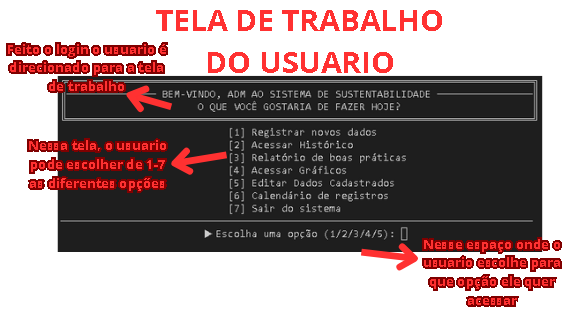
* **Pré-condições:** O usuário deve estar cadastrado no sistema
* **Validações:** O nome de usuário deve existir no banco de dados, a senha deve ser compatível com a cadastrada.
* **Ator principal:** usuário
* **Requisitos especiais:** Após 3 tentativas incorretas, o sistema bloqueará o acesso por 10 minutos

**Fluxo principal:**

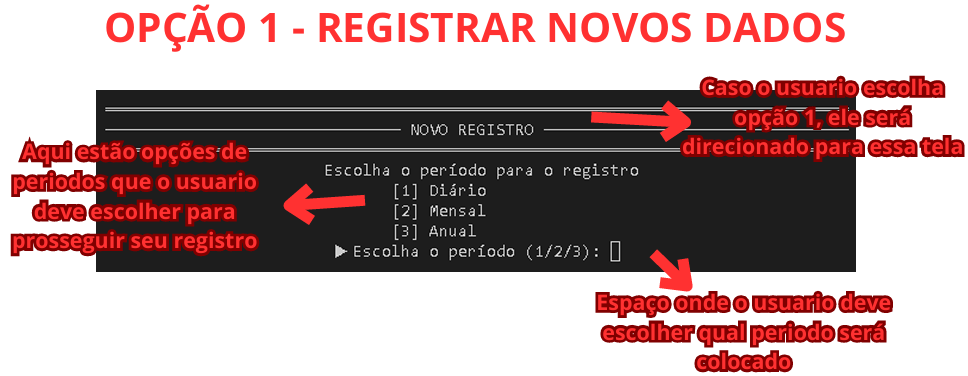
|  |  |
| --- | --- |
| **Ações do usuário** | **Ações do sistema** |
| Acessar tela de login | Exibir tela de login |
| Inserir nome de usuário e senha | Validar se os dados existem no banco |
|  | Verificar se a senha está correta |
|  | Se errar 3 vezes, bloquear o acesso |
|  | Se correto redirecionar para a tela principal |

**REQUISITO FUNCIONAL 3 –** Tela Principal

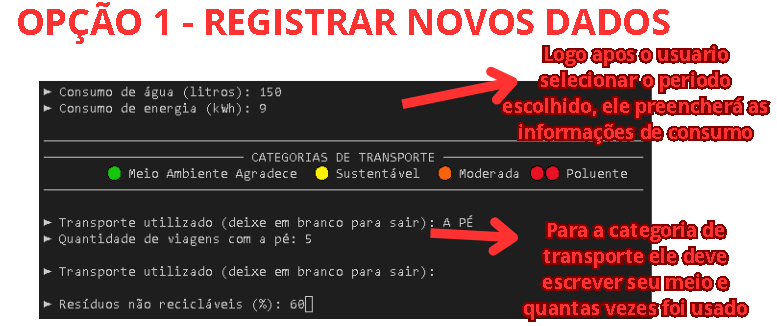
**Tela Principal:**

****

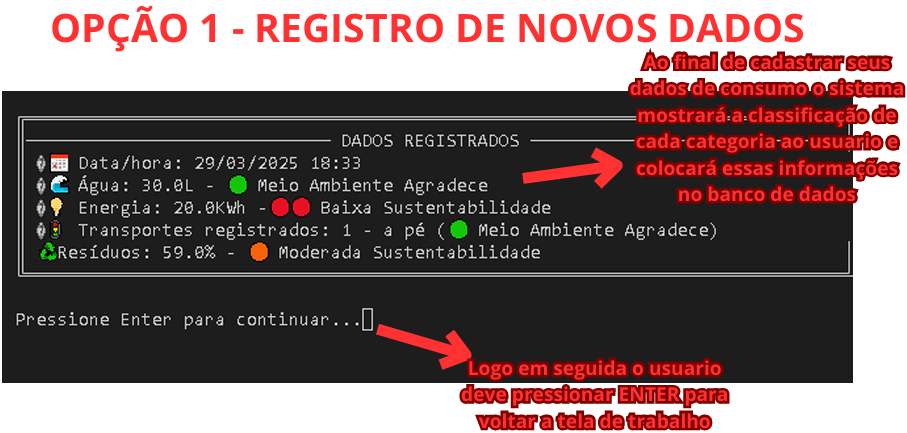
**Tela de Registro de Novos Dados - Período:**

****

**Tela de Registro de Novos Dados – Colocar Dados:**



**Tela de Registro de Novos Dados – Resultado:**

****

* **Pré-condições**: Usuário deve estar logado
* **Validações**: O usuário deve estar autenticado no sistema
* **Ator principal:** Usuário
* **Requisitos especiais:** Exibir a mensagem de boas-vindas personalizada.

|  |  |
| --- | --- |
| **Ações do usuário** | **Ações do sistema** |
| Realizar login no sistema | Exibir tela de boas-vindas personalizada |
| Selecionar opção de cadastrar informações | Exibir opções: Água, Energia Transporte, Resíduos |
| Inserir dados de consumo | Confirmar o registro dos dados |
|  | Armazenar informações no banco de dados |
|  | Mostrar o resultado da classificação de cada categoria |

**Fluxo principal:**

**REQUISITO FUNCIONAL 4 –** Tela de histórico de consumo

**Tela de Histórico – Histórico:**

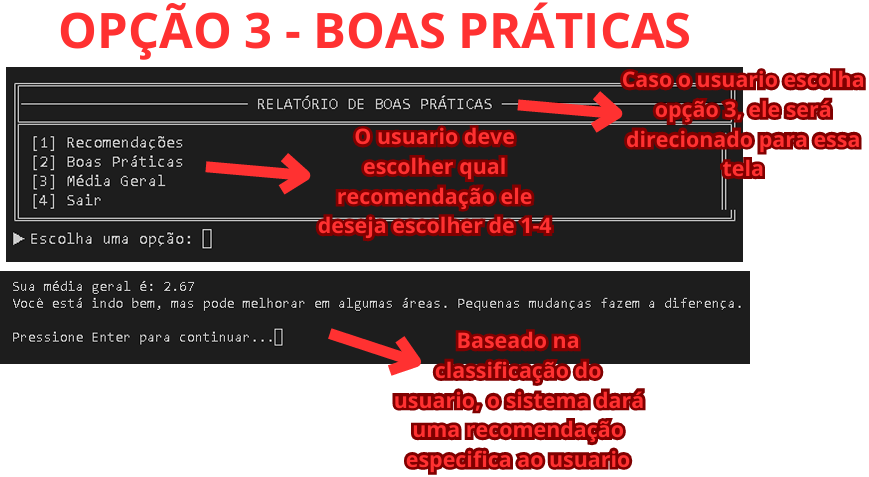
****

* **Pré-condições:** O usuário deve ter registrado os dados do consumo
* **Validação:** O software deve exibir os dados corretamente
* **Ator principal:** Usuário
* **Requisitos especiais:** Exibir uma mensagem com a classificação de consumo

**Fluxo principal:**

|  |  |
| --- | --- |
| **Ações do usuário** | **Ações do sistema** |
| Acessar a tela de histórico | Exibir tabela com opções de categoria |
| Escolher histórico por categoria | Exibir consumo da categoria escolhida |
| Conferir classificação do consumo | Exibir mensagens de sustentabilidade (Meio ambiente agradece, Desperdício, etc) |

**REQUISITO FUNCIONAL 5 –** Tela de boas práticas

**Tela de Boas Práticas:**

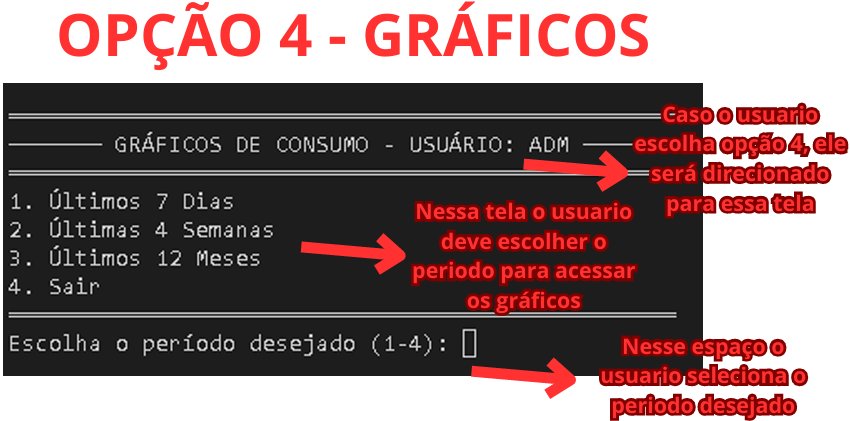
* **Pré-condições:** O usuário deve ter registrado os dados do consumo
* **Validação:** O software deve exibir os dados corretamente
* **Ator principal:** Usuário
* **Requisitos especiais:** Exibir uma mensagem de acordo com a classificação

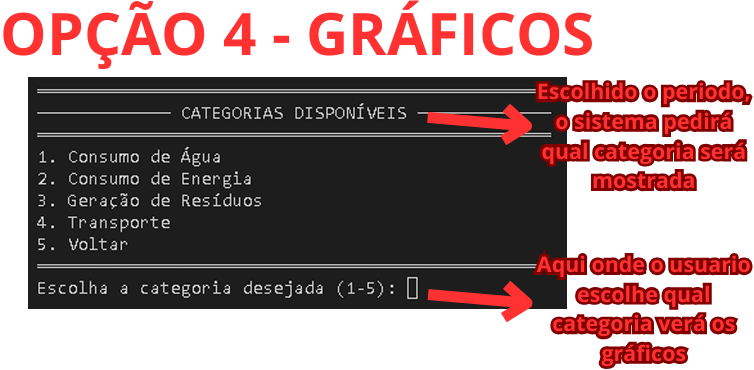
**Fluxo principal:**

|  |  |
| --- | --- |
| **Ações do Usuário** | **Ações do Sistema** |
| Acessar tela de boas práticas | Exibir menu de opções para boas práticas |
| Selecionar opção | Calcular classificação do usuário |
|  | Baseado na classificação e opção, exibir mensagens de acordo para o usuário |

**REQUISITO FUNCIONAL 6 –** Gráficos sobre o consumo

**Tela de Gráficos – Menu Periodo:**

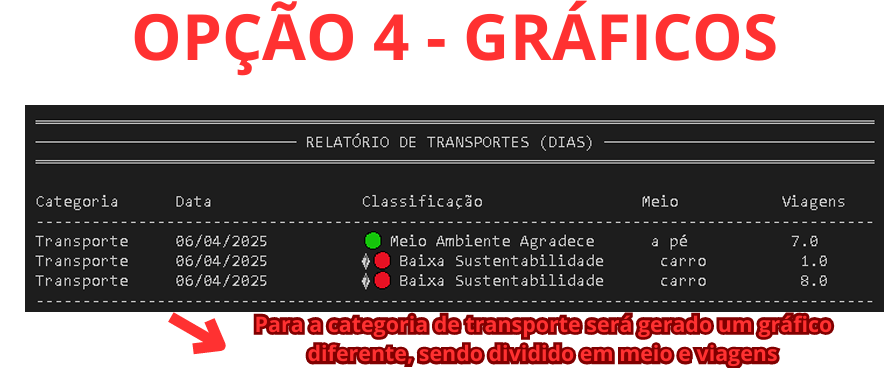
****

**Tela de Gráficos – Menu Categoria: **

**Tela de Gráficos – Gráficos Gerais:**

****

**Tela de Gráficos – Gráfico especifico para transporte:**

****

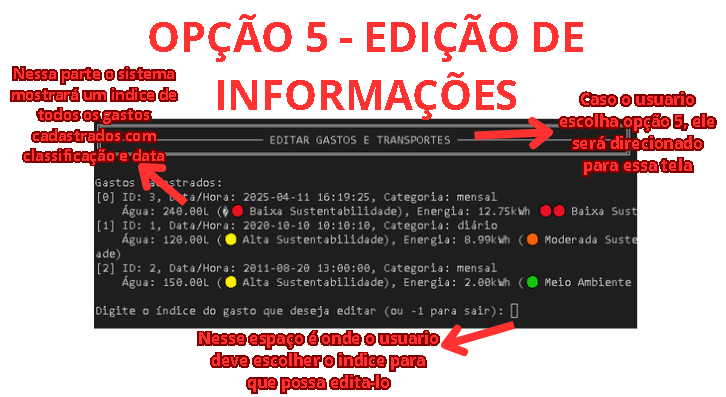
* **Pré-condições:** O usuário deve possuir dados armazenados no banco de dados
* **Validações:** O sistema deve exibir corretamente os gráficos de acordo com o período de tempo e categoria de consumo escolhidos.
* **Ator principal:** Usuário
* **Requisitos especiais:** Deve permitir alternância entre períodos e categorias de consumo

**Fluxo principal:**

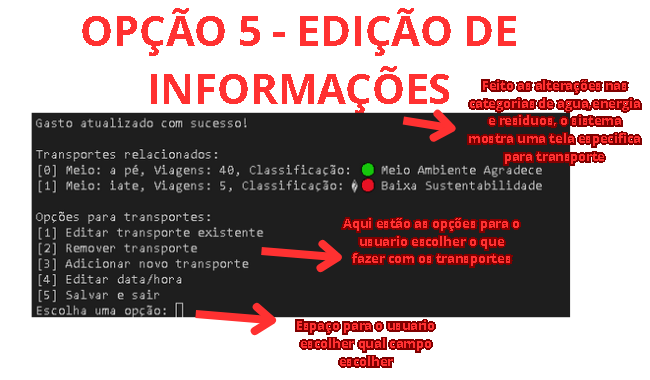
|  |  |
| --- | --- |
| **Ações do usuário** | **Ações do sistema** |
| Acessar a tela de gráficos | Exibir gráfico de barras do consumo |
| Selecionar categoria de consumo | Atualizar gráfico conforme a seleção |
| Alternar entre períodos (diário, semanal, mensal) | Atualizar gráfico conforme a seleção |
| Clicar em “voltar” | Retornar para a tela principal |

**REQUISITO FUNCIONAL 7 –** Edição de informações

**Tela de Edição de Informações - Geral:**



**Tela de Edição de Informações – Salvamento e especifico para transporte:**

****

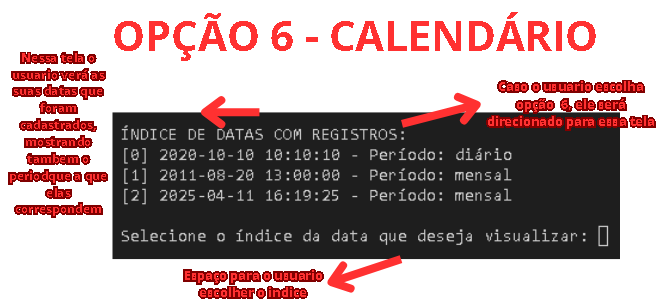
* **Pré-condições**: O usuário deve possuir dados armazenados no banco de dados
* **Validações**: O sistema exibir os dados cadastrados com a data especifica e período, mostrando o índice ao qual aquele registro pertence. E baseado na escolha, o sistema deve alterar no banco de dados
* **Ator principal**: Usuário
* **Requisitos especiais**: Deve permitir alternância entre períodos e categorias de consumo

**Fluxo principal:**

|  |  |
| --- | --- |
| **Ações do usuário** | **Ações do sistema** |
|  | Exibir índices com forme o banco de dados. Caso o usuário não possuir registro, não mostrar nada |
| Escolher um índice | Editar os dados e mandar ao banco de dados |
|  | Retorna dados editados com sucesso ou sem registro |
| Selecionar salvar os dados | Revisar os dados e salvar os dados editados |

**REQUISITO FUNCIONAL 8 –** Calendário

**Tela de Calendário – Tela inicial:**

****

**Tela de Calendário – Resultados:**

****

* **Pré-condições**: O usuário deve possuir dados armazenados no banco de dados
* **Validações**: O sistema deve exibir as datas e períodos que houve algum cadastros. E Logo abaixo mostrar quais dados foram registrados naquela data
* **Ator principal**: Usuário
* **Requisitos especiais**: Deve permitir alternância entre períodos e novas datas

|  |  |
| --- | --- |
| **Ações do usuário** | **Ações do sistema** |
|  | Exibir as datas em que houve cadastro e o período a que correspondem. Caso não haja registro retornar que não há cadastros |
| Selecionar qual data quer visualizar | Buscar ao banco de dados a data escolhida |
|  | Exibir um calendário com os dias que houveram registro em verde, e em vermelhos dias sem registro |
|  | Logo abaixo exibir quais informações foram colocadas |
| Escolher se deseja voltar a ver outro calendário ou não | Voltar ao início ou encerrar o sistema |

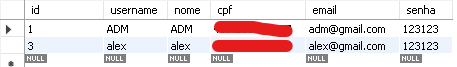
**REQUISITOS FUNCIONAIS 9 –** BackEnd (MySQL Workbench)

**Tables criadas:**

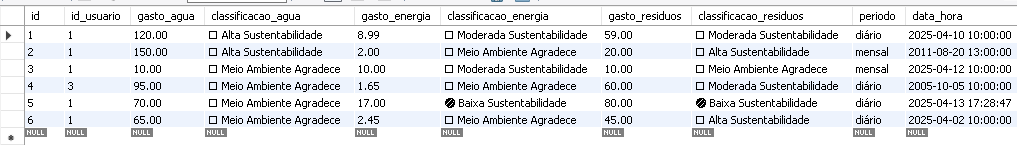
**Tables criado modelagem visual:**

****

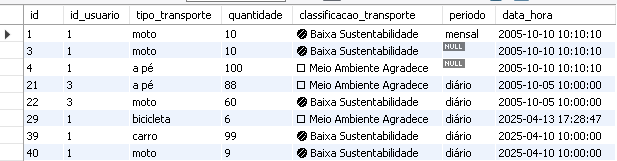
**Usuarios:**

****

**Gastos dos usuários:**

****

**Transportes dos usuários:**

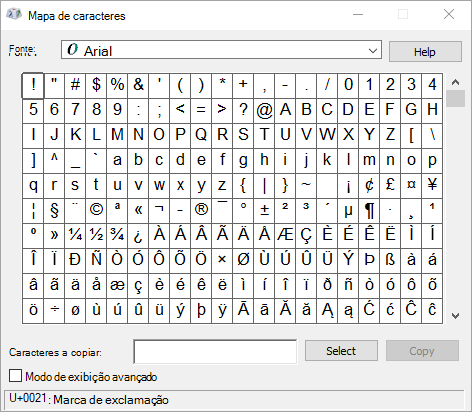
****

* **Pré-condições**: O banco de dados deve a todo momento responder ao FrontEnd, recebendo e enviando informações ao banco de dados, sem falha. Estar conectado a um banco de dados local ou rede
* **Validações**: O banco de dados deve estar devidamente ligado e operante. Para o usuário utilizar o banco de dados primeiramente ele deve fornecer dados de cadastro, e/ou fazer um cadastro ao sistema. Chaves únicas não podem se repetir, e são elas CPF, Email e username
* **Ator** **principal**: sistema

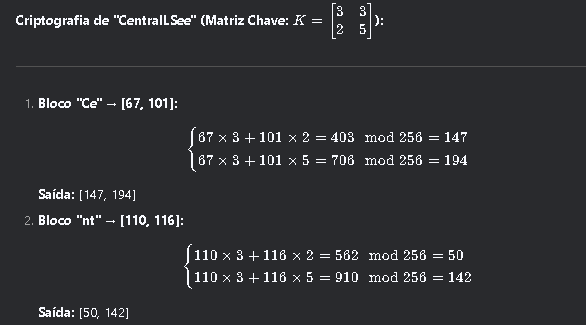
|  |  |
| --- | --- |
| **Ações do usuário** | **Ações do sistema** |
|  | Conectar o banco a uma rede local ou ethernet |
|  | Inserir table e unique keys |
| Fazer o cadastro para poder acessar o FrontEnd | Receber o cadastro. Caso informações já cadastradas, exibir erro |
|  | Criar um id próprio para aquele usuário |
| Utilização geral do frontend, seja pedindo ou colocando informação | Verificar se o usuário está cadastrado e fornecer as informações ao frontend |
|  | Editar as informações quando solicitado. Enviar informações quando solicitado. Conectar em tempo real tudo que ocorre no frontEnd |

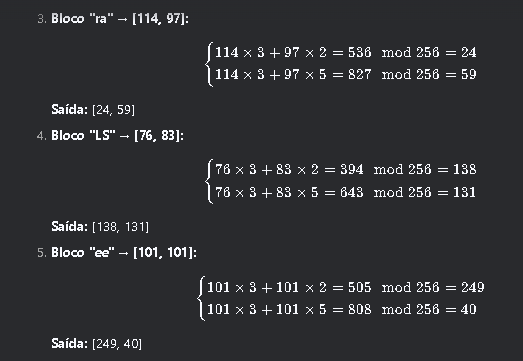
**REQUISITO FUNCIONAL 10 –** Criptografia

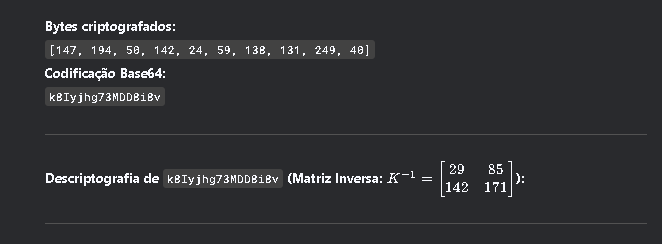
ASCII utilizado:

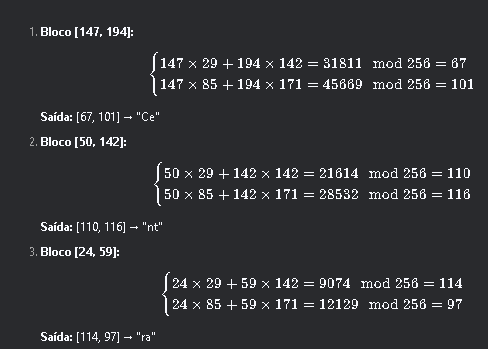


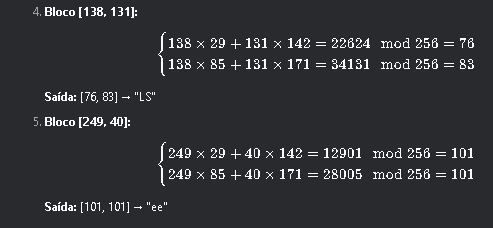
Exemplo de aplicação com senha de utilização: “CentralLSee”











* Pré-condições: O usuário deve criar e/ou fornecer uma senha para que seja criptografada
* Validações: O usuário deve seguir as orientações de próximas etapas, para seja enviado e salvo no sistema
* Ator principal: sistema

# REQUISITOS NÃO FUNCIONAIS

**DESEMPENHO**

* **Descrição**: O sistema deve ser capaz de processar as requisições dos usuários de forma rápida e eficiente.
* **Critérios de aceitação**:
  1. O sistema deve ser capaz de processar pelo menos 10 requisições simultâneas sem degradação no desempenho.
  2. A resposta do sistema para qualquer ação do usuário, como login ou exibição de gráficos, deve ser realizada em até 3 segundos.
  3. O tempo de carregamento das telas principais do sistema não deve ultrapassar 5 segundos.

**SEGURANÇA**

* **Descrição**: O sistema deve garantir a segurança dos dados dos usuários e do próprio sistema.
* **Critérios de aceitação**:
  1. A senha do usuário deve ser armazenada no banco de dados de forma criptografada.
  2. O sistema deve realizar verificações contra tentativas de ataques de força bruta no login, bloqueando o usuário após 3 tentativas falhas.

**USABILIDADE**

1. **Descrição**: O sistema deve ser intuitivo e fácil de usar, com uma interface amigável para os usuários.
2. **Critérios de aceitação**:
   1. A interface do usuário deve ser simples e objetiva, com todas as funcionalidades principais acessíveis em no máximo 3 cliques.
   2. O sistema deve apresentar mensagens de erro claras e concisas, explicando o que o usuário deve fazer para corrigir um erro.
   3. O sistema deve ter uma barra de navegação fixa e visível em todas as páginas principais.

# TESTES

Explicação do código, arquivo à arquivo.

**Arquivo Setup:**

Este código inicia um sistema com duas telas. Primeiro, executa o fluxo de login (tela1\_login). Se o login for bem-sucedido (usuario\_logado é válido), abre a tela principal de trabalho (tela3\_tela\_de\_trabalho), passando o usuário autenticado como parâmetro. Funciona como um controlador central, vinculando autenticação à interface principal do programa.

**Arquivo tela1\_login:**

Implementa uma tela de login de um sistema, com as seguintes funcionalidades principais:

1. Menu Inicial: Oferece opções de login (1) ou cadastro (2). Se o usuário escolher cadastro, redireciona para tela2\_registro.

2. Autenticação de Usuário:

* Verifica se o username existe no banco de dados (3 tentativas).
* Se encontrado, valida a senha (3 tentativas).

3. Criptografia:

* A senha armazenada no banco está em Base64 e criptografada com cifra de Hill.
* O código decodifica o Base64 e aplica a descriptografia para comparar com a senha digitada.

4. Tratamento de Erros:

* Mensagens claras para senha incorreta, usuário inexistente e erros internos (ex: formato inválido no banco).

5. Fluxo de Controle:

* Retorna o username se o login for bem-sucedido ou None em caso de falha/bloqueio.
* Limpa a tela em cada etapa para melhor experiência do usuário.

**Arquivo tela2\_registro:**

Este código implementa a tela de registro de um sistema, com as seguintes funcionalidades:

1. Validação de Dados:

* Username: mínimo 3 caracteres.
* E-mail: verificado via regex.
* CPF: validação matemática dos dígitos e formatação automática (000.000.000-00).
* Senha: mínimo 6 caracteres.

2. Criptografia:

* A senha é criptografada com Hill Cipher e codificada em Base64 antes de ser salva no banco de dados.

3. Interface Amigável:

* Confirmação dos dados antes do cadastro.
* Mensagens de erro/sucesso destacadas com cores e símbolos.

4. Integração com Banco de Dados:

* Usa a função criar\_usuario para inserir os dados validados e criptografados.
* Trata erros como dados duplicados (e-mail, CPF ou username já existentes).

5. Tratamento de Exceções:

* Captura interrupções do usuário (ex: Ctrl+C).
* Exibe mensagens claras para erros de criptografia ou falhas no banco.

6. Fluxo de Controle:

* Retorna True se o cadastro for bem-sucedido ou False em caso de falha/cancelamento.

**Arquivo tela3\_tela\_de\_trabalho:**

Aqui é a tela principal de trabalho do sistema, com as seguintes funcionalidades:

1. Menu de Opções:

* Permite escolher entre registrar dados (1), acessar histórico (2), relatórios (3), gráficos (4), editar dados (5), calendário (6), ou sair (7).

2. Registro de Dados Sustentáveis:

* Período: Diário, mensal ou anual.
* Consumo de Água/Energia: Validação numérica.

Transportes:

* Categorizados em 4 níveis de sustentabilidade (Eco, Sustentável, Moderado, Poluente) com base em listas pré-definidas.
* Registra quantidade de viagens por transporte.
* Resíduos Não Recicláveis: Validação percentual.

3. Classificação Automática:

* Define categorias (cores/emoji) para água, energia e resíduos com base em critérios pré-estabelecidos.

4. Integração com Banco de Dados:

* Salva dados de consumo (salvar\_gastos\_no\_mysql) e transportes (salvar\_transportes\_no\_mysql) no MySQL, vinculados ao ID do usuário.

5. Navegação para Outras Telas:

* Redireciona para funcionalidades como histórico (tela5), relatórios (tela4), gráficos (tela6), calendário (tela8) e edição de dados (tela7).

6. Interface Amigável:

* Exibe resumo dos dados registrados com cores e símbolos intuitivos.
* Mensagens de erro/sucesso durante o salvamento.

**Arquivo tela4\_relatório\_boas\_práticas:**

Este código implementa a tela de relatório de boas práticas do sistema, com as seguintes funcionalidades:

1. Carregamento de Dados:

* Busca no banco de dados os registros de consumo (água, energia, resíduos) e transportes do usuário, usando seu id\_usuario.

2. Análise de Classificações:

* Média Geral: Calcula a média das classificações (🔴, 🔴, 🔴, 🔴) convertidas em valores numéricos (4 a 1).
* Recomendações: Identifica a pior categoria (ex: água com 🔴) e exibe dicas aleatórias para melhorar.
* Boas Práticas: Destaca a melhor categoria (ex: energia com 🔴) e exibe mensagens motivacionais.

3. Feedback Visual:

* Média Geral: Classifica o desempenho em "alta" (≥3.5), "moderada" (≥2.5) ou "baixa" (<2.5), com textos explicativos.
* Interface formatada com bordas e emojis para melhor legibilidade.

4. Menu Interativo:

Opções para acessar:

1: Recomendações para áreas críticas.

2: Reconhecimento de práticas sustentáveis.

3: Média geral do usuário.

4: Sair do relatório.

**Arquivo tela5\_menu\_de\_opcoes\_para\_historico:**

Este código implementa a tela de histórico de dados do sistema, com as seguintes funcionalidades:

1. Menu de Históricos:

Permite escolher entre visualizar:

* Histórico de Água (1), Energia (2), Transporte (3), Resíduos (4), Todas as Categorias (5), ou Sair (6).

2. Carregamento de Dados:

* Busca registros do banco de dados (MySQL) usando o id\_usuario, incluindo consumo (água, energia, resíduos) e transportes.

3. Exibição em Tabela:

* Categorias Individuais: Mostra data/hora e classificação (🔴, 🔴, 🔴, 🔴) para cada categoria selecionada.
* Todas as Categorias: Exibe dados consolidados de todas as categorias em uma única tabela.
* Transportes: Evita duplicatas usando transportes\_exibidos (identifica por data e tipo).

4. Tratamento de Erros:

* Exibe "Nenhum dado encontrado" se não houver registros para a categoria ou usuário.

5. Interface Amigável:

* Tabelas formatadas com bordas e alinhamento de colunas.
* Limpeza automática da tela (limpar\_tela()) para melhor experiência.

**Arquivo tela6\_gráficos\_verticais:**

Este código implementa a tela de gráficos verticais do sistema, com as seguintes funcionalidades:

1. Seleção de Período e Categoria:

* Períodos: Últimos 7 dias (1), 4 semanas (2), 12 meses (3).
* Categorias: Água (1), Energia (2), Resíduos (3), Transportes (4).

2. Geração de Gráficos:

* Gráficos Textuais: Exibe consumo (água, energia, resíduos) em formato de barras verticais usando caracteres ().
* Tabela para Transportes: Lista detalhes como tipo, data, classificação e número de viagens.

3. Carregamento de Dados:

* Filtra registros do banco de dados (MySQL) com base no período selecionado.
* Agrupa dados por dia, semana ou mês usando cálculos de datas (datetime).

4. Normalização Dinâmica:

* Ajusta a escala do gráfico proporcionalmente ao maior valor encontrado (ex: divide o valor máximo em 10 níveis).

5. Interface Amigável:

* Bordas formatadas com caracteres (═, ─) e rótulos claros.
* Limpeza automática da tela (limpar\_tela()) para melhor navegação.

**Arquivo tela7\_teste:**

Este código é uma interface para editar gastos e transportes de um usuário, integrado a um banco de dados. Sobre suas funcionalidades:

1. Edição de Gastos:

* Lista gastos cadastrados (água, energia, resíduos) com classificações de sustentabilidade.
* Permite atualizar os valores dos gastos, recalculando automaticamente as classificações com base em limites pré-definidos.
* Atualiza os dados no banco via atualizar\_gastos\_no\_mysql.

2. Gestão de Transportes:

* Listagem: Exibe transportes associados à mesma data/hora do gasto selecionado.
* Edição: Permite alterar o meio de transporte ou a quantidade de viagens.
* Adição: Classifica automaticamente o transporte em categorias com base em listas extensas de termos.
* Remoção: Remove transportes da lista.
* Atualização de Data/Hora: Altera a data/hora tanto para gastos quanto para transportes no banco de dados.
* Salvamento: Atualiza os transportes no banco via atualizar\_transportes\_no\_mysql.

3. Classificações Automáticas:

* Para gastos, as classificações são baseadas em valores numéricos
* Para transportes, as classificações dependem de listas pré-definidas de meios.

4. Validações:

* Verifica entradas inválidas (índices fora do range, valores não numéricos).
* Mantém valores atuais se o usuário não inserir novos dados.

**Arquivo tela8\_calender:**

Este código é uma interface de calendário para visualizar registros de gastos e transportes de um usuário.

1. Calendário Interativo:

* Exibe um calendário no terminal com datas destacadas:
* Verde: Dias com registros de gastos/transportes.
* Vermelho: Dias sem registros.
* A formatação é baseada no período do registro selecionado (diário, mensal ou anual).

2. Consulta ao Banco de Dados:

* Busca registros de gastos (gastos\_usuarios) e transportes (transportes\_usuario) associados ao ID do usuário.
* Usa GROUP\_CONCAT para combinar múltiplos transportes em uma única entrada por data.

3. Gráfico de Registros:

Mostra detalhes dos registros selecionados:

* Data/hora, período (diário/mensal/anual).
* Valores de água, energia, resíduos.
* Lista de transportes utilizados.

4. Navegação:

* Permite ao usuário selecionar datas a partir de um índice.
* Oferece opção de visualizar outra data ou sair.

Fluxo:

1. O usuário seleciona uma data registrada a partir de uma lista.

2. O sistema gera um calendário para o mês/ano correspondente, destacando os dias com registros.

3. Exibe um calendário detalhado com todos os dados daquele período.

**Arquivo db\_manager:**

É um módulo de gerenciamento de banco de dados MySQL para um sistema de registro de usuários, gastos (água, energia, resíduos) e transportes. Este arquivo não possui função própria, sendo basicamente usado como banco de funções para integrar ao banco de dados. Suas funções são:

1. Conexão com o Banco:

* conectar\_db(): Estabelece conexão com o MySQL usando credenciais fixas (usuário, senha, banco "pi").

2. Operações CRUD:

* Usuários: Criação (criar\_usuario) e consulta (buscar\_usuario\_por\_id).
* Gastos: Registro, busca (buscar\_gastos\_usuario), atualização (atualizar\_gastos\_no\_mysql) e salvamento com classificações de sustentabilidade.
* Transportes: Registro, busca (buscar\_transportes\_usuario), atualização (atualizar\_transportes\_no\_mysql) e exclusão/inserção em lote.

3. Lógica de Persistência:

* As funções seguem um padrão: conectam ao banco, executam queries (INSERT, UPDATE, DELETE, SELECT), tratam erros e fecham conexões.
* Dados como data\_hora são registrados automaticamente (datetime.now()).
* Atualizações de transportes removem registros antigos antes de inserir novos para evitar duplicidades.

4. Classificações:

* Campos como classificacao\_agua indicam categorias de sustentabilidade, vinculadas a regras externas.

**Arquivo hill\_cipher\_logic:**

Implementa a Cifra de Hill, um algoritmo de criptografia. Este arquivo assim como arquivo db, não é um arquivo que tem uma função real, sendo este um arquivo de banco de funções para criptografar. Suas funcionalidades:

1. Configuração da Chave:

* Usa uma matriz chave 2x2 invertível módulo 256 (KEY\_MATRIX = [[3,3],[2,5]]).
* Calcula a inversa modular da matriz (KEY\_MATRIX\_INV) usando determinante e matriz adjunta, garantindo que o determinante tenha inverso módulo 256 (exigência da cifra).

2. Criptografia:

* Divide o texto em blocos de 2 caracteres (ASCII estendido, MODULUS = 256).
* Converte caracteres para números e aplica padding (caractere nulo) se o tamanho for ímpar.
* Multiplica cada bloco pela matriz chave e aplica módulo 256 para gerar o texto cifrado.

3. Descriptografia:

* Usa a inversa da matriz chave (KEY\_MATRIX\_INV) para reverter a criptografia.
* Remove o padding após a conversão dos números para caracteres.

4. Tratamento de Erros:

* Verifica se a matriz chave é invertível (caso contrário, o programa encerra).
* Lida com caracteres não imprimíveis (sugere-se uso de Base64 em versões futuras).

**Arquivo BC de PI:**

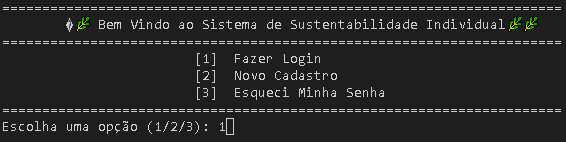
Este é o arquivo MySQL, responsável por criar uma banco de dados e todas as suas tables e banco de dados no MySQL Workbench, sendo o arquivo ao qual cria as tables e adiciona os níveis de primary key e unique, além de limitar seu varchar. Suas tables são usuário, gastos\_usuarios e transportes\_usuarios.

**Testes de telas**

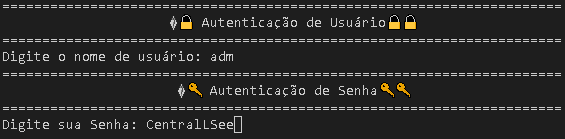
**Testes válidos:**

O usuário vai entrar nessa tela incial onde deve escolher o que fazer entre 1 (login), 2 (novo cadastro) e 3 (esqueci minha senha).

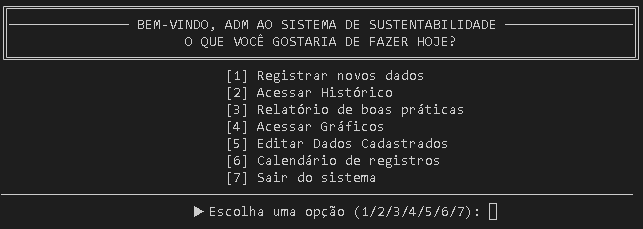
Para fazer login, o usuário já deve ter cadastro:



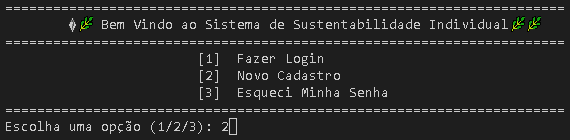
Digitar seu usuário e senha:



Tela de trabalho:



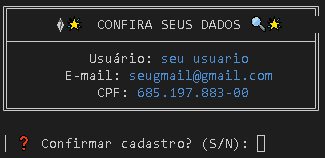
Para fazer registro:



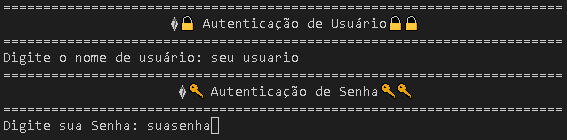
Quando o usuário selecionar a opção 2, ele deve cadastrar seu dados (nome, email, cpf e senha). O email deve ser válido e o CPF deve ser real, e a senha deve conter ao menos 6 caracteres.



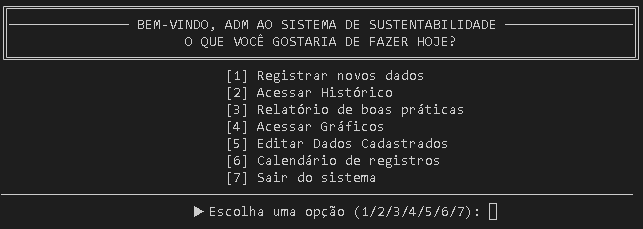
Após colocar seus dados, o usuário deve escolher se ‘S’ para salvar seu cadastro ou ‘N’ para não salvar seu cadastro:



Independente de escolher S ou N, o sistema voltará para essa tela, para que possa tentar acessar o sistema:



Tela de trabalho:



Caso o usuário selecione 1:

# METODOLOGIA APLICADA AO PROJETO

Para o desenvolvimento do projeto Sustentabilidade Individual, foi aplicada a Metodologia de Aprendizagem Baseada em Projetos (PBL), seguindo etapas estruturadas para garantir o alinhamento entre teoria, prática e colaboração em equipe. A metodologia foi dividida nas seguintes fases:

**Etapa 1: Introdução e Planejamento**

* Organização da equipe: Os integrantes foram divididos em um time de 5 membros, com papéis definidos (ex.: líder técnico, responsável por documentação, designer de interface).
* Definição do escopo: Apresentação do tema pelo orientador, discussão dos requisitos básicos (ODS 3 e 12, métricas de consumo, funcionalidades da plataforma) e alinhamento com o cronograma geral.
* Cronograma inicial: Elaboração de um plano com marcos como levantamento de requisitos, desenvolvimento do protótipo, implementação em Python, testes e apresentação final.
* Ferramentas adotadas: Definição de tecnologias (Python para backend, Canva para design de telas, Trello para gestão de tarefas) e padrões de documentação.

**Etapa 2: Coleta e Pesquisa**

* Referencial teórico: Pesquisa de artigos científicos, relatórios da ONU e dados nacionais (ex.: desperdício de água no Brasil, emissões de CO₂) para embasar as métricas de sustentabilidade.
* Benchmarking: Análise de plataformas similares para identificar boas práticas em UX e funcionalidades de diagnóstico personalizado.
* Definição de requisitos: Validação dos requisitos funcionais (cadastro, login, gráficos) e não funcionais (desempenho, segurança) por meio de discussões em equipe e feedback do orientador.
* Documentação: Elaboração do relatório inicial com escopo, tabelas de classificação e diagramas de telas.

**Etapa 3: Desenvolvimento**

* Prototipagem: Criação das telas no Canvas (login, cadastro, dashboards) e validação da usabilidade com testes preliminares.

Implementação técnica:

* Backend: Desenvolvimento em MySQL Workbench, utilizando o banco de dados local, ou o disponível pela universidade.
* Frontend: Construção da interface gráfica para exibição de métricas, gráficos e recomendações, totalmente feita em oyhton
* Integração contínua: Reuniões semanais com o orientador para apresentar avanços, ajustar requisitos e resolver desafios técnicos (ex.: validação de dados via fotos de hidrômetros).

**Etapa 4: Revisão e Iteração**

* Testes de usabilidade: Avaliação da interface com usuários reais para identificar pontos de melhoria (ex.: clareza nas mensagens de classificação).
* Ajustes técnicos: Correção de bugs no código, otimização do desempenho (tempo de resposta < 3 segundos) e reforço da segurança (criptografia de senhas).
* Validação das métricas: Revisão das tabelas de classificação (água, energia, resíduos) com base em feedback do orientador e dados atualizados da ONU.

**Etapa 5: Finalização e Publicação**

* Documentação final: Consolidação do relatório com todas as seções (justificativa, escopo, requisitos), inclusão de referências e formatação conforme normas da universidade.
* Testes finais: Verificação da integridade do sistema (ex.: fluxo de cadastro, exibição de gráficos em diferentes períodos) e validação dos critérios de aceitação.
* Preparação para apresentação: Elaboração de slides e demonstração ao vivo da plataforma, destacando funcionalidades-chave (ex.: recomendações personalizadas, histórico de consumo).

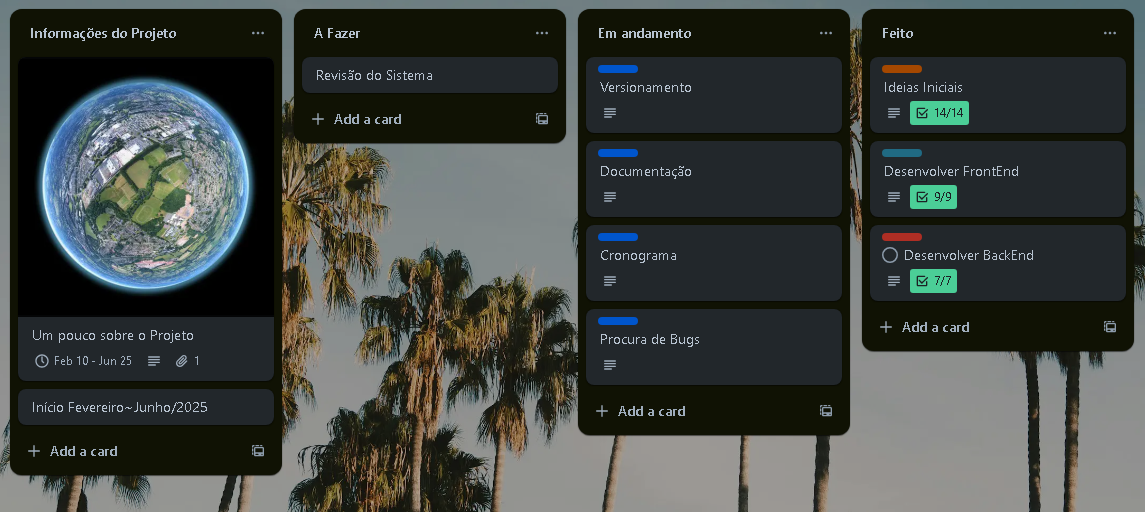
**Avaliação e Resultados**

* Critérios de avaliação: O projeto foi avaliado em etapas intermediárias (documentação, protótipos) e na apresentação final, considerando:
* Alinhamento com os ODS 3 e 12.
* Funcionalidade da plataforma.
* Qualidade técnica e usabilidade.

Resultado: Entrega de uma plataforma operacional, com capacidade de classificar hábitos sustentáveis e gerar impacto mensurável, conforme os objetivos propostos.

Esta metodologia permitiu integrar conhecimentos multidisciplinares (engenharia de software, sustentabilidade, design) e transformar conceitos abstratos em uma solução prática, alinhada às demandas globais de sustentabilidade.

1. **ACOMPANHAMENTO DA GESTÃO DO PROJETO**



Acesso Trello:

<https://trello.com/b/vnM2WO0h/sustentabilidade-individual-projeto-integrador>

Acesso GitHub:

HTTPS: <https://github.com/Alekkzsx/Projeto-Sustentabilidade-Individual-Classificatorio.git>

SSH: [git@github.com:Alekkzsx/Projeto-Sustentabilidade-Individual-Classificatorio.git](mailto:git@github.com:Alekkzsx/Projeto-Sustentabilidade-Individual-Classificatorio.git)

1. **PREMISSAS**

**Infraestrutura técnica**

* A equipe terá acesso a computadores com Python instalado e para desenvolvimento do backend e análise de dados.
* Será utilizado um banco de dados gratuito (MySQL Workbench) para armazenamento das informações dos usuários.
* Ferramentas de design (Canva) e gestão de tarefas (Trello).

**Acesso à internet**

* A equipe contará com conexão estável à internet para colaboração em ferramentas de versionamento (Git) e acesso a documentações técnicas.
* Os usuários finais necessitarão de internet para cadastro, login e sincronização de dados em tempo real.

**Dispositivos dos usuários**

* Os usuários possuirão dispositivos (computadores ou smartphones) com navegadores atualizados (Chrome, Firefox) para acessar a plataforma.
* Para envio de fotos de hidrômetros e medidores de energia, os usuários terão câmeras funcionais em seus dispositivos.

**Validade dos dados**

* As informações inseridas pelos usuários (ex.: consumo de água, energia) serão consideradas precisas, sem necessidade de verificação física adicional.
* As métricas de sustentabilidade (ODS 3 e 12) fornecidas pela ONU serão adotadas como base válida para os cálculos, sem revisão externa.

**Recursos humanos e conhecimento**

* A equipe possui conhecimento básico em Python, gestão de bancos de dados e design de interfaces para cumprir os requisitos técnicos.
* O orientador estará disponível para reuniões periódicas e validação de etapas críticas do projeto.

**Prazos e estabilidade de requisitos**

* O cronograma estabelecido será seguido sem atrasos significativos, considerando a disponibilidade contínua da equipe e do orientador.
* Não haverá alterações drásticas nos requisitos funcionais ou não funcionais após a fase de aprovação inicial.

**Conformidade legal**

* O armazenamento de dados pessoais (CPF, e-mail) seguirá a Lei Geral de Proteção de Dados (LGPD), sem necessidade de consultoria jurídica externa.

**Recursos institucionais**

* A universidade fornecerá acesso a laboratórios de informática, se necessário, para testes em ambientes controlados.
* Licenças gratuitas ou acadêmicas serão utilizadas para softwares de desenvolvimento (Python, Canva).

**Restrições de segurança**

* A criptografia de senhas e a proteção contra ataques de força bruta (bloqueio após 3 tentativas) serão implementadas conforme planejado, sem exigência de ferramentas pagas adicionais

# RESTRIÇÕES

* O projeto deverá ser concluído e entregue até Junho de 2025, conforme cronograma institucional. Atrasos na entrega de etapas intermediárias (ex.: protótipos, testes) impactarão a fase final de desenvolvimento.
* A ausência prolongada de um integrante da equipe por motivos de força maior (ex.: saúde, desistência) poderá exigir redistribuição de tarefas, revisão de escopo ou extensão de prazos, conforme aprovação do orientador.
* A precisão dos cálculos de sustentabilidade depende da qualidade dos dados inseridos pelos usuários (ex.: fotos de hidrômetros, medição manual de resíduos). Erros ou inconsistências nesses dados podem comprometer a classificação gerada pela plataforma.
* Não haverá integração com sistemas legados (ex.: bancos de dados de empresas de água ou energia), limitando-se aos dados inseridos manualmente pelos usuários.
* O projeto não possui verba para aquisição de softwares pagos, hardware adicional ou serviços premium. Soluções alternativas gratuitas serão priorizadas.

# REFERÊNCIAS

ORGANIZAÇÃO PAN-AMERICANA DA SAÚDE. Agenda para o Desenvolvimento Sustentável. Disponível em: https://www.paho.org/pt/topicos/agenda-para-desenvolvimento-sustentavel. Acesso em: 13 mar. 2025.

IBERDROLA. Estilo de vida sustentável. Disponível em: https://www.iberdrola.com/sustentabilidade/estilo-de-vida-sustentavel. Acesso em: 13 mar. 2025.

ABRELPE. Taxa de reciclagem de resíduos sólidos urbanos. 2023. Disponível em: https://www.abrema.org.br/2024/12/12/reciclagem-de-residuos-chega-a-8-no-pais-com-trabalho-informal-aponta-estudo/. Acesso em: 14 mar. 2025.

ONU MEIO AMBIENTE. Relatório sobre lacuna entre conhecimento e prática em sustentabilidade. 2021. Disponível em: https://www.unep.org/pt-br/resources/relatorio-sobre-lacuna-de-emissoes-2024. Acesso em: 14 mar. 2025.

SEEG. Emissões de CO₂ no setor de transporte. 2022. Disponível em: https://esginsights.com.br/transportes-respondem-por-11-das-emissoes-de-co2-no-brasil/. Acesso em: 15 mar. 2025.