

**UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"**

**FACULDADE DE CIÊNCIAS - CAMPUS BAURU**

**DEPARTAMENTO DE COMPUTAÇÃO**

**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**GABRIEL DADAMOS ROSSETTO**

**SOFTWARE PARA GESTÃO DE INVENTÁRIO DE CRÉDITO DE  
CARBONO VISANDO A COMPENSAÇÃO DA EMISSÃO DE  
GASES DE EFEITO ESTUFA**

**BAURU**

**Janeiro/2023**

GABRIEL DADAMOS ROSSETTO

**SOFTWARE PARA GESTÃO DE INVENTÁRIO DE CRÉDITO DE  
CARBONO VISANDO A COMPENSAÇÃO DA EMISSÃO DE  
GASES DE EFEITO ESTUFA**

Trabalho de Conclusão de Curso do Curso  
de Bacharelado em Ciência da Computação  
da Universidade Estadual Paulista “Júlio  
de Mesquita Filho”, Faculdade de Ciências,  
Campus Bauru.

Orientador: Prof. Kleber Rocha de Oliveira

R829s Rossetto, Gabriel Dadamos  
Software para gestão de inventário de crédito de carbono visando a  
compensação da emissão de gases de efeito estufa / Gabriel Dadamos  
Rossetto. -- Bauru, 2023  
52 p.  
  
Trabalho de conclusão de curso (Bacharelado - Ciência da  
Computação) - Universidade Estadual Paulista (Unesp), Faculdade de  
Ciências, Bauru  
Orientador: Kleber Rocha de Oliveira  
  
1. Crédito de carbono. 2. Gases de efeito estufa. 3. Aquecimento  
global. 4. Sustentabilidade. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de  
Ciências, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Gabriel Dadamos Rossetto

# **SOFTWARE PARA GESTÃO DE INVENTÁRIO DE CRÉDITO DE CARBONO VISANDO A COMPENSAÇÃO DA EMISSÃO DE GASES DE EFEITO ESTUFA**

Trabalho de Conclusão de Curso do Curso de  
Bacharelado em Ciência da Computação da Uni-  
versidade Estadual Paulista "Júlio de Mesquita  
Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

**Prof. Kleber Rocha de Oliveira**

Orientador

Departamento de Engenharia de Energia  
Faculdade de Engenharia e Ciências  
Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

**Prof<sup>a</sup>. Dr<sup>a</sup>. Simone das Graças Domingues  
Prado**

Departamento de Computação  
Faculdade de Ciências  
Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

**Prof. Dr. João Pedro Albino**

Departamento de Computação  
Faculdade de Ciências  
Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

Bauru, 17 de Janeiro de 2023.

*Dedico esta monografia à minha família e amigos por todo apoio que sempre me deram, e que com a ausência do mesmo, não haveria chegado até aqui.*

# Agradecimentos

Gostaria primeiramente de agradecer a Deus, que fez com que meus objetivos fossem alcançados, durante todos os meus anos de estudos.

Agradeço aos meus amigos e familiares, em especial aos meus pais e irmã, por todo o apoio e pela ajuda, que muito contribuíram para a realização deste trabalho.

Agradeço também aos professores, pelas correções e ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso. Dentre eles, fica um agradecimento em especial ao meu professor orientador Kleber Rocha de Oliveira, por ter me auxiliado muito com conselhos e direcionamentos referentes à este trabalho.

*Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination.*

Leo Cherne

# Resumo

A redução das emissões de gases com efeito de estufa é fundamental na luta contra as alterações climáticas, uma vez que estas as emissões contribuem para o aquecimento global e têm uma série de impactos negativos no meio ambiente e na saúde humana. Os créditos de carbono são uma ferramenta importante para alcançar reduções de emissões, pois fornecem um incentivo financeiro para que as empresas reduzam suas emissões, permitindo-lhes venderem créditos por cada tonelada de dióxido de carbono que reduzem ou removem da atmosfera. Ao usar créditos de carbono para recompensar pelas reduções de emissões, governos e empresas podem encorajar o desenvolvimento e implementação de tecnologias e práticas limpas, assim ajudando a reduzir o nível geral de emissões de gases do efeito estufa. Este projeto de *software* visa ajudar as empresas a rastrear e controlar suas emissões de gases do efeito estufa, a fim de cumprir os requisitos regulamentares e reduzir o seu impacto ambiental. Usando o método de linha de base, o *software* calcula o saldo de créditos de carbono aos quais uma empresa tem direito com base em suas reduções de emissões. Além disso, o *software* gera relatórios sobre o desempenho da empresa referente as emissões que podem ser submetidas aos órgãos reguladores. Ao fornecer uma interface fácil de usar para o rastreamento e geração de relatórios de emissões, este *software* visa facilitar com que empresas possam demonstrar seu compromisso com a sustentabilidade e cumprir suas metas de redução de emissões.

**Palavras-chave:** Crédito de Carbono, Gas do Efeito Estufa, Aquecimento Global, Sustentabilidade



# Abstract

Greenhouse gas emissions reductions are critical in the fight against climate change, as these emissions contribute to global warming and have a range of negative impacts on the environment and human health. Carbon credits are an important tool for achieving emissions reductions, as they provide a financial incentive for companies to reduce their emissions by allowing them to sell credits for every tonne of carbon dioxide they reduce or remove from the atmosphere. By using carbon credits to reward emissions reductions, governments and businesses can encourage the development and implementation of clean technologies and practices, ultimately helping to reduce the overall level of greenhouse gas emissions. This software project aims to help companies track and control their greenhouse gas emissions in order to meet regulatory requirements and reduce their environmental impact. Using the baseline method, the software calculates the carbon credits that a company is entitled to based on their emissions reductions. In addition, the software generates reports on the company's emissions which can be submitted to regulatory organizations. By providing an easy-to-use platform for tracking and reporting emissions, this software aims to make it easier for companies to demonstrate their commitment to sustainability and meet their emissions reduction goals.

**Keywords:** Carbon Credit, Greenhouse Gas, Global Warming, Sustainability

# Lista de figuras

|   |    |
|---|----|
| Figura 1 – Diagrama representando o fluxo do painel de controle. . . . .  | 29 |
| Figura 2 – Imagem da organização de pastas do painel de controle ao nível da aplicação. . . . .                                       | 31 |
| Figura 3 – Imagem da organização de pastas da api. . . . .  | 32 |
| Figura 4 – Chave de conexão disponibilizada para integração do banco de dados com a API utilizando o usuário ' <i>root</i> '. . . . . | 34 |
| Figura 5 – Página de Login. . . . .   | 43 |
| Figura 6 – Página de Registro. . . . .  | 43 |
| Figura 7 – Página de Saldo de Crédito de Carbono. . . . .   | 44 |
| Figura 8 – Página de Listagem de Emissões. . . . .  | 45 |
| Figura 9 – Página de Criação de Emissão. . . . .  | 46 |
| Figura 10 – Página de Edição de Emissão. . . . .  | 47 |
| Figura 11 – Página de Relatório de Emissões Emitido. . . . .  | 47 |

# Lista de códigos

|  |    |
|--|----|
| Código 1 – Exemplo de código de um componente de classe em React. . . . .  | 23 |
| Código 2 – Exemplo de código de um componente de função em React. . . . .  | 23 |
| Código 3 – Exemplo de utilização do componente <i>Button</i> fornecido pela biblioteca Material-UI. . . . .                      | 26 |
| Código 4 – Arquivo index.js com o setup inicial do servidor Express.js da API rodando na porta 3000. . . . .                     | 33 |
| Código 5 – Arquivo index.js agora com a integração com o banco de dados MongoDB. . . . .   | 34 |
| Código 6 – Modelo de dados de usuário. . . . .   | 35 |
| Código 7 – Modelo de dados das emissões de GEE. . . . .  | 36 |
| Código 8 – Rotas de autenticação de usuário. . . . .   | 37 |
| Código 9 – Rotas de registro de emissões de GEE definidas . . . . .  | 38 |
| Código 10 – Exemplos de gases adicionados na base de dados de GEE. . . . .   | 40 |
| Código 11 – Componente Navbar construído utilizando diversos componentes fornecidos pela biblioteca <i>Material-UI</i> . . . . . | 41 |
| Código 12 – Função simples que realiza o cálculo do saldo de créditos de carbono. . . . .  | 44 |
| Código 13 – Botão incluindo o componente <i>PDFDownloadLink</i> disponibilizado pelo pacote <i>React-pdf</i> . . . . .           | 44 |
| Código 14 – Função que realiza o cálculo de tCO2e de cada emissão listada. . . . .   | 45 |
| Código 15 – Definição das rotas de navegação do painel de controle. . . . .  | 46 |
| Código 16 – Integração do painel de controle com a API para a criação de uma nova emissão de GEE . . . . .                       | 48 |

# Lista de abreviaturas e siglas

|                    |  |
|--------------------|--|
| API                | <i>Application Programming Interface</i> |
| GEE                | <i>Gases do Efeito Estufa</i>            |
| GWP                | <i>Global Warming Potential</i>          |
| ID                 | <i>Identificator</i>                     |
| NPM                | <i>Node Package Manager</i>              |
| tCO <sub>2</sub> e | Toneladas de CO <sub>2</sub> equivalente |

# Sumário

|            |  |           |
|------------|--|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>                                  | <b>15</b> |
| <b>1.1</b> | <b>Problema</b>                                    | <b>15</b> |
| <b>1.2</b> | <b>Justificativa</b>                               | <b>16</b> |
| <b>1.3</b> | <b>Objetivos</b>                                   | <b>16</b> |
| <b>2</b>   | <b>FUNDAMENTAÇÃO TEÓRICA</b>                       | <b>17</b> |
| <b>2.1</b> | <b>Efeito Estufa</b>                               | <b>17</b> |
| 2.1.1      | IPCC <i>Fourth Assessment Report</i>               | 17        |
| 2.1.2      | Acordos Internacionais sobre Alterações Climáticas | 18        |
| 2.1.2.1    | UNFCCC   | 18        |
| 2.1.2.2    | Protocolo de Quioto                                | 19        |
| 2.1.2.3    | Protocolo de Paris                                 | 19        |
| 2.1.3      | Potencial de Aquecimento Global Relativo           | 19        |
| 2.1.4      | tCO <sub>2</sub> e                                 | 20        |
| 2.1.5      | Crédito de Carbono                                 | 20        |
| <b>2.2</b> | <b>Javascript</b>                                  | <b>21</b> |
| <b>2.3</b> | <b>React</b>                                       | <b>22</b> |
| 2.3.1      | Reconciliação                                      | 22        |
| 2.3.2      | Componentes  | 22        |
| <b>2.4</b> | <b>Node.JS</b>                                     | <b>23</b> |
| 2.4.1      | Express.js   | 23        |
| <b>2.5</b> | <b>API</b>   | <b>24</b> |
| <b>2.6</b> | <b>MongoDB</b>                                     | <b>24</b> |
| <b>2.7</b> | <b>NPM</b>   | <b>25</b> |
| <b>3</b>   | <b>METODOLOGIA</b>                                 | <b>26</b> |
| <b>3.1</b> | <b>Ferramentas utilizadas</b>                      | <b>26</b> |
| 3.1.1      | Create-React-App                                   | 26        |
| 3.1.2      | Material-UI  | 26        |
| 3.1.3      | WebStorm   | 27        |
| 3.1.4      | React-pdf  | 27        |
| 3.1.5      | React Router                                       | 28        |
| 3.1.6      | Mongoose   | 28        |
| <b>3.2</b> | <b>Aplicação</b>                                   | <b>28</b> |
| 3.2.1      | Painel de Controle                                 | 28        |
| 3.2.1.1    | Prototipagem do Fluxo                              | 29        |

|            |  |           |
|------------|--|-----------|
| 3.2.1.2    | Componentes Desenvolvidos . . . . .                                  | 29        |
| 3.2.1.3    | Estrutura de Pastas . . . . .  | 30        |
| 3.2.2      | API . . . . .  | 30        |
| 3.2.2.1    | Planejamento das Rotas . . . . .                                     | 30        |
| 3.2.2.2    | Estrutura de Pastas . . . . .  | 31        |
| <b>4</b>   | <b>DESENVOLVIMENTO . . . . .</b>                                     | <b>33</b> |
| <b>4.1</b> | <b>Desenvolvimento da API . . . . .</b>                              | <b>33</b> |
| 4.1.1      | Configuração Inicial do Servidor Utilizando Express.js . . . . .     | 33        |
| 4.1.2      | Configuração e Integração do Banco de Dados MongoDB . . . . .        | 34        |
| 4.1.3      | Definição dos Modelos de Dados . . . . .                             | 35        |
| 4.1.3.1    | Modelo de Dados de Usuário . . . . .                                 | 35        |
| 4.1.3.2    | Modelo de Dados das Emissões de GEE . . . . .                        | 36        |
| 4.1.4      | Desenvolvimento das Rotas de Autenticação . . . . .                  | 37        |
| 4.1.5      | Desenvolvimento das Rotas dos Registros de Emissões de GEE . . . . . | 38        |
| <b>4.2</b> | <b>Desenvolvimento do Painel de Controle . . . . .</b>               | <b>40</b> |
| 4.2.1      | Configuração Inicial do Projeto em React . . . . .                   | 40        |
| 4.2.2      | Instalação dos Pacotes Externos utilizados . . . . .                 | 40        |
| 4.2.3      | Construção da Base de Dados de GEE . . . . .                         | 40        |
| 4.2.4      | Desenvolvimento dos Componentes . . . . .                            | 41        |
| 4.2.5      | Desenvolvimento das Páginas . . . . .                                | 42        |
| 4.2.5.1    | Página de <i>Login</i> . . . . .                                     | 42        |
| 4.2.5.2    | Página de Registro. . . . .  | 43        |
| 4.2.5.3    | Página de Saldo de Crédito de Carbono . . . . .                      | 43        |
| 4.2.5.4    | Página de Listagem de Emissões . . . . .                             | 44        |
| 4.2.5.5    | Página de Criação de Emissão . . . . .                               | 46        |
| 4.2.5.6    | Página de Edição de Emissão . . . . .                                | 46        |
| 4.2.5.7    | Página de Relatório de Emissões Emitido . . . . .                    | 46        |
| 4.2.6      | Configuração das Rotas de Navegação . . . . .                        | 46        |
| 4.2.7      | Integração do Painel de Controle com a API . . . . .                 | 48        |
| <b>5</b>   | <b>CONCLUSÃO . . . . .</b>   | <b>50</b> |
| <b>5.1</b> | <b>Trabalhos Futuros . . . . .</b>                                   | <b>50</b> |
|            | <b>REFERÊNCIAS . . . . .</b>   | <b>51</b> |

# 1 Introdução

A cada dia que passa, fica cada vez mais evidente que o homem pode sim influenciar no clima do planeta, através da emissão de Gases do Efeito Estufa (GEE), geralmente emitidos em grande escala por indústrias de diversos setores. Os resultados do aumento drástico da emissão desses gases, principalmente após o início da Revolução Industrial, podem ser percebidos através das alterações climáticas, como por exemplo o aumento das temperaturas ao redor do planeta ano após ano.

Com o aumento da emissão desses gases e também das suas consequências, os governos de países de todo mundo voltaram a sua atenção para este tópico com tratados internacionais para redução das emissões destes gases, principalmente à partir do fim do século 20, com destaque para o Protocolo de Kyoto, assinado em 1997 no Japão, e mais recentemente com o Acordo de Paris, assinado em 2016 na França, e em vigência até hoje.

Estes tratados internacionais definiram objetivos com limites para a quantidade de gases estufa emitidos por cada país, com quotas definidas para as empresas, e também a criação de instrumentos para incentivar essas empresas a reduzirem as suas emissões desses gases, como o crédito de carbono. (GUPTA, 2011)

Um crédito de carbono equivale à permissão de emissão de uma tonelada de dióxido de carbono, ou uma quantidade correspondente de outros gases estufa. Com a criação do crédito de carbono, também surgiu a possibilidade da negociação do mesmo, para que caso uma empresa necessite emitir uma quantidade de gases estufa que exceda a sua quota permitida, possa adquirir créditos de carbono de uma empresa que tenha emitido menos gases que a sua quota permitida. (GUPTA, 2011)

Com o surgimento das regulamentações sobre a emissão de gases estufa introduzidas desde o fim do século passado, além do surgimento dos mercados de crédito de carbono, se tornou essencial para as empresas ter um controle preciso de suas emissões de gases para atender as exigências dos tratados internacionais. Este projeto visa fornecer de maneira flexível, esse controle sobre o registro das emissões de gases estufa dos diferentes setores de cada negócio, além de fornecer informações sobre o saldo de crédito de carbono baseando se nas reduções de emissões esperadas.

## 1.1 Problema

O problema que o presente trabalho visa resolver é a falta de um sistema eficiente para o cadastro e controle de emissões de gases do efeito estufa, bem como a falta de visibilidade sobre o saldo de crédito de carbono.

De acordo com o Relatório de Emissões de Gases de Efeito Estufa no Brasil, as emissões de gases de efeito estufa no país têm aumentado significativamente nas últimas décadas, principalmente devido ao crescimento da indústria e do setor de energia. Além disso, o relatório destaca a necessidade de um sistema eficiente de cadastro e controle dessas emissões, bem como de uma visão clara do saldo de crédito de carbono, para auxiliar na tomada de decisões informadas sobre como reduzir as emissões de gases de efeito estufa. (IBGE, 2016)

O crédito de carbono é um sistema de comércio de emissões que permite às empresas e governos compensar suas emissões de gases de efeito estufa através da compra de créditos de carbono de outras fontes que têm um saldo de crédito positivo. No entanto, sem um sistema de cadastro e controle eficiente, as empresas e governos podem encontrar dificuldades em manter uma visão clara do saldo de crédito de carbono e, conseqüentemente, em tomar decisões informadas sobre como reduzir suas emissões de gases de efeito estufa. (SILVA; ALMEIDA; SANTOS, 2018)

## 1.2 Justificativa

A necessidade de um sistema eficiente de cadastro e controle de emissões de gases do efeito estufa e de uma visão clara do saldo de crédito de carbono é cada vez mais evidente diante da crescente preocupação com o aquecimento global e seus impactos ambientais e sociais.

O *software* de cadastro e controle de emissões de gases de efeito estufa e visão facilitada sobre o saldo de crédito de carbono proposto neste trabalho, busca solucionar esses problemas, fornecendo uma ferramenta fácil de usar e acessível para monitorar e controlar as emissões de gases de efeito estufa, além de fornecer uma visão clara do saldo de crédito de carbono.

## 1.3 Objetivos

**Objetivo Geral:** Desenvolver um software de cadastro e controle de emissões de gases de efeito estufa e visão facilitada sobre o saldo de crédito de carbono.

**Objetivos Específicos:**

- Identificar as necessidades e expectativas dos usuários em relação a um *software* de cadastro e controle de emissões de gases de efeito estufa e visão facilitada sobre o saldo de crédito de carbono;
- Projetar e implementar um *software* de cadastro e controle de emissões de gases de efeito estufa e visão facilitada sobre o saldo de crédito de carbono;
- Realizar testes para garantir a funcionalidade e a qualidade do *software*; e
- Elaborar uma documentação técnica detalhada do projeto;



## 2 Fundamentação Teórica

A execução deste trabalho exige conhecimento de conceitos relacionados a emissões de GEE e de como essas emissões contribuem para as mudanças climáticas. Se faz também necessário entender o conceito de crédito de carbono e como ele é utilizado para incentivar a redução da emissão de GEE.

Também é essencial para a execução deste projeto, um entendimento técnico e teórico sobre as ferramentas a serem utilizadas no desenvolvimento do mesmo, bem como sobre os padrões de desenvolvimento utilizados.

### 2.1 Efeito Estufa

O efeito estufa é um fenômeno natural que ocorre na atmosfera da Terra. Ele é causado por GEE, como o dióxido de carbono ( $\text{CO}_2$ ), o metano ( $\text{CH}_4$ ) e o óxido nitroso ( $\text{N}_2\text{O}$ ), que retêm parte da radiação infravermelha emitida pelo planeta, impedindo que ela escape para o espaço. (IPCC, 2007)

A presença desses gases na atmosfera é essencial para manter a temperatura da Terra em um nível adequado para a vida. No entanto, a atividade humana, como a queima de combustíveis fósseis, a produção de carne e a degradação de florestas, tem contribuído para o aumento desses gases na atmosfera, intensificando o efeito estufa e levando ao aquecimento global.

De acordo com o *Intergovernmental Panel on Climate Change* (IPCC), ou Painel Intergovernamental sobre Mudanças Climáticas, o aumento da concentração de GEE na atmosfera provavelmente é causado pelas atividades humanas. O aumento da temperatura média global tem sido acompanhado por mudanças climáticas observadas em todo o mundo, como o derretimento das geleiras, o aumento do nível do mar e a intensificação de fenômenos meteorológicos extremos, como furacões e secas. (IPCC, 2007)

#### 2.1.1 IPCC *Fourth Assessment Report*

O IPCC *Fourth Assessment Report*, ou Quarto Relatório de Avaliação do Painel Intergovernamental sobre Mudanças Climáticas, é um relatório científico publicado em 2007 que fornece uma avaliação atualizada dos conhecimentos científicos sobre as mudanças climáticas. (IPCC, 2007)

O quarto relatório de avaliação conclui que é extremamente provável que o aumento da temperatura média global observado nas últimas décadas seja principalmente causado pelas

atividades humanas, especialmente pela queima de combustíveis fósseis e pelo aumento da concentração de GEE na atmosfera. (MASSON-DELMOTTE et al., 2018)

Os GEE, como CO<sub>2</sub>, CH<sub>4</sub> e o N<sub>2</sub>O, são substâncias presentes na atmosfera que retêm parte da radiação infravermelha emitida pela Terra, impedindo que ela escape para o espaço. O efeito estufa é um fenômeno natural que é essencial para manter a temperatura da Terra em um nível adequado para a vida. No entanto, a atividade humana tem contribuído para o aumento desses gases na atmosfera, intensificando o efeito estufa e levando ao aquecimento global. (BIERMANN; BOAS, 2014)

O quarto relatório de avaliação também destaca o papel das florestas e dos oceanos na regulação do clima, pois esses sistemas são capazes de absorver grandes quantidades de CO<sub>2</sub> da atmosfera. No entanto, o desmatamento e a degradação das florestas, assim como a acidificação dos oceanos devido ao aumento do CO<sub>2</sub> dissolvido, podem reduzir essa capacidade de absorção, intensificando ainda mais o aquecimento global. (IPCC, 2007)

## 2.1.2 Acordos Internacionais sobre Alterações Climáticas

Os acordos internacionais sobre alterações climáticas são tratados internacionais que visam estabelecer um quadro para a cooperação internacional na luta contra as mudanças climáticas causadas pelo aquecimento global. Alguns dos principais acordos nesse sentido incluem a *United Nations Framework Convention on Climate Change* (UNFCCC), o Protocolo de Quioto e o Acordo de Paris.

### 2.1.2.1 UNFCCC

A UNFCCC é um tratado internacional adotado em 1992 e entrando em vigor em 1994 com o objetivo de estabelecer um quadro para a cooperação internacional na redução das emissões de GEE e na adaptação às mudanças climáticas já ocorridas. A UNFCCC tem como signatários todos os países do mundo, incluindo os países em desenvolvimento e os países industrializados. (UNFCCC, s.d.)

A UNFCCC estabelece um quadro para a cooperação internacional na redução das emissões de GEE, mas não estabelece metas de redução específicas para os países. Em vez disso, a UNFCCC estabelece um processo para que os países desenvolvidos reportem suas emissões de GEE e os progressos realizados na redução dessas emissões. Além disso, a UNFCCC estabelece um mecanismo de financiamento para ajudar os países em desenvolvimento a implementar medidas de redução de emissões e de adaptação às mudanças climáticas.

A UNFCCC também estabelece o processo de Conferências das Partes (COP, sigla em inglês), que é uma reunião anual dos signatários da convenção onde são discutidos os progressos realizados na implementação da convenção e são tomadas decisões sobre medidas adicionais para enfrentar as mudanças climáticas.

### 2.1.2.2 Protocolo de Quioto

O Protocolo de Quioto é um tratado internacional adotado em 1997 com o objetivo de estabelecer metas de redução das emissões de GEE para os países desenvolvidos. O Protocolo de Quioto entrou em vigor em 2005 e foi ratificado por mais de 190 países. (BIERMANN; BOAS, 2014)

O Protocolo de Quioto estabeleceu metas de redução de emissões de GEE para 37 países industrializados e países em desenvolvimento com economias em transição, que juntos representam cerca de 55% das emissões globais de GEE. Essas metas de redução variam de país para país e são expressas em termos de redução percentual em relação aos níveis de emissões de 1990. Além disso, o Protocolo de Quioto estabeleceu o mecanismo de créditos de carbono, que permite aos países atingir suas metas de redução de emissões de GEE através da compra e venda de créditos de carbono emitidos pelo sistema de comércio de carbono aos países que realizam projetos de redução de emissões, como a implementação de tecnologias limpas ou a restauração de florestas. (BIERMANN; BOAS, 2014)

### 2.1.2.3 Protocolo de Paris

O Protocolo de Paris é um tratado internacional adotado em 2015 com o objetivo de estabelecer metas de redução das emissões de GEE para todos os países, com o objetivo de limitar o aumento da temperatura média global ao nível mais baixo possível, com um aumento máximo de 2°C em relação aos níveis pré-industriais. O Protocolo de Paris entrou em vigor em 2016 e foi assinado por mais de 190 países. (ROGELJ, 2016)

O Protocolo de Paris estabeleceu o compromisso de revisar e aumentar as metas de redução de emissões de GEE a cada cinco anos, a fim de garantir que os países estejam no caminho para atingir a meta final de zero emissões líquidas de GEE até o final do século. Além disso, o Protocolo de Paris incluiu o objetivo de promover a adaptação às mudanças climáticas já ocorridas e de apoiar o desenvolvimento de tecnologias limpas e a transferência de tecnologia para os países em desenvolvimento. (ROGELJ, 2016)

## 2.1.3 Potencial de Aquecimento Global Relativo

O *Global Warming Potential* (GWP), ou Potencial de Aquecimento Global Relativo, é uma medida usada para avaliar o impacto de diferentes GEE na temperatura da Terra. Ele é definido como o calor retido na atmosfera por um GEE em relação ao calor retido pelo CO<sub>2</sub> ao longo de um determinado período de tempo. (STOCKER et al., 2013)

O quarto relatório de avaliação do IPCC calculou o GWP de diferentes GEE usando um modelo de equilíbrio radiativo, que leva em consideração a absorção e emissão de radiação infravermelha pelos gases na atmosfera. O modelo também considerou os diferentes períodos

de tempo em que os gases ficam presentes na atmosfera, pois alguns gases têm um tempo de permanência mais longo do que outros. (IPCC, 2007)

De acordo com o quarto relatório de avaliação do IPCC, o GWP do CO<sub>2</sub> é de 1, enquanto o GWP do CH<sub>4</sub> é de 28, e o GWP do N<sub>2</sub>O é de 265. Isso significa que, por exemplo, a emissão de 1 tonelada de CH<sub>4</sub> na atmosfera tem o mesmo impacto no aquecimento global que a emissão de 28 toneladas de CO<sub>2</sub> durante um período de 100 anos. (IPCC, 2007)

#### 2.1.4 tCO<sub>2</sub>e

A tonelada de CO<sub>2</sub> equivalente (tCO<sub>2</sub>e) é uma medida usada para quantificar as emissões de GEE. Ela é utilizada para comparar as emissões de diferentes GEE, pois cada gas tem um potencial de aquecimento diferente. (IPCC, 2014)

O CO<sub>2</sub> é o principal gas responsável pelo aquecimento global, e é usado como referência para medir as emissões de outros gases. Assim, a tCO<sub>2</sub>e é a quantidade de CO<sub>2</sub> que teria o mesmo efeito de aquecimento que uma determinada quantidade de outro gas de efeito estufa.

Por exemplo, o CH<sub>4</sub> é um gas de efeito estufa que tem um efeito de aquecimento mais potente do que o CO<sub>2</sub>. Isso significa que uma pequena quantidade de CH<sub>4</sub> pode ter o mesmo efeito de aquecimento que uma grande quantidade de CO<sub>2</sub>. Por isso, a tCO<sub>2</sub>e é usada para comparar as emissões de CH<sub>4</sub> com as emissões de CO<sub>2</sub>. (IPCC, 2014)

A tCO<sub>2</sub>e é usada em diversos contextos, como na avaliação de projetos de redução de emissões, na regulamentação de emissões de GEE e na comunicação de informações sobre emissões para o público em geral.

#### 2.1.5 Crédito de Carbono

Os créditos de carbono surgiram como uma forma de promover a redução das emissões de GEE e de mitigar os impactos das mudanças climáticas. Eles foram desenvolvidos como parte do Protocolo de Quioto, um tratado internacional adotado em 1997 com o objetivo de estabelecer metas de redução de emissões de GEE para os países desenvolvidos. (FISCHER; TAYLOR, 2008)

O Protocolo de Quioto estabeleceu o Mecanismo de Desenvolvimento Limpo (MDL), que permite aos países desenvolvidos alcançarem suas metas de redução de emissões de GEE através da compra de créditos de carbono gerados por projetos de redução de emissões em países em desenvolvimento. Os créditos de carbono gerados pelo MDL são conhecidos como créditos de carbono "verdes" ou "MDL". (GOHARI, 2016)

Além do MDL, outros mecanismos de comércio de créditos de carbono foram desenvolvidos, como o Mecanismo de Flexibilidade do Protocolo de Quioto (MFQ), que permite aos países desenvolvidos alcançarem suas metas de redução de emissões de GEE através da

compra de créditos de carbono gerados por projetos de redução de emissões de GEE em países desenvolvidos. O MFQ inclui o Mecanismo Internacional de Atribuição de Créditos (ICAM), que permite aos países desenvolvidos adquirirem créditos de carbono de projetos de redução de emissões de GEE em outros países desenvolvidos. (GOHARI, 2016)

Os créditos de carbono também são negociados em mercados de carbono, como o Mercado de Emissões de GEE da União Europeia. Esses mercados permitem que empresas e governos comprem e vendam créditos de carbono, o que pode incentivar a redução de emissões de GEE de maneira econômica. (FISCHER; TAYLOR, 2008)

O cálculo dos créditos de carbono é baseado em modelos de avaliação de emissões de GEE, que comparam os níveis de emissões de um projeto com os níveis de emissões que seriam esperados sem a implementação do projeto, conhecidos como *baseline*. Por exemplo, se um projeto de energia limpa substitui uma usina térmica a carvão, os créditos de carbono são calculados com base na diferença entre as emissões de GEE da usina térmica a carvão e as emissões de GEE da usina limpa. (GOHARI, 2016)

A escolha do método *baseline* é um aspecto crítico no cálculo dos créditos de carbono, pois pode influenciar significativamente o tamanho da redução de emissões de GEE verificada e, conseqüentemente, o número de créditos de carbono emitidos. Por isso, é importante que o método *baseline* seja escolhido de maneira apropriada e justa, considerando fatores como as condições econômicas, tecnológicas e ambientais específicas de cada projeto. (FISCHER; TAYLOR, 2008)

O método *baseline* foi utilizado neste projeto para realizar o cálculo do saldo de créditos de carbono do usuário utilizando os registros de emissões de GEE inseridos através do *software* desenvolvido neste trabalho.

## 2.2 Javascript

Javascript é uma linguagem de programação de script de alto nível, interpretada e orientada a objetos. Ela foi criada em 1995 pelo programador americano Brendan Eich, enquanto trabalhava na Netscape Communications Corporation, e tornou-se uma das linguagens mais populares e utilizadas na *web*, sendo suportada por todos os principais navegadores. Ela foi originalmente criada para ser utilizada no lado do cliente em aplicações *web*, mas atualmente é utilizada em uma ampla variedade de aplicações, incluindo aplicativos de desktop e mobile. (HAVERBEKE, 2016)

Uma das principais vantagens do Javascript é sua capacidade de ser executada diretamente pelo navegador, o que elimina a necessidade de um servidor para processar os scripts. Isso permite que as aplicações *web* desenvolvidas com Javascript sejam rápidas e responsivas, fornecendo uma experiência de usuário agradável.

## 2.3 React

React é uma biblioteca Javascript de código aberto criada pelo Facebook para o desenvolvimento de interfaces de usuário. Ela utiliza uma abordagem baseada em componentes, permitindo que os desenvolvedores criem interfaces de usuário, dividindo-as em componentes menores e reutilizáveis. Isso torna mais fácil manter e atualizar aplicações, pois cada componente pode ser atualizado independentemente dos outros.

Uma das principais vantagens do React é sua capacidade de gerenciar e atualizar de forma eficiente os elementos da interfaces de usuário, o que é conhecido como "reconciliação". (CHINNATHAMBI, 2018)

Além disso, o React é compatível com outras bibliotecas e frameworks, como o Angular e o Vue.js, o que o torna uma opção versátil para o desenvolvimento de aplicações *web*.

### 2.3.1 Reconciliação

A reconciliação é um processo central no funcionamento do React. Ela se refere à maneira como a biblioteca gerencia e atualiza os elementos da interface de usuário em uma aplicação. Quando os dados de uma aplicação são alterados, o React comparará a representação atual da interface de usuário com a representação desejada e atualizará apenas os elementos que precisam ser alterados. Isso é possível graças à utilização de componentes em React, que são blocos de código reutilizáveis que representam elementos da interface de usuário.

A reconciliação é importante porque permite que as aplicações React tenham um desempenho eficiente, mesmo em grandes escala. Ao invés de atualizar toda a interface de usuário toda vez que os dados são alterados, o React atualiza apenas os componentes afetados, o que pode levar a um ganho significativo de desempenho em aplicações de grande escala. (CHINNATHAMBI, 2018)

### 2.3.2 Componentes

Os componentes são um dos principais conceitos em React. Eles são blocos de código reutilizáveis que representam elementos da interface de usuário em uma aplicação. Os componentes podem ser pensados como pequenas máquinas de estado que recebem entradas (através de propriedades) e produzem saídas (através de seus elementos filhos). Isso permite que os componentes sejam facilmente compostos para criar interfaces de usuário mais complexas.

Existem dois tipos principais de componentes em React: componentes de classe e componentes de função, cujos exemplos podem ser vistos nos Códigos 1 e 2 respectivamente. Os componentes de classe são implementados como classes Javascript e possuem um estado próprio, enquanto os componentes de função são implementados como funções Javascript e não possuem estado próprio.

Os componentes também são importantes porque permitem a reutilização de código e a manutenção mais fácil de aplicações. Ao invés de ter que reescrever código sempre que um elemento da interface de usuário precisar ser utilizado em vários lugares, os componentes podem ser reutilizados e atualizados independentemente uns dos outros.

Código 1 – Exemplo de código de um componente de classe em React.

```
1 class Welcome extends React.Component {  
2   render() {  
3     return <h1>Hello, {this.props.name}</h1>;  
4   }  
5 }
```

Código 2 – Exemplo de código de um componente de função em React.

```
1 function Welcome(props) {  
2   return <h1>Hello, {props.name}</h1>;  
3 }
```

## 2.4 Node.JS

Node.js é um ambiente de execução Javascript baseado em código aberto que permite a execução de código Javascript fora do navegador. Ele foi criado com o objetivo de permitir o desenvolvimento de aplicações de rede escaláveis e de alta performance utilizando Javascript.

Node.js funciona através da utilização de um loop de eventos que permite a execução assíncrona de código. Isso significa que, ao invés de esperar por uma tarefa ser concluída antes de prosseguir com a próxima, o Node.js pode tratar várias solicitações ao mesmo tempo, o que pode levar a um desempenho melhor em aplicações de grande escala. ([SEREDENKO, 2018](#))

Além disso, o Node.js vem com uma ampla variedade de módulos pré-construídos, que podem ser facilmente utilizados para adicionar funcionalidades adicionais às aplicações. Esses módulos incluem tudo, desde o gerenciamento de arquivos e criação de servidores até integração com banco de dados e autenticação de usuário.

### 2.4.1 Express.js

Express.js é uma biblioteca para aplicativos da *web* para Node.js que fornece uma série de recursos para facilitar o desenvolvimento de APIs.

O Express.js é particularmente útil para a construção de APIs pois ele fornece uma série de recursos para lidar com rotas, *middlewares* e tratamento de erros. As rotas são os endereços URL que são mapeados para determinadas ações em uma aplicação, enquanto os *middlewares* são funções que são executadas em uma requisição antes de chegar à rota final.

O tratamento de erros permite que os desenvolvedores lidem de forma eficiente com problemas que possam ocorrer durante o processamento de uma requisição.

O Express.js também é compatível com vários tipos de bancos de dados, o que o torna uma opção versátil para o desenvolvimento de APIs que precisam armazenar e recuperar dados.

## 2.5 API

API (*Application Programming Interface*) é um conjunto de rotinas, protocolos e ferramentas para construir software e permitir que diferentes sistemas se comuniquem entre si. As APIs são amplamente utilizadas para permitir que diferentes aplicações acessem dados e funcionalidades de outras aplicações de forma controlada e segura.

Existem vários tipos de APIs, incluindo APIs de sistema operacional, APIs de aplicativo e APIs da *web*. As APIs de sistema operacional permitem que os aplicativos acessem funcionalidades do sistema operacional, enquanto as APIs de aplicativo permitem que os aplicativos acessem funcionalidades de outros aplicativos. As APIs da *web* são APIs que são acessadas através da internet e geralmente são expostas através de uma URL. (LAURET, 2019)

As APIs são importantes porque permitem que diferentes aplicações se integrem e compartilhem dados e funcionalidades de forma controlada. Isso pode levar a uma ampla variedade de possibilidades, como a criação de aplicativos mais poderosos e a integração de diferentes sistemas e plataformas.

## 2.6 MongoDB

MongoDB é um banco de dados NoSQL de código aberto que armazena dados em formato de documento. Em vez de utilizar tabelas como em bancos de dados relacionais, o MongoDB armazena os dados em documentos JSON (*Javascript Object Notation*) similares, o que permite que os dados sejam representados de forma mais flexível e natural.

Um dos principais benefícios do MongoDB é sua escalabilidade. Ele foi projetado para ser executado em clusters de múltiplos servidores, o que permite que ele gerencie grandes quantidades de dados de forma eficiente. Além disso, o MongoDB oferece uma série de recursos de replicação e agrupamento que permitem que os dados sejam distribuídos entre vários servidores e mantidos de forma segura. (BRADSHAW; BRAZIL; CHODOROW, 2019)

O MongoDB também é conhecido por sua facilidade de uso. Ele possui uma linguagem de consulta simples e intuitiva, chamada de *MongoDB Query Language* (MQL), que permite que os desenvolvedores consultem e manipulem os dados de forma rápida e eficiente.



## 2.7 NPM

NPM (*Node Package Manager*) é o gerenciador de pacotes padrão para o ambiente de desenvolvimento Javascript. Ele permite que os desenvolvedores instalem, atualizem e gerenciem bibliotecas de código e ferramentas de terceiros de forma rápida e eficiente.

O NPM possui uma enorme coleção de pacotes disponíveis em seu repositório online, o que permite que os desenvolvedores encontrem e utilizem as bibliotecas e ferramentas de que precisam de forma rápida. Além disso, o NPM fornece uma série de recursos para gerenciar as dependências de um projeto, o que permite que os desenvolvedores mantenham o código atualizado e evitem conflitos de versão.

O NPM é um componente fundamental da plataforma Node.js e é amplamente utilizado em aplicativos *web* e em outros tipos de aplicativos.

## 3 Metodologia

Neste capítulo será apresentada a metodologia utilizada para o desenvolvimento da aplicação, que consiste em um painel de controle desenvolvido na biblioteca React, que será integrado com uma API programada utilizando Node.JS e Express.js, e que se comunicará com um banco de dados MongoDB.

### 3.1 Ferramentas utilizadas

Para o desenvolvimento do painel de controle e da API, foram utilizadas uma série de ferramentas como IDEs, bibliotecas de componentes para React e bibliotecas adicionais para Javascript.

#### 3.1.1 Create-React-App

Create-React-App é um pacote de configuração criado pelo Facebook para ajudar os desenvolvedores a começar a desenvolver aplicativos React de forma rápida e fácil. Ele fornece uma estrutura de projeto padrão, juntamente com scripts de configuração, permitindo que os desenvolvedores se concentrem no desenvolvimento da lógica do aplicativo sem se preocupar com a configuração do ambiente de desenvolvimento. O Create-React-App foi utilizado para gerar o *template* inicial do projeto do painel de controle em React.

#### 3.1.2 Material-UI

Material-UI é uma biblioteca de componentes para interface de usuário para React. Ela segue o *design system* do Google Material Design, um conjunto de diretrizes de design que visam criar interfaces de usuário consistentes e intuitivas.

A biblioteca Material-UI fornece uma ampla variedade de componentes pré-construídos, incluindo botões (como pode ser visto no Código 3), caixas de seleção, menus e muito mais. Esses componentes são projetados para serem fáceis de usar e personalizar, permitindo que os desenvolvedores criem interfaces consistentes e atraentes de forma rápida e eficiente.

Além disso, a biblioteca Material-UI é compatível com diferentes tamanhos de tela e dispositivos móveis, o que a torna uma opção versátil para o desenvolvimento de aplicações *web*.

Diversos componentes visuais básicos da biblioteca Material-UI foram utilizados ao longo do desenvolvimento do painel de controle deste trabalho.

Código 3 – Exemplo de utilização do componente *Button* fornecido pela biblioteca Material-UI.

```
1 import * as React from 'react';
2 import Button from '@mui/material/Button';
3
4 export default function MyApp() {
5   return (
6     <div>
7       <Button variant="contained">Hello World</Button>
8     </div>
9   );
10 }
```

### 3.1.3 WebStorm

WebStorm é uma *Integrated Development Environment* (IDE) para o desenvolvimento de aplicações *web*. Ela é desenvolvida pela JetBrains e é especialmente popular entre os desenvolvedores de JavaScript, HTML e CSS.

A WebStorm fornece uma série de recursos para ajudar os desenvolvedores a escrever, depurar e testar código de forma mais rápida e eficiente. Isso inclui funcionalidades como auto-completar, refatoração de código, depurador integrado e integração com sistemas de controle de versão. Além disso, a WebStorm é compatível com uma ampla variedade de *frameworks* e bibliotecas JavaScript, como React, Angular e Vue.js.

A WebStorm também é conhecida por sua interface intuitiva e personalizável, o que permite que os desenvolvedores configurem a IDE de acordo com suas necessidades e preferências pessoais.

A IDE WebStorm foi utilizada no desenvolvimento do painel de controle e da API desta aplicação.

### 3.1.4 React-pdf

O pacote *react-pdf* é um componente React para a renderização de documentos PDF. Ele permite que os desenvolvedores criem e exibam documentos PDF dinamicamente em aplicações React, permitindo que os usuários vejam, imprimam e façam download dos documentos.

O *react-pdf* é baseado na biblioteca *PDF.js*, que é uma biblioteca de código aberto para exibir documentos PDF no navegador. Ele utiliza a tecnologia de *canvas* do HTML5 para renderizar os documentos PDF e fornece uma série de recursos para navegar e interagir com os documentos, incluindo zoom, rolagem e seleção de texto.

Neste projeto, o react-pdf foi utilizado para a geração dos relatórios de emissões de gases do efeito estufa.

### 3.1.5 React Router

React Router é um pacote que fornece uma solução para gerenciar o roteamento em aplicações React. Ele permite que os desenvolvedores criem rotas e links em suas aplicações, permitindo que os usuários naveguem entre diferentes páginas e componentes de forma fluida.

O React Router é baseado em componentes, o que significa que os desenvolvedores podem inserir os componentes de roteamento em qualquer lugar da sua aplicação. Isso permite que os desenvolvedores criem interfaces de usuário avançadas, como menus de navegação e guias, sem precisar lidar com os detalhes do roteamento por trás das cenas.

Neste projeto, o React Router foi utilizado para o gerenciamento das rotas do painel de controle.

### 3.1.6 Mongoose

Mongoose é um pacote para aplicações Node.js que fornece uma solução para modelagem de dados em aplicações que utilizam o banco de dados MongoDB. Ele permite que desenvolvedores criem esquemas de dados e realizem operações de CRUD (create, read, update, delete) de forma rápida e eficiente.

O Mongoose fornece uma série de recursos para ajudar os desenvolvedores a trabalhar com os dados armazenados no MongoDB. Isso inclui validação de dados, gerenciamento de relacionamentos entre documentos e suporte a middlewares. Além disso, o Mongoose oferece uma linguagem de consulta fluente e intuitiva, que permite que os desenvolvedores consultem e manipulem os dados de forma rápida e eficiente.

No desenvolvimento deste trabalho, o Mongoose foi utilizado para conectar a API em um banco de dados MongoDB.

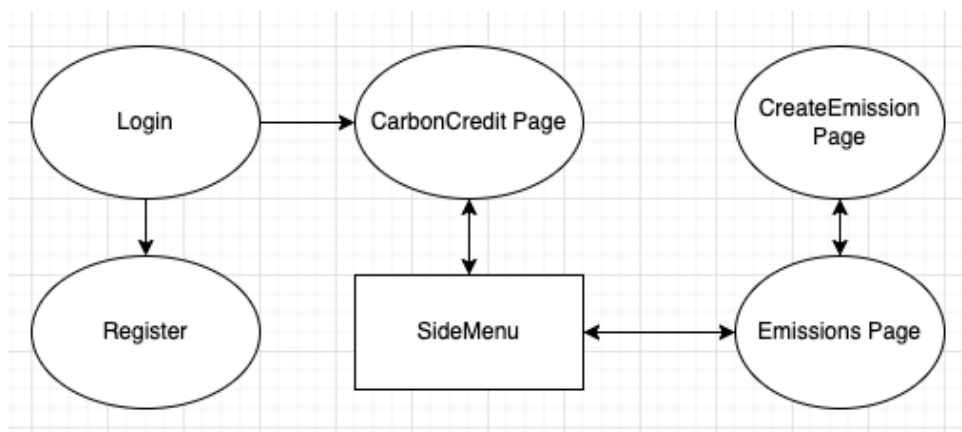
## 3.2 Aplicação

A aplicação desenvolvida neste trabalho consiste em um painel de controle desenvolvido em React e uma API desenvolvida em Node.JS juntamente com a biblioteca Express.js.

### 3.2.1 Painel de Controle

O painel de controle desta aplicação fornece ao usuário a funcionalidade de registro e visualização de emissões de GEE realizadas, além da emissão de relatórios referentes à essas

Figura 1 – Diagrama representando o fluxo do painel de controle.



Fonte: Elaborada pelo autor.

emissões. Também é disponibilizado ao usuário uma visão facilitada sobre seu saldo de crédito de carbono baseado nas emissões e na *baseline* do mesmo.

### 3.2.1.1 Prototipagem do Fluxo

O fluxo do painel de controle foi desenhado para ser o mais simples e intuitivo possível, possuindo telas de login e registro para a autenticação do usuário, sendo que após o login do usuário, o mesmo é levado diretamente à tela CarbonCredit, que possui a visão geral com os valores total de emissões e saldo atual do crédito de carbono.

Após ser autenticado, o usuário sempre tem acesso a um menu lateral, que pode levá-lo para a tela CarbonCredit, e também para a tela Emissions, onde o usuário tem uma visão em lista das emissões de GEE registradas, além de permitir que o usuário crie um novo registro de emissão ao levá-lo para a tela CreateEmission. O fluxo de navegação do painel de controle pode ser visto na Figura 1.

### 3.2.1.2 Componentes Desenvolvidos

Para a construção da interface de usuário, foram desenvolvidos diversos componentes em React. Em seguida se encontra uma breve explicação da função de cada componente construído para esta aplicação:

- *AuthContainer*: Um componente que serve como "pai" e traz um estilo similar às páginas de login e registro.
- *DashboardContainer*: Um componente pai para todas as telas do painel de controle após a autenticação, e que têm incluso em si os componentes Navbar e SideMenu
- *Navbar*: Um componente que exibe uma barra de navegação no topo da tela, e que possui o botão que exibe o componente SideMenu, e também o nome da aplicação.

- SideMenu: Um componente que pode ser aberto clicando no botão de menu presente no componente Navbar, e que exibe a lista de páginas da aplicação, levando o usuário até a página escolhida através do clique.
- Error: Componente que renderiza uma mensagem de erro caso o usuário tente acessar uma rota inexistente.
- PDFReport: Componente criado utilizando a biblioteca react-pdf, e que cuida da geração dos relatórios de emissões de GEE.
- TableComponents: Conjunto de componentes simples que formam as tabelas presentes exclusivamente no relatório gerado em PDF. Esses componentes se tornaram necessários para a exibição de tabelas utilizando o pacote react-pdf.

### 3.2.1.3 Estrutura de Pastas

Existem diversas maneiras de se organizar pastas quando tratamos de uma aplicação em React, e cada uma delas têm seus pontos positivos e negativos. Para este projeto, foi escolhida uma organização bem simplista, colocando os componentes visuais e sem regras de negócio dentro da pasta *components*, enquanto os componentes de cada página, e que possuem regras de negócio, foram colocados dentro da pasta *pages*. Foi criada também a pasta *mocks*, que contém arquivos com dados para teste e dados dos GEE disponibilizados no *software*. Presente também está a pasta *utils*, onde foram colocados arquivos com funções Javascript reutilizáveis que realizam cálculos ou formatações de dados. A organização de pastas do projeto do painel de controle pode ser observada na Figura 2.

## 3.2.2 API

A API desenvolvida neste trabalho tem como objetivo criar a conexão entre o painel de controle e o banco de dados MongoDB. Com ela, é possível realizar as operações básicas de CRUD (criação, leitura, atualização e exclusão) nos documentos do MongoDB de forma rápida e simples.

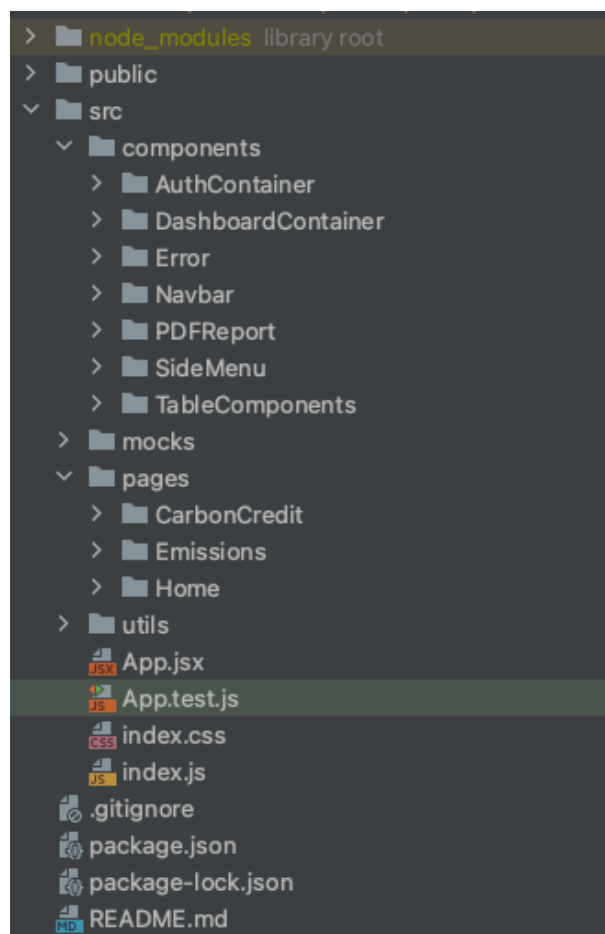
### 3.2.2.1 Planejamento das Rotas

Para esta API, foram criadas diversas rotas com funções de autenticação de usuário e para operações com os registros de emissões.

Em seguida se encontra uma breve descrição da função de cada rota:

- POST */register*: Esta rota cria um novo documento no banco de dados com os dados enviados no corpo da requisição para registrar um novo usuário.
- POST */login*: Esta rota tem como função autenticar o usuário com base nos dados enviados no corpo da requisição.

Figura 2 – Imagem da organização de pastas do painel de controle ao nível da aplicação.



Fonte: Elaborada pelo autor.

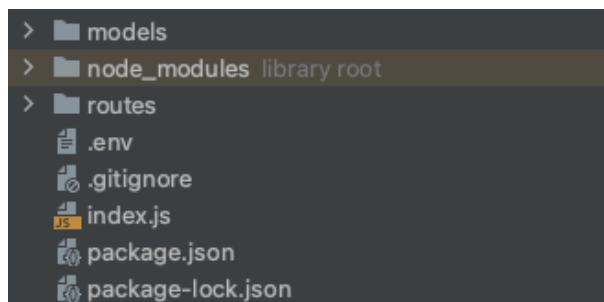
- POST */createEmission*: Esta rota cria um novo registro de emissão no banco de dados com os dados enviados no corpo da requisição.
- GET */getAll/:userId*: Esta rota retorna todos os registros de emissão cadastrados pelo usuário.
- GET */getById/:id*: Esta rota retorna o registro de emissão correspondente ao id contido na rota.
- PATCH */update/:id*: Esta rota atualiza o registro de emissão correspondente ao id contido na rota com os dados presentes no corpo da requisição.
- GET */delete/:id*: Esta rota apaga o registro de emissão correspondente ao id contido na rota.

### 3.2.2.2 Estrutura de Pastas

Para a organização de pastas da API, foi necessária a criação de duas pastas para manter um bom nível organizacional do projeto. A pasta *models* contém os arquivos de definição dos modelos dos documentos arquivados no banco de dados. Já a pasta *routes*, contém os arquivos

com declarações de rotas disponibilizadas pela API. A Figura 3 demonstra a organização de pastas do projeto da API.

Figura 3 – Imagem da organização de pastas da api.



Fonte: Elaborada pelo autor.



## 4 Desenvolvimento

### 4.1 Desenvolvimento da API

Para o desenvolvimento da API desta aplicação, foi necessário realizar a configuração inicial do servidor com o pacote Express.js, criar e integrar um banco de dados MongoDB, e por fim configurar e desenvolver as rotas a serem disponibilizadas ao painel de controle referentes à autenticação de usuário e registros de emissões de GEE.

#### 4.1.1 Configuração Inicial do Servidor Utilizando Express.js

Após a criação da pasta do projeto da API, foi necessário acessar a mesma pasta utilizando o terminal, e executar os seguintes comandos:

- *'npm init'*: Realiza as configurações iniciais do NPM no projeto. Adiciona o arquivo *package.json*, responsável pelas configurações de todos os pacotes externos a serem utilizados no projeto, além de informações sobre versionamento e configurações de eventuais *scripts* a serem definidos.
- *'npm install express'*: Instala o pacote Express.js.

Após a inicialização do projeto e a instalação do Express.js, foi escrito no arquivo *index.js*, como visto no Código 4, o código necessário para que o servidor começasse a funcionar na porta local 3000. Também foi definido que as futuras rotas a serem desenvolvidas, serão antecedidas pelo prefixo *'/api'*. Sendo assim, um exemplo de rota a ser acessada no servidor rodando localmente na porta 3000 seria *'http://localhost:3000/api/createEmission'*.

Código 4 – Arquivo *index.js* com o setup inicial do servidor Express.js da API rodando na porta 3000.

```
1  const express = require('express');
2
3  const app = express();
4  app.use(express.json());
5
6  const routes = require('./routes/routes');
7
8  app.use('/api', routes)
9
10 app.listen(3000, () => {
```

```

11 |     console.log(`Server Started at ${3000}`)
12 | })

```

### 4.1.2 Configuração e Integração do Banco de Dados MongoDB

Com a configuração inicial do servidor concluída, o próximo passo foi criar um novo banco de dados MongoDB e integrar o mesmo com o servidor através do pacote Mongoose, que foi instalado através do comando `'npm install mongoose'`.

Para a criação de um novo banco de dados MongoDB, foi utilizada uma conta gratuita no servidor do próprio MongoDB, localizado no *website* `'https://cloud.mongodb.com/'`. Após seguir o passo-a-passo disponibilizado pela plataforma para a criação de um banco de dados gratuito, foi disponibilizada uma chave de conexão que deverá ser utilizada para conectar este banco de dados na API. Esta chave pode ser observada na Figura 4.

Figura 4 – Chave de conexão disponibilizada para integração do banco de dados com a API utilizando o usuário `'root'`

```
mongodb+srv://root:<password>@cluster0.fqyxb.mongodb.net/test
```

Fonte: Elaborada pelo autor.

Com o banco de dados criado e chave de conexão disponibilizada, o próximo passo foi realizar a configuração necessária no projeto da API para a integração com este banco de dados utilizando o pacote Mongoose, cuja instalação foi realizada utilizando o comando `'npm install mongoose'`. Com Mongoose instalado, foi adicionado ao código já existente, a configuração da integração da API com o banco de dados criado, utilizando a chave de conexão citada anteriormente. O Código 5 demonstra o estado do arquivo `index.js` após a integração com o banco de dados.

Código 5 – Arquivo `index.js` agora com a integração com o banco de dados MongoDB.

```

1 | const express = require('express');
2 | const mongoose = require('mongoose');
3 | const mongoString = `mongodb+srv://root:password@cluster0.
  |   fqyxb.mongodb.net/test`;
4 |
5 | mongoose.connect(mongoString);
6 | const database = mongoose.connection;
7 |
8 | database.on('error', (error) => {
9 |     console.log(error)
10 | })

```

```

11
12 database.once('connected', () => {
13     console.log('Database Connected');
14 })
15 const app = express();
16 app.use(express.json());
17
18 const routes = require('./routes/routes');
19
20 app.use('/api', routes)
21
22 app.listen(3000, () => {
23     console.log(`Server Started at ${3000}`)
24 })

```

### 4.1.3 Definição dos Modelos de Dados

Com o servidor funcionando e integrado com o banco de dados, o próximo passo foi definir os modelos dos dados a serem registrados pelo painel de controle e salvos no banco.

#### 4.1.3.1 Modelo de Dados de Usuário

Como visto no Código 6, a estrutura definida para o modelo de dados de usuário contém apenas as informações de autenticação, nome e valor a ser utilizado como *baseline* para o cálculo do saldo atual de credito de carbono.

Código 6 – Modelo de dados de usuário.

```

1 const mongoose = require('mongoose');
2
3 const dataSchema = new mongoose.Schema({
4     name: {
5         required: true,
6         type: String
7     },
8     login: {
9         required: true,
10        type: String
11    },
12    password: {
13        required: true,
14        type: String
15    },

```

```

16     baseline: {
17         required: true,
18         type: Number
19     },
20 })
21
22 module.exports = mongoose.model('UserData', dataSchema)

```

#### 4.1.3.2 Modelo de Dados das Emissões de GEE

A estrutura pensada para o modelo de dados de usuário contém uma descrição da origem da emissão, o escopo no qual a emissão se enquadra, o tipo de GEE emitido, a quantidade emitida, o tCO<sub>2</sub>e emitido, e o *id* do usuário registrando a emissão. O Código 7 mostra o resultado da definição do modelo de dados de emissão.

Código 7 – Modelo de dados das emissões de GEE.

```

1  const mongoose = require('mongoose');
2
3  const dataSchema = new mongoose.Schema({
4      description: {
5          required: true,
6          type: String
7      },
8      scope: {
9          required: true,
10         type: Number
11     },
12     gas: {
13         required: true,
14         type: String
15     },
16     amount: {
17         required: true,
18         type: Number
19     },
20     co2Equivalent: {
21         required: true,
22         type: Number
23     },
24     userId: {
25         required: true,
26         type: String

```

```

27     },
28   })
29
30   module.exports = mongoose.model('Data', dataSchema)

```

#### 4.1.4 Desenvolvimento das Rotas de Autenticação

Para autenticação de usuário, foi necessária a criação de duas rotas do tipo *POST*, sendo elas a rota *'/register'* para realizar o registro de um novo usuário, e a rota *'/login'* para efetuar o *login* de um usuário já existente. A definição de ambas as rotas foi exemplificada no Código 8.

Código 8 – Rotas de autenticação de usuário.

```

1  router.post('/register', async (req, res) => {
2    const data = new User({
3      name: req.body.name,
4      login: req.body.login,
5      password: req.body.password,
6      baseline: req.body.baseline,
7    })
8    try {
9      const dataToSave = await data.save();
10     res.status(200).json(dataToSave)
11   }
12   catch (error) {
13     res.status(400).json({ message: error.message })
14   }
15 })
16
17 router.post('/login', async (req, res) => {
18   try {
19     const user = await User.findOne({
20       login: req.body.login,
21       password: req.body.password
22     })
23     res.json(user)
24   }
25   catch (error) {
26     res.status(500).json({ message: error.message })
27   }
28 })

```

#### 4.1.5 Desenvolvimento das Rotas dos Registros de Emissões de GEE

Para o registro de emissões de GEE, foram criadas as rotas `'/createEmission'` para inserção de uma nova emissão, `'/getAll/:userId'` para retornar todas as emissões registradas pelo usuário, `'/getById/:id'` para retornar uma emissão específica, `'/update/:id'` para alterar uma emissão, e `'/delete/:id'` para remover um registro de emissão. Todas as definições das rotas citadas podem ser encontradas no Código 9.

Código 9 – Rotas de registro de emissões de GEE definidas

```

1 router.post('/createEmission', async (req, res) => {
2     const data = new Emission({
3         description: req.body.description,
4         scope: req.body.scope,
5         gas: req.body.gas,
6         amount: req.body.amount,
7         co2Equivalent: req.body.co2Equivalent,
8         userId: req.body.userId
9     })
10
11     try {
12         const dataToSave = await data.save();
13         res.status(200).json(dataToSave)
14     }
15     catch (error) {
16         res.status(400).json({ message: error.message })
17     }
18 })
19
20 router.get('/getAll/:userId', async (req, res) => {
21     try {
22         const data = await Emission.find({
23             userId: req.params.userId
24         });
25         res.json(data)
26     }
27     catch (error) {
28         res.status(500).json({ message: error.message })
29     }
30 })
31 router.get('/getById/:id', async (req, res) => {
32     try {
33         const data = await Emission.findById(req.params.id);

```

```
34     res.json(data)
35   }
36   catch (error) {
37     res.status(500).json({ message: error.message })
38   }
39 })
40 router.patch('/update/:id', async (req, res) => {
41   try {
42     const id = req.params.id;
43     console.log(id)
44
45     const updatedData = req.body;
46     const options = { new: true };
47
48     const result = await Emission.findByIdAndUpdate(
49       id, updatedData, options
50     )
51
52     res.send(result)
53   }
54   catch (error) {
55     res.status(500).json({ message: error.message })
56   }
57 })
58 router.delete('/delete/:id', async (req, res) => {
59   try {
60     const id = req.params.id;
61     console.log(id)
62     const data = await Emission.findByIdAndDelete(id)
63     res.send(`Document with ${data.name} has been deleted
64       ..`)
65   }
66   catch (error) {
67     console.log(error)
68     res.status(400).json({ message: error.message })
69   }
70 })
```

## 4.2 Desenvolvimento do Painel de Controle

O processo de desenvolvimento do painel de controle da aplicação envolveu a utilização do pacote Create-React-App para gerar a estrutura inicial do projeto em React, a instalação dos demais pacotes externos, a construção manual da base de dados de GEE, o desenvolvimento dos componentes e páginas utilizando React, a configuração das rotas de navegação entre as telas e a integração do painel de controle com a API através de requisições HTTP.

### 4.2.1 Configuração Inicial do Projeto em React

A estrutura inicial do projeto React do painel de controle foi gerada utilizando o pacote Create-React-App através do comando `'npx create-react-app CarbonCred'`. A estrutura inicial gerada contém os pacotes essenciais para uma aplicação em React, além de também já disponibilizar scripts pré-definidos para inicializar a aplicação.

### 4.2.2 Instalação dos Pacotes Externos utilizados

Para o desenvolvimento do painel de controle, foram utilizados alguns pacotes externos como `'react-pdf'` para a geração de relatórios de emissões em PDF, `'React Router'` para o gerenciamento de rotas de navegação entre páginas, e `Material-UI` como a biblioteca de componentes visuais básicos que foi utilizada durante o desenvolvimento dos componentes mais complexos.

A instalação desses pacotes foi realizada através do comando `'npm install @react-pdf/renderer react-router-dom @mui/material @mui/icons-material'`.

### 4.2.3 Construção da Base de Dados de GEE

Como não foi possível encontrar uma base de dados completa contendo todos os gases do efeito estufa e seus respectivos GWP, foi necessária a criação de um arquivo Javascript e a inserção manual de todos os GEE presentes no Quarto Relatório de Avaliação do Painel Inter-governamental sobre Mudanças Climáticas, com os valores de GWP de cada gás presentes no relatório, seu nome e sua fórmula química. No Código 10, é possível ver a estrutura utilizada para a construção desta base de dados.

Código 10 – Exemplos de gases adicionados na base de dados de GEE.

```

1 {
2     "name": "Carbon dioxide",
3     "formula": "CO2",
4     "relativeGWP": "1"
5 },
6 {
```



```

7      "name": "Methane",
8      "formula": "CH4",
9      "relativeGWP": "25"
10   },
11   {
12       "name": "Nitrous oxide",
13       "formula": "N2O",
14       "relativeGWP": "298"
15   },
16   {
17       "name": "CFC-11",
18       "formula": "CCl3F",
19       "relativeGWP": "4,750"
20   },
21   {
22       "name": "CFC-12",
23       "formula": "CCl2F2",
24       "relativeGWP": "10,900"
25   },

```

#### 4.2.4 Desenvolvimento dos Componentes

Com a utilização do pacote *Material-UI*, que disponibiliza uma enorme gama de componentes visuais básicos como botões, *containers*, menus, entre outros, apenas se fez necessária a criação de componentes mais complexos, e que geralmente utilizam os componentes presentes em *Material-UI* na sua composição, como pode ser visto no componente *Navbar*, demonstrado no Código 11, que se trata da barra de navegação presente no topo do painel de controle.

Código 11 – Componente *Navbar* construído utilizando diversos componentes fornecidos pela biblioteca *Material-UI*.

```

1  import Box from "@mui/material/Box";
2  import AppBar from "@mui/material/AppBar";
3  import Toolbar from "@mui/material/Toolbar";
4  import IconButton from "@mui/material/IconButton";
5  import MenuIcon from "@mui/icons-material/Menu";
6  import Typography from "@mui/material/Typography";
7  import {Link} from "react-router-dom";
8  import Button from "@mui/material/Button";
9  import * as React from "react";
10
11  function Navbar({onClickMenu}) {

```

```

12     return (
13         <Box sx={{flexGrow: 1}}>
14             <AppBar position="static">
15                 <Toolbar>
16                     <IconButton
17                         size="large"
18                         edge="start"
19                         color="inherit"
20                         aria-label="menu"
21                         sx={{mr: 2}}
22                         onClick={onClickMenu}
23                     >
24                         <MenuIcon/>
25                     </IconButton>
26                     <Typography variant="h6" component="div"
27                         sx={{flexGrow: 1}} color={"white"}>
28                         <Link to={'/'} style={{color: "white",
29                             textDecoration: "none"}}>
30                             CarbonCred
31                         </Link>
32                     </Typography>
33                     <Button href={"/login"} color="inherit">
34                         Logout </Button>
35                 </Toolbar>
36             </AppBar>
37         </Box>
38     );
39 }
40 export default Navbar;

```

## 4.2.5 Desenvolvimento das Páginas

Para o desenvolvimento do painel de controle, foram necessárias a criação de telas de *login* e registro para autenticação do usuário, uma tela para a exibição do saldo de crédito de carbono, uma tela para a listagem de emissões de GEE cadastradas, uma tela para a edição de uma emissão, e também foi necessário desenvolver o modelo do relatório de emissão a ser emitido.

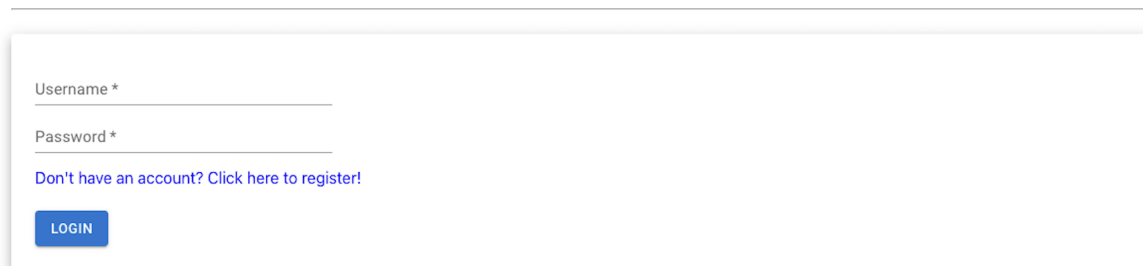
### 4.2.5.1 Página de *Login*

Como visto na Figura 5, a tela de *login* consiste de campos de texto para inserção das credencias do usuário, e caso o usuário insira seus dados, corretamente, leva o usuário até a

tela do saldo de crédito de carbono.

Figura 5 – Página de Login.

### Login

A login form with a white background and a light gray border. It contains two input fields: 'Username \*' and 'Password \*'. Below the password field is a link that says 'Don't have an account? Click here to register!'. At the bottom is a blue button with the text 'LOGIN' in white capital letters.

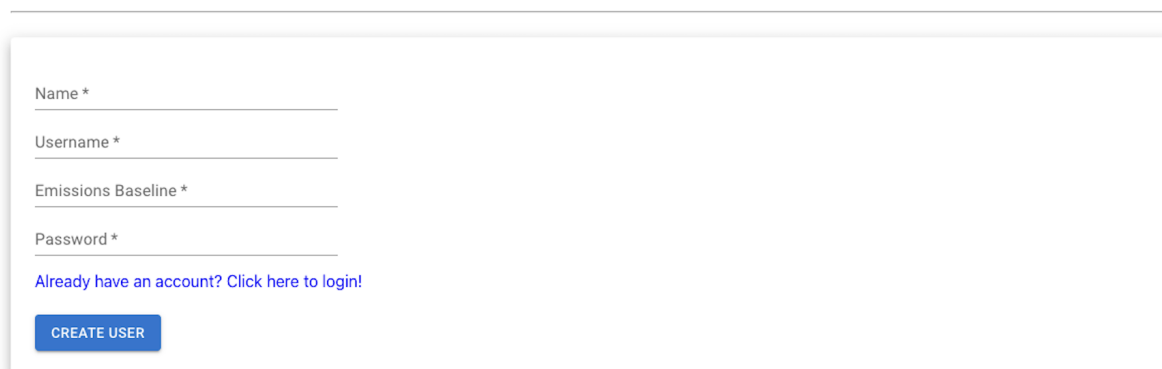
Fonte: Elaborada pelo autor.

#### 4.2.5.2 Página de Registro.

Como visto na Figura 6, a tela de registro consiste de campos de texto para inserção das credencias do usuário, nome e *baseline* de emissões, e caso o registro seja concluído, leva o usuário até a tela de *login*.

Figura 6 – Página de Registro.

### Create new user

A 'Create new user' form with a white background and a light gray border. It contains four input fields: 'Name \*', 'Username \*', 'Emissions Baseline \*', and 'Password \*'. Below the password field is a link that says 'Already have an account? Click here to login!'. At the bottom is a blue button with the text 'CREATE USER' in white capital letters.

Fonte: Elaborada pelo autor.

#### 4.2.5.3 Página de Saldo de Crédito de Carbono

Como visto na Figura 7, a tela de saldo de credito de carbono exhibe o total de emissões e a *baseline* de emissões em tCO<sub>2</sub>e, e baseando-se nesses dois valores, realiza o cálculo que resulta no saldo de créditos de carbono (cálculo este cuja implementação está demonstrada no Código 12), e exhibe o resultado na parte inferior da página. O saldo de crédito de carbono se trata da diferença entre a *baseline* de emissões (que geralmente é definida por valores de emissões de períodos anteriores), e os valores de emissão atuais.

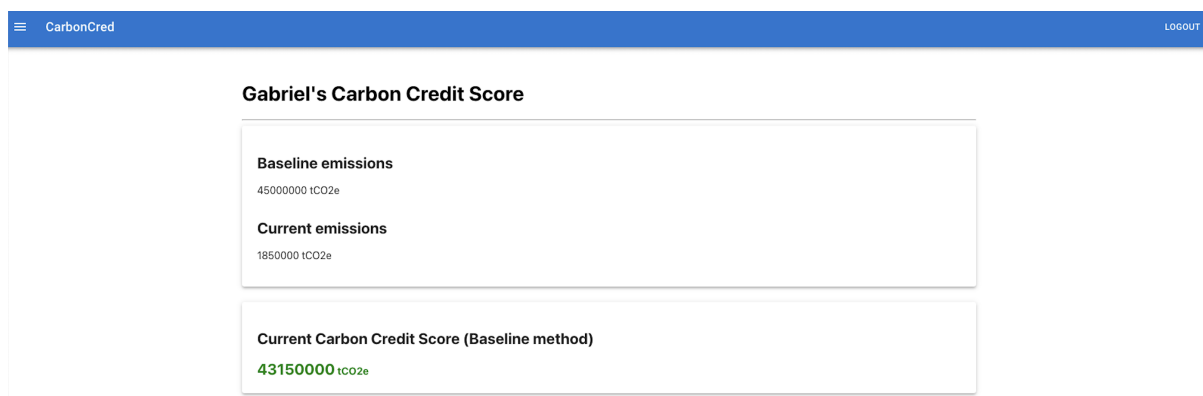
Código 12 – Função simples que realiza o cálculo do saldo de créditos de carbono.

```

1  const getCarbonCreditScore = () => {
2      const score = user.baseline - totalEmissionValues?.
        co2Equivalent;
3      return score
4  }

```

Figura 7 – Página de Saldo de Crédito de Carbono.



Fonte: Elaborada pelo autor.

#### 4.2.5.4 Página de Listagem de Emissões

Como é possível observar na Figura 8, a tela de listagem de emissões, consiste de duas tabelas, sendo que a primeira delas contém a listagem geral de emissões do usuário, enquanto que a segunda tabela exibe os valores gerais resultantes das emissões, tendo uma visão voltada para a individualidade de cada escopo de emissões. Também está presente nesta página, botões para a criação de um novo registro de emissões, edição e exclusão de uma emissão específica, e geração do relatório de emissões em formato PDF, sendo que a implementação deste último pode ser observada no Código 13.

Código 13 – Botão incluindo o componente PDFDownloadLink disponibilizado pelo pacote React-pdf

```

1  return (
2      <Button
3          variant="contained"
4          style={{display: "block", height: "max-content",
            width: "max-content",}}
5      >
6          <PDFDownloadLink style={{color: "white",
            textDecoratoin: "none"}} document={<Table data
              ={data}/>} fileName="somenome.pdf">
7              {(blob, url, loading, error)} =>

```

```

8         loading ? 'Creating report...' : 'Download
          report'
9     }
10    </PDFDownloadLink>
11    </Button>
12
13    );

```

Figura 8 – Página de Listagem de Emissões.

| CarbonCred  |                                |                  |                     |              |                                |                      |   | LOGOUT   |
|---|--------------------------------|------------------|---------------------|--------------|--------------------------------|----------------------|---|--|
| Emissions   |                                |                  |                     |              |                                |                      |   | <a href="#">ADD EMISSION</a> <a href="#">GENERATE REPORT</a> |
| Emission  | Gas emitted                    | Chemical formula | Tons of gas emitted | Relative GWP | Tons of CO2 equivalent (TCO2e) | Scope                | Actions                                     |  |
| Compostagem   | Methane                        | CH4              | 100                 | 25           | 2500                           | 1                    | <a href="#">Edit</a> <a href="#">Delete</a> |  |
| Geradores   | PFC-14                         | CF4              | 250                 | 7,390        | 1847500                        | 2                    | <a href="#">Edit</a> <a href="#">Delete</a> |  |
| <a href="#">&lt;</a> <a href="#">1</a> <a href="#">&gt;</a> |                                |                  |                     |              |                                |                      |   |  |
| Total Emissions   |                                |                  |                     |              |                                |                      |   |  |
| Scope   | Tons of CO2 equivalent (TCO2e) |                  |                     |              |                                | Representativity (%) |   |  |
| 1   | 2500                           |                  |                     |              |                                | 0.14%                |   |  |
| 2   | 1847500                        |                  |                     |              |                                | 99.86%               |   |  |
| 3   | 0                              |                  |                     |              |                                | 0.00%                |   |  |
| Total   | 1850000                        |                  |                     |              |                                | 100.00%              |   |  |

Fonte: Elaborada pelo autor.

O cálculo do tCO2e de cada emissão registrada pelo usuário, foi realizado pela função *getCarbonEquivalent*, demonstrada no Código 14 .

Código 14 – Função que realiza o cálculo de tCO2e de cada emissão listada.

```

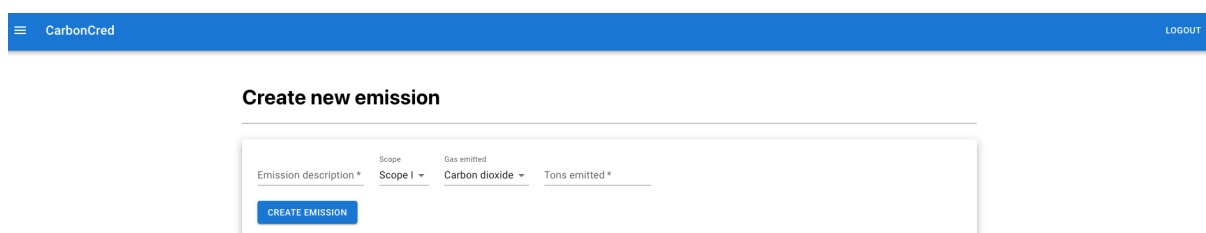
1 export const getCarbonEquivalent = (tons, gasType) => {
2   const greenhouseGasses = gees;
3   const selectedGas = greenhouseGasses.filter((item) => item
      .name === gasType);
4
5   const carbonDioxideEquivalentMultiplier = parseFloat(
      selectedGas[0]?.relativeGWP.replace(/,/g, '.'));
6
7   const carbonDioxideEquivalentTons = tons *
      carbonDioxideEquivalentMultiplier;
8   return carbonDioxideEquivalentTons;
9 }

```

#### 4.2.5.5 Página de Criação de Emissão

Como pode ser observado na Figura 9, a página de criação de uma nova emissão consiste de campos de texto e seletores com os quais o usuário categoriza e descreve a nova emissão, e ao finalizar a inserção, retorna à tela de listagem de emissões.

Figura 9 – Página de Criação de Emissão.



The screenshot shows the 'Create new emission' form in the CarbonCred application. The form is titled 'Create new emission' and contains the following fields and controls:

- Emission description \***: A text input field.
- Scope**: A dropdown menu with 'Scope 1' selected.
- Gas emitted**: A dropdown menu with 'Carbon dioxide' selected.
- Tons emitted \***: A text input field.
- CREATE EMISSION**: A blue button to submit the form.

Fonte: Elaborada pelo autor.

#### 4.2.5.6 Página de Edição de Emissão

Como pode ser observado na Figura 10, a página de edição de uma emissão existente consiste de campos de texto e seletores que já vêm preenchidos com as informações da emissão que está sendo editada, com os quais o usuário categoriza e descreve as alterações a serem feitas no registro, e ao finalizar a edição, retorna à tela de listagem de emissões.

#### 4.2.5.7 Página de Relatório de Emissões Emitido

Como exibido na Figura 11, o relatório de emissões gerado, consiste de duas tabelas, sendo que a primeira delas contém a listagem geral de emissões do usuário, enquanto que a segunda tabela exibe os valores gerais resultantes das emissões, tendo uma visão voltada para a individualidade de cada escopo de emissões.

### 4.2.6 Configuração das Rotas de Navegação

A configuração das rotas de navegação do painel de controle utilizando o pacote React Router pode ser vista a seguir no Código 15.

Figura 10 – Página de Edição de Emissão.

**Edit emission**

|                        |         |             |                |
|------------------------|---------|-------------|----------------|
| Emission description * | Scope   | Gas emitted | Tons emitted * |
| Compostagem            | Scope 1 | Methane     | 100            |

[UPDATE EMISSION](#)

Fonte: Elaborada pelo autor.

Figura 11 – Página de Relatório de Emissões Emitido.

**Emission List**

| Description | Gas     | Tons of gas | Rel. GWP | tCO2e   | Scope |
|-------------|---------|-------------|----------|---------|-------|
| Compostagem | Methane | 100         | 25       | 2500    | 1     |
| Geradores   | PFC-14  | 250         | 7,390    | 1847500 | 2     |

**Emissions Total**

| Scope        | tCO2e (tons)   | Representativity (%) |
|--------------|----------------|----------------------|
| 1            | 2500           | 0.13513513513513514  |
| 2            | 1847500        | 99.86486486486487    |
| 3            | 0              | 0                    |
| <b>Total</b> | <b>1850000</b> | <b>100</b>           |

Fonte: Elaborada pelo autor.

Código 15 – Definição das rotas de navegação do painel de controle.

```

1 const router = createBrowserRouter([
2   {
3     path: "/",
4     element: <CarbonCredit/>,
5     errorElement: <Error/>
6   },
7   {

```

```

8      path: "/carbon-credit",
9      element: <CarbonCredit/>,
10     errorElement: <Error/>
11   },
12   {
13     path: "/emissions",
14     element: <Emissions/>,
15     errorElement: <Error/>
16   },
17   {
18     path: "/emissions/create",
19     element: <CreateEmission/>,
20     errorElement: <Error/>
21   },
22   {
23     path: "/emissions/edit/:id",
24     element: <UpdateEmission/>,
25     errorElement: <Error/>
26   },
27   {
28     path: "/login",
29     element: <Login/>,
30     errorElement: <Error/>
31   },
32   {
33     path: "/register",
34     element: <Register/>,
35     errorElement: <Error/>
36   },

```

#### 4.2.7 Integração do Painel de Controle com a API

A integração entre o painel de controle e a API é realizada através de requisições HTTP utilizando a função *fetch*, presente no conjunto de funções padrão do Javascript. No Código 16, é possível ver como é realizada a requisição do tipo *POST* para a criação de um novo registro de emissão através da rota *'/createEmission'*.

Código 16 – Integração do painel de controle com a API para a criação de uma nova emissão de GEE

```

1  const createEmission = () => {
2      const emission = {
3          description,

```



```
4         scope ,
5         gas ,
6         amount ,
7         co2Equivalent: getCarbonEquivalent(amount, gas),
8         userId: user?._id,
9     }
10
11     fetch('http://localhost:3000/api/createEmission', {
12         method: 'POST',
13         body: JSON.stringify(emission),
14         headers: {
15             'Content-Type': 'application/json'
16             // 'Content-Type': 'application/x-www-form-
17                 urlencoded',
18         },
19     })
20     navigate("/emissions")
21 }
```

## 5 Conclusão

Este trabalho teve como objetivo desenvolver um *software* de controle de emissões de gases de efeito estufa e visão facilitada sobre o saldo de crédito de carbono e alcançou esse objetivo de forma satisfatória, contribuindo para a redução das emissões de gases de efeito estufa através da tomada de decisões informadas sobre as emissões de gases realizadas e sobre o saldo de crédito de carbono.

É possível afirmar que o *software* cumpriu o seu objetivo, apresentando uma interface intuitiva e funcionalidades que atendem às necessidades e expectativas referentes ao fácil controle de emissões de GEE, além de uma visão facilitada do saldo de créditos de carbono. Além disso, o *software* foi testado e avaliado com usuários, o que permitiu verificar a sua qualidade e funcionalidade.

Este trabalho agrega conteúdos importantes a respeito das alterações climáticas, os diversos tratados internacionais que buscam resolver este problema, além de também fornecer um olhar atencioso e específico se tratando das tecnologias e padrões de desenvolvimento utilizados durante o desenvolvimento do *software*.

### 5.1 Trabalhos Futuros

Visto que para o desenvolvimento deste trabalho foram utilizados parâmetros e informações básicas sobre emissões de GEE e crédito de carbono, é possível incrementar a flexibilidade do *software*, através da adição de mais informações a serem registradas sobre as emissões, além de melhorias de experiência do usuário, como adição de filtros na listagem dessas emissões.

Além disso, seria interessante para trabalhos futuros, disponibilizar uma integração entre este *software* e os mercados de crédito de carbono, permitindo assim a negociação desses créditos para os usuários desta aplicação.

# Referências

- BIERMANN, F.; BOAS, I. *International Environmental Law and Policy*. Londres: Earthscan, 2014.
- BRADSHAW, S.; BRAZIL, E.; CHODOROW, K. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. [S.l.]: O'Reilly Media, 2019. ISBN 9781491954416.
- CHINNATHAMBI, K. *Learning React: A Hands-On Guide to Building Web Applications Using React and Redux*. [S.l.]: Addison-Wesley Professional, 2018.
- FISCHER, C.; TAYLOR, M. *Linking Carbon Markets: Options for the Development of an International Carbon Market*. [S.l.]: Edward Elgar Publishing, 2008.
- GOHARI, E. Carbon offsetting: a review of the most popular methods. *Renewable and Sustainable Energy Reviews*, v. 54, p. 1135–1150, 2016.
- GUPTA, Y. Carbon credit: A step towards green environment. *Global Journal of Management and Business Research*, Global Journals Inc, v. 11, abr. 2011. ISSN 0975-5853.
- HAVERBEKE, M. *Eloquent JavaScript*. [S.l.: s.n.], 2016.
- IBGE. *Relatório de Emissões de Gases de Efeito Estufa no Brasil*. 2016.
- IPCC. *Mudança do Clima 2007: Síntese do Quinto Relatório de Avaliação*. 2007. Disponível em: <<https://www.ipcc.ch/report/ar4/syr/>>. Acesso em: 5 jan. 2023.
- IPCC. *Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. 2014. Disponível em: <[https://www.ipcc.ch/site/assets/uploads/2018/02/SYR\\_AR5\\_FINAL\\_full\\_wcover.pdf](https://www.ipcc.ch/site/assets/uploads/2018/02/SYR_AR5_FINAL_full_wcover.pdf)>. Acesso em: 13 Dez. 2022.
- LAURET, A. *The Design of Web APIs*. [S.l.: s.n.], 2019.
- MASSON-DELMOTTE, V.; ZHAI, P.; PÖRTNER, H.-O.; ROBERTS, D.; SKEA, J.; SHUKLA, P. R.; PIRANI, A.; MOUFOUMA-OKIA, W.; PÉAN, C.; PIDCOCK, R.; CONNORS, S.; MATTHEWS, J. B. R.; CHEN, Y.; ZHOU, X.; GOMIS, M. I.; LONNOY, E.; MAYCOCK, T.; TIGNOR, M.; WATERFIELD, T. (Ed.). *Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty*. [S.l.: s.n.], 2018. In press.
- ROGELJ, J. The paris agreement on climate change: A commentary. *Nature Climate Change*, v. 6, p. 709–715, 2016.
- SEREDENKO, B. P. *Node.js: The Complete Guide to Build RESTful APIs with Loopback.js*. [S.l.: s.n.], 2018.
- SILVA, J.; ALMEIDA, D.; SANTOS, M. O comércio de créditos de carbono e a mitigação das emissões de gases de efeito estufa. *Ambiente Sociedade*, v. 21, p. 193–214, 2018.

STOCKER, T.; QIN, D.; PLATTNER, G.-K.; TIGNOR, M.; ALLEN, S.; BOSCHUNG, J.; NAUELS, A.; XIA, Y.; BEX, V.; MIDGLEY, P. (Ed.). *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, 2013. 659-740 p.

UNFCCC. s.d. Disponível em: <<https://unfccc.int/process-and-meetings/what-is-the-united-nations-framework-convention-on-climate-change>>. Acesso em: 5 jan. 2023.