



GREENPRINT

Calcule sua pegada de carbono para uma vida
mais sustentável

João Vinícius Alves (559369) – Arthur Ribeiro (560417) – Juan Pablo – (560445)

Sumário

- Introdução..... 2
- Menu e Funcionalidades..... 2
 - Iniciar cálculo de emissão de carbono 2
 - Ver últimos dados..... 3
 - Sair 3
- Fluxo do Programa..... 3
- Código Fonte 4
- Acesse o Projeto 13

Introdução

O GreenPrint é um projeto desenvolvido para calcular e monitorar as emissões de carbono de uma pessoa, com base em três principais atividades: transporte, consumo de energia e alimentação. A ideia é oferecer aos usuários uma forma de calcular sua pegada de carbono, acompanhando os impactos ambientais de suas ações no dia a dia, e possibilitar a comparação entre os cálculos de emissão ao longo do tempo.

Este projeto coleta informações do usuário sobre o uso de veículos pessoais e transporte público, consumo de energia e gás natural, e hábitos alimentares, como o consumo de carne, frango, peixe e laticínios. Com essas informações, o sistema calcula a quantidade de CO₂ gerado por cada uma dessas atividades e fornece um total de emissões.

Menu e Funcionalidades

O menu principal apresenta as seguintes opções:

Iniciar cálculo de emissão de carbono

Ao selecionar essa opção, o usuário será guiado por três etapas, onde fornecerá informações sobre:

- Veículos: Tipo de combustível e quilômetros dirigidos semanalmente.
- Transporte público: Meio de transporte utilizado e distância percorrida semanalmente.
- Viagens aéreas: Número de viagens realizadas por ano e distância média dos voos.
- Energia residencial: Consumo de energia elétrica, gás natural e GLP.
- Alimentos: Frequência de consumo de carne vermelha, frango, peixe e laticínios.
- Com base nessas informações, o sistema calculará as emissões de carbono de cada área e exibirá o total.

Ver últimos dados

Esta opção permite ao usuário visualizar as informações de emissões de carbono dos cálculos anteriores. Será exibido:

- O total de emissões de carbono geradas até o momento.
- O cálculo de emissão de carbono com maior valor.
- O cálculo com menor valor.
- A média de emissões de carbono de todos os cálculos realizados.

Sair

- Ao escolher essa opção, o programa será encerrado.

Fluxo do Programa

1. O programa começa com o menu principal, oferecendo as opções de realizar um novo cálculo, verificar os dados anteriores ou sair.
2. Quando o usuário escolhe iniciar cálculo de emissão de carbono, o programa começa a fazer perguntas sobre as categorias relevantes (veículos, transporte público, energia, alimentação), e ao final realiza os cálculos e exibe os resultados.

3. Quando o usuário escolhe ver últimos dados, o programa exibe os totais e comparações dos cálculos de emissões anteriores.
4. O usuário pode continuar interagindo com o menu até escolher a opção sair, encerrando o programa.

Essa abordagem simples visa facilitar a compreensão da pegada de carbono de cada usuário, proporcionando uma ferramenta prática para reduzir o impacto ambiental.

Código Fonte

```
from datetime import datetime

def calcular_emissao(consumo, fator):

    return consumo * fator

def emissao_veiculos():

    print("\nEssa primeira etapa é sobre seus meios de locomoção\n"

          "Primeiro vou fazer algumas perguntas sobre seu veiculo pessoal.")

    while True:

        try:

            print("\nQual o tipo de combustivel do seu veiculo?\n"

                  "1. Diesel\n"

                  "2. Gasolina\n"

                  "3. Elétrico\n"

                  "4. Não possui veiculo\n"

                  "=====")

            veiculo_escolhido = int(input("Digite a opção desejada:\n"))
```

```

match veiculo_escolhido:

    case 1:

        consumo_veiculo_escolhido = 0.13

        break

    case 2:

        consumo_veiculo_escolhido = 0.2

        break

    case 3:

        consumo_veiculo_escolhido = 0.05

        break

    case 4:

        consumo_veiculo_escolhido = 0

        break

    case _:

        print("Opção inválida")

except ValueError:

    print("Valor inválido")


if veiculo_escolhido != 4:

    while True:

        try:

            km_veiculo_pessoal = float(input("Quantos quilômetros por semana, em média, você dirige?\n"))

            break

        except ValueError:

            print("Valor inválido\n")

    else:

        km_veiculo_pessoal = 0

```

```
emissao_veiculo_pessoal = calcular_emissao(consumo_veiculo_escolhido,  
km_veiculo_pessoal)
```

```
print("\nAgora vou fazer algumas perguntas sobre seu uso de transporte publico")
```

```
while True:
```

```
    try:
```

```
        print("Qual o principal meio de transporte publico que você usa?\n"
```

```
              "1. Ônibus\n"
```

```
              "2. Trem/Metro\n"
```

```
              "3. Não utilizo transporte publico\n"
```

```
              "=====")
```

```
    transporte_escolhido = int(input("Digite a opção desejada:\n"))
```

```
    match transporte_escolhido:
```

```
        case 1:
```

```
            consumo_transporte_escolhido = 0.05
```

```
            break
```

```
        case 2:
```

```
            consumo_transporte_escolhido = 0.03
```

```
            break
```

```
        case 3:
```

```
            consumo_transporte_escolhido = 0
```

```
            break
```

```
        case _:
```

```
            print("Opção inválida")
```

```
    except ValueError:
```

```
        print("Valor inválido\n")
```

```
if transporte_escolhido != 3:

    while True:

        try:

            km_transporte_publico = float(input("Quantos quilômetros por semana, em média, você anda de transporte público?\n"))

            break

        except ValueError:

            print("Valor inválido\n")

    else:

        km_transporte_publico = 0

    emissao_transporte_publico = calcular_emissao(consumo_transporte_escolhido, km_transporte_publico)

    print("\nPara finalizar essa etapa, vou fazer umas perguntas sobre viagens aéreas.")

    while True:

        try:

            vezes_viagem_aviao = int(input("Quantas vezes por ano você viaja de avião?\n"))

            break

        except ValueError:

            print("Valor inválido\n")

    while True:

        try:

            km_aviao = float(input("Qual a distância média dos seus voos?\n"))

            break

        except ValueError:

            print("Valor inválido\n")

    emissao_aviao = calcular_emissao(vezes_viagem_aviao * 0.09, km_aviao)
```



```
return emissao_veiculo_pessoal + emissao_transporte_publico + emissao_aviao
```

```
def emissao_energia():
```

```
    print("\nEssa segunda etapa é sobre sua energia residencial\n"
```

```
        "Primeiro vou fazer algumas perguntas sobre seu uso de energia.")
```

```
    while True:
```

```
        try:
```

```
            kwh_mes = float(input("Quantos kWh você consome por mês?\n"))
```

```
            break
```

```
        except ValueError:
```

```
            print("Valor inválido\n")
```

```
    emissao_energia_residencial = calcular_emissao(0.5, kwh_mes)
```

```
    while True:
```

```
        try:
```

```
            m3_gas_natural_mes = float(input("Quantos m³ de gás natural você usa por mês?\n"))
```

```
            break
```

```
        except ValueError:
```

```
            print("Valor inválido\n")
```

```
    emissao_gas_natural = calcular_emissao(2.2, m3_gas_natural_mes)
```

```
    while True:
```

```
        try:
```

```
            kgs_glp_mes = float(input("Quantos kg de GLP você usa por mês?\n"))
```

```
            break
```

```
        except ValueError:
```

```
            print("Valor inválido\n")
```

```
    emissao_glp_mes = calcular_emissao(3, kgs_glp_mes)
```

```
return emissao_energia_residencial + emissao_gas_natural + emissao_glp_mes
```

```
def emissao_alimentos():
```

```
    print("\nEssa ultima etapa é sobre sua alimentação\n"
```

```
        "Vou fazer algumas perguntas do seu consumo de alimento.")
```

```
    while True:
```

```
        try:
```

```
            porcoes_carne_vermelha = int(input("Quantas vezes, por semana, você consome carne  
vermelha?\n"))
```

```
            break
```

```
        except ValueError:
```

```
            print("Valor inválido")
```

```
    emissao_carne_vermelha = calcular_emissao(4.5, porcoes_carne_vermelha)
```

```
    while True:
```

```
        try:
```

```
            porcoes_frango = int(input("Quantas vezes, por semana, você consome frango?\n"))
```

```
            break
```

```
        except ValueError:
```

```
            print("Valor inválido\n")
```

```
    emissao_frango = calcular_emissao(1.1, porcoes_frango)
```

```
    while True:
```

```
        try:
```

```
            porcoes_peixe = int(input("Quantas vezes, por semana, você consome peixe?\n"))
```

```
            break
```

```
        except ValueError:
```

```
            print("Valor inválido\n")
```

```
emissao_peixe = calcular_emissao(1.5, porcoes_peixe)
```

```
while True:
```

```
    try:
```

```
        porcoes_ovos_laticinios = int(input("Quantas vezes, por semana, você consome  
ovos/laticinios?\n"))
```

```
        break
```

```
    except ValueError:
```

```
        print("Valor inválido\n")
```

```
emissao_ovos_laticinios = calcular_emissao(0.5, porcoes_ovos_laticinios)
```

```
return emissao_carne_vermelha + emissao_frango + emissao_peixe +  
emissao_ovos_laticinios
```

```
def iniciar_calculo():
```

```
    valor_emissao_veiculo = emissao_veiculos()
```

```
    valor_emissao_energia = emissao_energia()
```

```
    valor_emissao_alimento = emissao_alimentos()
```

```
    total_emissao = valor_emissao_alimento + valor_emissao_energia + valor_emissao_veiculo
```

```
    ultimos_calculos.append({
```

```
        "data_criacao": datetime.now().date().strftime("%d/%m/%Y"), #.strftime("%d/%m/%Y")
```

```
        "valor_emissao_veiculo":valor_emissao_veiculo,
```

```
        "valor_emissao_energia":valor_emissao_energia,
```

```
        "valor_emissao_alimento":valor_emissao_alimento,
```

```
        "total_emissao":total_emissao
```

```
    })
```

```
    print(f"\nAqui estão seu niveis de emissão de carbono:\n")
```

```
        f"Você gerou {valor_emissao_veiculo} kg de CO2 em veiculos\n")
```

```
f"Você gerou {valor_emissao_energia} kg de CO2 em energia\n"
f"Você gerou {valor_emissao_alimento} kg de CO2 em alimentos\n"
f"Em geral você produziu {total_emissao} kg de CO2")
```

```
def ver_dados():
```

```
    sum_total_emissao_carbono = 0
```

```
    maior_emissao_carbono = {
```

```
        "data_criacao":datetime,
```

```
        "total_emissao": 0
```

```
    }
```

```
    menor_emissao_carbono = {
```

```
        "data_criacao": datetime,
```

```
        "total_emissao": 999999999999999999999999999999
```

```
    }
```

```
for calculo in ultimos_calculos:
```

```
    sum_total_emissao_carbono += calculo["total_emissao"]
```

```
if calculo["total_emissao"] > maior_emissao_carbono["total_emissao"]:
```

```
    maior_emissao_carbono = calculo
```

```
if calculo["total_emissao"] < menor_emissao_carbono["total_emissao"]:
```

```
    menor_emissao_carbono = calculo
```

```
media_emissao = sum_total_emissao_carbono/len(ultimos_calculos)
```

```

print("Aqui estão seus dados:\n"

      f"O total de carbono gerado por você foi de {sum_total_emissao_carbono:.2f} kg de
CO2\n"

      f"Seu calculo de emissão do dia {maior_emissao_carbono["data_criacao"]} foi o que mais
gerou CO2, gerando {maior_emissao_carbono["total_emissao"]} kg de CO2\n"

      f"Seu calculo de emissão do dia {menor_emissao_carbono["data_criacao"]} foi o que
menos gerou CO2, gerando {menor_emissao_carbono["total_emissao"]} kg de CO2\n"

      f"Em média você gera {media_emissao:.2f} kg de CO2\n")

```

```

ultimos_calculos = []

```

```

while True:

```

```

    try:

```

```

        print("\nBem vindo ao GreenPrint\n"

              "1. Iniciar calculo de emissão de carbono\n"

              "2. Ver ultimos dados\n"

              "o. Sair\n"

              "=====")

```

```

        opcao = int(input("Digite a opção desejada:\n"))

```

```

    match opcao:

```

```

        case 0:

```

```

            print("Saindo...")

```

```

            break

```

```

        case 1:

```

```

            iniciar_calculo()

```

```

        case 2:

```

```

            ver_dados()

```

```
case _:  
    print("Opção inválida")  
except ValueError:  
    print("Valor inválido")
```

Acesse o Projeto

[MMChallengeMM/GS24-Python](#)