import streamlit as st

import pandas as pd

# Configuração

st.set\_page\_config(*page\_title*="Sistema de Cargas CORRETO", *layout*="centered")

# Funções de cálculo

def calcular\_distribuicao(*total*, *num\_lotes*, *cristal*=None, *demerara*=None, *sacos*=False):

    lotes\_inteiros = int(*num\_lotes*)

    tem\_meio\_lote = *num\_lotes* % 1 != 0

    if tem\_meio\_lote:

        tamanho\_lote = int(*total* // (lotes\_inteiros + 0.5))

        resto = *total* - (lotes\_inteiros \* tamanho\_lote)

    else:

        tamanho\_lote = *total* // lotes\_inteiros

        resto = *total* % lotes\_inteiros

    primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

    outros\_lotes = tamanho\_lote

    dados = []

    if *cristal* is not None:  # Caso Cristal + Demerara

        cristal\_restante = *cristal*

        if not tem\_meio\_lote and cristal\_restante >= primeiro\_lote:

            divisor = 5 if *sacos* else 12

            dados.append(["Lote 1", f"{divisor}\*{primeiro\_lote//divisor}+{primeiro\_lote%divisor}", "0", primeiro\_lote])

            cristal\_restante -= primeiro\_lote

        elif not tem\_meio\_lote:

            demerara\_lote1 = primeiro\_lote - cristal\_restante

            divisor\_cristal = 5 if *sacos* else 12

            dados.append(["Lote 1",

                         f"{divisor\_cristal}\*{cristal\_restante//divisor\_cristal}+{cristal\_restante%divisor\_cristal}",

                         f"12\*{demerara\_lote1//12}+{demerara\_lote1%12}",

                         primeiro\_lote])

            cristal\_restante = 0

        for i in range(1 if tem\_meio\_lote else 0, lotes\_inteiros + (1 if tem\_meio\_lote else 0)):

            if cristal\_restante <= 0:

                dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                             "0",

                             f"12\*{outros\_lotes//12}+{outros\_lotes%12}",

                             outros\_lotes])

            else:

                if cristal\_restante >= outros\_lotes:

                    divisor = 5 if *sacos* else 12

                    dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                                 f"{divisor}\*{outros\_lotes//divisor}+{outros\_lotes%divisor}",

                                 "0",

                                 outros\_lotes])

                    cristal\_restante -= outros\_lotes

                else:

                    demerara\_lote = outros\_lotes - cristal\_restante

                    divisor\_cristal = 5 if *sacos* else 12

                    dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                                 f"{divisor\_cristal}\*{cristal\_restante//divisor\_cristal}+{cristal\_restante%divisor\_cristal}",

                                 f"12\*{demerara\_lote//12}+{demerara\_lote%12}",

                                 outros\_lotes])

                    cristal\_restante = 0

        if tem\_meio\_lote:

            dados.append(["Meio lote", "0", f"6\*{resto//6}+{resto%6}", resto])

        demerara\_distribuido = *total* - (*cristal* - cristal\_restante)

        if cristal\_restante == 0 and demerara\_distribuido == *demerara*:

            st.success(f"✓ CERTO! Cristal: {*cristal*} | Demerara: {*demerara*} | Total: {*total*}")

        else:

            st.error(f"ERRO! Cristal: {*cristal*-cristal\_restante}/{*cristal*} | Demerara: {demerara\_distribuido}/{*demerara*}")

        df = pd.DataFrame(dados, *columns*=["Lote", "Cristal", "Demerara", "Total"])

    else:  # Caso apenas Cristal ou Sacos

        divisor = 5 if *sacos* else 12

        meio\_divisor = 5 if *sacos* else 6

        if not tem\_meio\_lote:

            dados.append(["Lote 1", f"{divisor}\*{(primeiro\_lote//divisor)}+{primeiro\_lote%divisor}", primeiro\_lote])

            for i in range(1, lotes\_inteiros):

                dados.append([f"Lote {i+1}", f"{divisor}\*{outros\_lotes//divisor}+{outros\_lotes%divisor}", outros\_lotes])

        else:

            for i in range(lotes\_inteiros):

                dados.append([f"Lote {i+1}", f"{divisor}\*{outros\_lotes//divisor}+{outros\_lotes%divisor}", outros\_lotes])

            dados.append(["Meio lote", f"{meio\_divisor}\*{resto//meio\_divisor}+{resto%meio\_divisor}", resto])

        df = pd.DataFrame(dados, *columns*=["Lote", "Lastro", "Total"])

    st.write(df.to\_html(*index*=False), *unsafe\_allow\_html*=True)

def calcular\_sacos(*total\_sacos*, *num\_lotes\_sacos*):

    lotes\_inteiros = int(*num\_lotes\_sacos*)

    tem\_meio\_lote = *num\_lotes\_sacos* % 1 != 0

    if tem\_meio\_lote:

        tamanho\_lote = int(*total\_sacos* // (lotes\_inteiros + 0.5))

        resto = *total\_sacos* - (lotes\_inteiros \* tamanho\_lote)

    else:

        tamanho\_lote = *total\_sacos* // lotes\_inteiros

        resto = *total\_sacos* % lotes\_inteiros

    primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

    outros\_lotes = tamanho\_lote

    dados = []

    if not tem\_meio\_lote:

        dados.append(["Lote 1", f"5\*{(primeiro\_lote//5)}+{primeiro\_lote%5}", primeiro\_lote])

        for i in range(1, lotes\_inteiros):

            dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

    else:

        for i in range(lotes\_inteiros):

            dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

        dados.append(["Meio lote", f"5\*{resto//5}+{resto%5}", resto])

    df = pd.DataFrame(dados, *columns*=["Lote", "Sacos (5)", "Total"])

    st.write(df.to\_html(*index*=False), *unsafe\_allow\_html*=True)

# Interface

st.title("📦 SISTEMA DE CARGAS (REGRA EXATA)")

tab1, tab2, tab3 = st.tabs(["Fardo Cristal + Demerara", "Apenas Cristal", "Apenas Sacos"])

with tab1:

    st.header("Cálculo Fardo Cristal + Demerara")

    total\_fardo = st.number\_input("Total:", *min\_value*=1, *step*=1, *value*=1070, *key*="total\_fardo")

    lotes\_fardo = st.number\_input("Nº Lotes:", *min\_value*=0.5, *step*=0.5, *value*=10.5, *key*="lotes\_fardo")

    cristal = st.number\_input("Cristal:", *min\_value*=0, *step*=1, *value*=500, *key*="cristal")

    demerara = st.number\_input("Demerara:", *min\_value*=0, *step*=1, *value*=570, *key*="demerara")

    if st.button("Calcular Fardo", *key*="btn\_fardo"):

        if (cristal + demerara) != total\_fardo:

            st.error("Soma incorreta!")

        else:

            calcular\_distribuicao(total\_fardo, lotes\_fardo, cristal, demerara)

with tab2:

    st.header("Cálculo Apenas Cristal")

    total\_cristal = st.number\_input("Total:", *min\_value*=1, *step*=1, *value*=1070, *key*="total\_cristal")

    lotes\_cristal = st.number\_input("Nº Lotes:", *min\_value*=0.5, *step*=0.5, *value*=10.5, *key*="lotes\_cristal")

    usar\_sacos = st.checkbox("Usar sacos (lastro de 5)", *key*="usar\_sacos")

    if st.button("Calcular Cristal", *key*="btn\_cristal"):

        calcular\_distribuicao(total\_cristal, lotes\_cristal, *sacos*=usar\_sacos)

with tab3:

    st.header("Cálculo Apenas Sacos")

    total\_sacos = st.number\_input("Total Sacos:", *min\_value*=1, *step*=1, *value*=300, *key*="total\_sacos")

    lotes\_sacos = st.number\_input("Nº Lotes Sacos:", *min\_value*=0.5, *step*=0.5, *value*=11.0, *key*="lotes\_sacos")

    if st.button("Calcular Sacos", *key*="btn\_sacos"):

        calcular\_sacos(total\_sacos, lotes\_sacos)

st.markdown("---")

st.markdown("\*\*Desenvolvido por Lucas Cardoso\*\*")

aqui sim finalizou

import streamlit as st

import pandas as pd

# Configuração

st.set\_page\_config(page\_title="Sistema de Cargas CORRETO", layout="centered")

# Funções de cálculo

def calcular\_distribuicao\_fardos(total, num\_lotes, cristal=None, demerara=None):

lotes\_inteiros = int(num\_lotes)

tem\_meio\_lote = num\_lotes % 1 != 0

if tem\_meio\_lote:

tamanho\_lote = int(total // (lotes\_inteiros + 0.5))

resto = total - (lotes\_inteiros \* tamanho\_lote)

else:

tamanho\_lote = total // lotes\_inteiros

resto = total % lotes\_inteiros

primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

outros\_lotes = tamanho\_lote

dados = []

if cristal is not None: # Caso Cristal + Demerara

cristal\_restante = cristal

if not tem\_meio\_lote and cristal\_restante >= primeiro\_lote:

dados.append(["Lote 1", f"12\*{primeiro\_lote//12}+{primeiro\_lote%12}", "0", primeiro\_lote])

cristal\_restante -= primeiro\_lote

elif not tem\_meio\_lote:

demerara\_lote1 = primeiro\_lote - cristal\_restante

dados.append(["Lote 1",

f"12\*{cristal\_restante//12}+{cristal\_restante%12}",

f"12\*{demerara\_lote1//12}+{demerara\_lote1%12}",

primeiro\_lote])

cristal\_restante = 0

for i in range(1 if tem\_meio\_lote else 0, lotes\_inteiros + (1 if tem\_meio\_lote else 0)):

if cristal\_restante <= 0:

dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

"0",

f"12\*{outros\_lotes//12}+{outros\_lotes%12}",

outros\_lotes])

else:

if cristal\_restante >= outros\_lotes:

dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

f"12\*{outros\_lotes//12}+{outros\_lotes%12}",

"0",

outros\_lotes])

cristal\_restante -= outros\_lotes

else:

demerara\_lote = outros\_lotes - cristal\_restante

dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

f"12\*{cristal\_restante//12}+{cristal\_restante%12}",

f"12\*{demerara\_lote//12}+{demerara\_lote%12}",

outros\_lotes])

cristal\_restante = 0

if tem\_meio\_lote:

dados.append(["Meio lote", "0", f"6\*{resto//6}+{resto%6}", resto])

demerara\_distribuido = total - (cristal - cristal\_restante)

if cristal\_restante == 0 and demerara\_distribuido == demerara:

st.success(f"✓ CERTO! Cristal: {cristal} | Demerara: {demerara} | Total: {total}")

else:

st.error(f"ERRO! Cristal: {cristal-cristal\_restante}/{cristal} | Demerara: {demerara\_distribuido}/{demerara}")

df = pd.DataFrame(dados, columns=["Lote", "Cristal", "Demerara", "Total"])

else: # Caso apenas Fardos Cristal

if not tem\_meio\_lote:

dados.append(["Lote 1", f"12\*{(primeiro\_lote//12)}+{primeiro\_lote%12}", primeiro\_lote])

for i in range(1, lotes\_inteiros):

dados.append([f"Lote {i+1}", f"12\*{outros\_lotes//12}+{outros\_lotes%12}", outros\_lotes])

else:

for i in range(lotes\_inteiros):

dados.append([f"Lote {i+1}", f"12\*{outros\_lotes//12}+{outros\_lotes%12}", outros\_lotes])

dados.append(["Meio lote", f"6\*{resto//6}+{resto%6}", resto])

df = pd.DataFrame(dados, columns=["Lote", "Fardos (12)", "Total"])

st.write(df.to\_html(index=False), unsafe\_allow\_html=True)

def calcular\_sacos(total\_sacos, num\_lotes\_sacos):

lotes\_inteiros = int(num\_lotes\_sacos)

tem\_meio\_lote = num\_lotes\_sacos % 1 != 0

if tem\_meio\_lote:

tamanho\_lote = int(total\_sacos // (lotes\_inteiros + 0.5))

resto = total\_sacos - (lotes\_inteiros \* tamanho\_lote)

else:

tamanho\_lote = total\_sacos // lotes\_inteiros

resto = total\_sacos % lotes\_inteiros

primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

outros\_lotes = tamanho\_lote

dados = []

if not tem\_meio\_lote:

dados.append(["Lote 1", f"5\*{(primeiro\_lote//5)}+{primeiro\_lote%5}", primeiro\_lote])

for i in range(1, lotes\_inteiros):

dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

else:

for i in range(lotes\_inteiros):

dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

dados.append(["Meio lote", f"5\*{resto//5}+{resto%5}", resto])

df = pd.DataFrame(dados, columns=["Lote", "Sacos (5)", "Total"])

st.write(df.to\_html(index=False), unsafe\_allow\_html=True)

# Interface

st.title("📦 SISTEMA DE CARGAS (REGRA EXATA)")

tab1, tab2, tab3 = st.tabs(["Fardo Cristal + Demerara", "Apenas Fardos Cristal", "Apenas Sacos"])

with tab1:

st.header("Cálculo Fardo Cristal + Demerara")

total\_fardo = st.number\_input("Total:", min\_value=1, step=1, value=1070, key="total\_fardo")

lotes\_fardo = st.number\_input("Nº Lotes:", min\_value=0.5, step=0.5, value=10.5, key="lotes\_fardo")

cristal = st.number\_input("Cristal:", min\_value=0, step=1, value=500, key="cristal")

demerara = st.number\_input("Demerara:", min\_value=0, step=1, value=570, key="demerara")

if st.button("Calcular Fardo", key="btn\_fardo"):

if (cristal + demerara) != total\_fardo:

st.error("Soma incorreta!")

else:

calcular\_distribuicao\_fardos(total\_fardo, lotes\_fardo, cristal, demerara)

with tab2:

st.header("Cálculo Apenas Fardos Cristal")

total\_cristal = st.number\_input("Total Fardos:", min\_value=1, step=1, value=1070, key="total\_cristal")

lotes\_cristal = st.number\_input("Nº Lotes:", min\_value=0.5, step=0.5, value=10.5, key="lotes\_cristal")

if st.button("Calcular Fardos Cristal", key="btn\_cristal"):

calcular\_distribuicao\_fardos(total\_cristal, lotes\_cristal)

with tab3:

st.header("Cálculo Apenas Sacos")

total\_sacos = st.number\_input("Total Sacos:", min\_value=1, step=1, value=300, key="total\_sacos")

lotes\_sacos = st.number\_input("Nº Lotes Sacos:", min\_value=0.5, step=0.5, value=11.0, key="lotes\_sacos")

if st.button("Calcular Sacos", key="btn\_sacos"):

calcular\_sacos(total\_sacos, lotes\_sacos)

st.markdown("---")

st.markdown("\*\*Desenvolvido por Lucas Cardoso\*\*")

aqui fiz ajustes meu

import streamlit as st

import pandas as pd

# Configuração

st.set\_page\_config(*page\_title*="Sistema de Cargas CORRETO", *layout*="centered")

# Funções de cálculo

def calcular\_distribuicao\_fardos(*total*, *num\_lotes*, *cristal*=None, *demerara*=None):

    lotes\_inteiros = int(*num\_lotes*)

    tem\_meio\_lote = *num\_lotes* % 1 != 0

    if tem\_meio\_lote:

        tamanho\_lote = int(*total* // (lotes\_inteiros + 0.5))

        resto = *total* - (lotes\_inteiros \* tamanho\_lote)

    else:

        tamanho\_lote = *total* // lotes\_inteiros

        resto = *total* % lotes\_inteiros

    primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

    outros\_lotes = tamanho\_lote

    dados = []

    if *cristal* is not None:  # Caso Cristal + Demerara

        cristal\_restante = *cristal*

        if not tem\_meio\_lote and cristal\_restante >= primeiro\_lote:

            dados.append(["Lote 1", f"12\*{primeiro\_lote//12}+{primeiro\_lote%12}", "0", primeiro\_lote])

            cristal\_restante -= primeiro\_lote

        elif not tem\_meio\_lote:

            demerara\_lote1 = primeiro\_lote - cristal\_restante

            dados.append(["Lote 1",

                         f"12\*{cristal\_restante//12}+{cristal\_restante%12}",

                         f"12\*{demerara\_lote1//12}+{demerara\_lote1%12}",

                         primeiro\_lote])

            cristal\_restante = 0

        for i in range(1 if tem\_meio\_lote else 0, lotes\_inteiros + (1 if tem\_meio\_lote else 0)):

            if cristal\_restante <= 0:

                dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                             "0",

                             f"12\*{outros\_lotes//12}+{outros\_lotes%12}",

                             outros\_lotes])

            else:

                if cristal\_restante >= outros\_lotes:

                    dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                                 f"12\*{outros\_lotes//12}+{outros\_lotes%12}",

                                 "0",

                                 outros\_lotes])

                    cristal\_restante -= outros\_lotes

                else:

                    demerara\_lote = outros\_lotes - cristal\_restante

                    dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                                 f"12\*{cristal\_restante//12}+{cristal\_restante%12}",

                                 f"12\*{demerara\_lote//12}+{demerara\_lote%12}",

                                 outros\_lotes])

                    cristal\_restante = 0

        if tem\_meio\_lote:

            dados.append(["Meio lote", "0", f"6\*{resto//6}+{resto%6}", resto])

        demerara\_distribuido = *total* - (*cristal* - cristal\_restante)

        if cristal\_restante == 0 and demerara\_distribuido == *demerara*:

            st.success(f"✓ CERTO! Cristal: {*cristal*} | Demerara: {*demerara*} | Total: {*total*}")

        else:

            st.error(f"ERRO! Cristal: {*cristal*-cristal\_restante}/{*cristal*} | Demerara: {demerara\_distribuido}/{*demerara*}")

        df = pd.DataFrame(dados, *columns*=["Lote", "Cristal", "Demerara", "Total"])

    else:  # Caso apenas Fardos Cristal

        if not tem\_meio\_lote:

            dados.append(["Lote 1", f"12\*{(primeiro\_lote//12)}+{primeiro\_lote%12}", primeiro\_lote])

            for i in range(1, lotes\_inteiros):

                dados.append([f"Lote {i+1}", f"12\*{outros\_lotes//12}+{outros\_lotes%12}", outros\_lotes])

        else:

            for i in range(lotes\_inteiros):

                dados.append([f"Lote {i+1}", f"12\*{outros\_lotes//12}+{outros\_lotes%12}", outros\_lotes])

            dados.append(["Meio lote", f"6\*{resto//6}+{resto%6}", resto])

        df = pd.DataFrame(dados, *columns*=["Lote", "Fardos (12)", "Total"])

    st.write(df.to\_html(*index*=False), *unsafe\_allow\_html*=True)

def calcular\_sacos(*total\_sacos*, *num\_lotes\_sacos*):

    lotes\_inteiros = int(*num\_lotes\_sacos*)

    tem\_meio\_lote = *num\_lotes\_sacos* % 1 != 0

    if tem\_meio\_lote:

        tamanho\_lote = int(*total\_sacos* // (lotes\_inteiros + 0.5))

        resto = *total\_sacos* - (lotes\_inteiros \* tamanho\_lote)

    else:

        tamanho\_lote = *total\_sacos* // lotes\_inteiros

        resto = *total\_sacos* % lotes\_inteiros

    primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

    outros\_lotes = tamanho\_lote

    dados = []

    if not tem\_meio\_lote:

        dados.append(["Lote 1", f"5\*{(primeiro\_lote//5)}+{primeiro\_lote%5}", primeiro\_lote])

        for i in range(1, lotes\_inteiros):

            dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

    else:

        for i in range(lotes\_inteiros):

            dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

        dados.append(["Meio lote", f"5\*{resto//5}+{resto%5}", resto])

    df = pd.DataFrame(dados, *columns*=["Lote", "Sacos (5)", "Total"])

    st.write(df.to\_html(*index*=False), *unsafe\_allow\_html*=True)

# Interface

st.title("📦 CALCULADORA DE CARGAS ARMAZÉM 2025")

tab1, tab2, tab3 = st.tabs(["Fardo Cristal + Demerara", "Apenas Fardos Cristal", "Apenas Sacos"])

with tab1:

    st.header("Cálculo Fardo Cristal + Demerara")

    total\_fardo = st.number\_input("Total:", *min\_value*=1, *step*=1, *value*=1070, *key*="total\_fardo")

    lotes\_fardo = st.number\_input("Nº Lotes:", *min\_value*=0.5, *step*=0.5, *value*=10.5, *key*="lotes\_fardo")

    cristal = st.number\_input("Cristal:", *min\_value*=0, *step*=1, *value*=500, *key*="cristal")

    demerara = st.number\_input("Demerara:", *min\_value*=0, *step*=1, *value*=570, *key*="demerara")

    if st.button("Calcular Fardo", *key*="btn\_fardo"):

        if (cristal + demerara) != total\_fardo:

            st.error("Soma incorreta!")

        else:

            calcular\_distribuicao\_fardos(total\_fardo, lotes\_fardo, cristal, demerara)

with tab2:

    st.header("Cálculo Apenas Fardos Cristal")

    total\_cristal = st.number\_input("Total Fardos:", *min\_value*=1, *step*=1, *value*=1070, *key*="total\_cristal")

    lotes\_cristal = st.number\_input("Nº Lotes:", *min\_value*=0.5, *step*=0.5, *value*=10.5, *key*="lotes\_cristal")

    if st.button("Calcular Fardos Cristal", *key*="btn\_cristal"):

        calcular\_distribuicao\_fardos(total\_cristal, lotes\_cristal)

with tab3:

    st.header("Cálculo Apenas Sacos")

    total\_sacos = st.number\_input("Total Sacos:", *min\_value*=1, *step*=1, *value*=300, *key*="total\_sacos")

    lotes\_sacos = st.number\_input("Nº Lotes Sacos:", *min\_value*=0.5, *step*=0.5, *value*=11.0, *key*="lotes\_sacos")

    if st.button("Calcular Sacos", *key*="btn\_sacos"):

        calcular\_sacos(total\_sacos, lotes\_sacos)

st.markdown("---")

st.markdown("\*\*Desenvolvido por Lucas Cardoso dia 15/04/2025\*\*")

st.markdown("Sistema de calculo de cargas armazém agrovale 2025/2026")

CODIGO FINALIZADO AGORA E SO FAZER O DEPLOY

import streamlit as st

import pandas as pd

# Configuração da pagina

st.set\_page\_config(*page\_title*="calculadora armazem", *layout*="centered")

# Funções de cálculo (aqui faço as funçoes)

def calcular\_distribuicao\_fardos(*total*, *num\_lotes*, *cristal*=None, *demerara*=None):

    lotes\_inteiros = int(*num\_lotes*)

    tem\_meio\_lote = *num\_lotes* % 1 != 0

    if tem\_meio\_lote:

        tamanho\_lote = int(*total* // (lotes\_inteiros + 0.5))

        resto = *total* - (lotes\_inteiros \* tamanho\_lote)

    else:

        tamanho\_lote = *total* // lotes\_inteiros

        resto = *total* % lotes\_inteiros

    primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

    outros\_lotes = tamanho\_lote

    dados = []

    if *cristal* is not None:  # Caso Cristal + Demerara

        cristal\_restante = *cristal*

        if not tem\_meio\_lote and cristal\_restante >= primeiro\_lote:

            dados.append(["Lote 1", f"12\*{primeiro\_lote//12}+{primeiro\_lote%12}", "0", primeiro\_lote])

            cristal\_restante -= primeiro\_lote

        elif not tem\_meio\_lote:

            demerara\_lote1 = primeiro\_lote - cristal\_restante

            dados.append(["Lote 1",

                         f"12\*{cristal\_restante//12}+{cristal\_restante%12}",

                         f"12\*{demerara\_lote1//12}+{demerara\_lote1%12}",

                         primeiro\_lote])

            cristal\_restante = 0

        for i in range(1 if tem\_meio\_lote else 0, lotes\_inteiros + (1 if tem\_meio\_lote else 0)):

            if cristal\_restante <= 0:

                dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                             "0",

                             f"12\*{outros\_lotes//12}+{outros\_lotes%12}",

                             outros\_lotes])

            else:

                if cristal\_restante >= outros\_lotes:

                    dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                                 f"12\*{outros\_lotes//12}+{outros\_lotes%12}",

                                 "0",

                                 outros\_lotes])

                    cristal\_restante -= outros\_lotes

                else:

                    demerara\_lote = outros\_lotes - cristal\_restante

                    dados.append([f"Lote {i+1}" if not tem\_meio\_lote else f"Lote {i}",

                                 f"12\*{cristal\_restante//12}+{cristal\_restante%12}",

                                 f"12\*{demerara\_lote//12}+{demerara\_lote%12}",

                                 outros\_lotes])

                    cristal\_restante = 0

        if tem\_meio\_lote:

            dados.append(["Meio lote", "0", f"6\*{resto//6}+{resto%6}", resto])

        demerara\_distribuido = *total* - (*cristal* - cristal\_restante)

        if cristal\_restante == 0 and demerara\_distribuido == *demerara*:

            st.success(f"✓ CERTO! Cristal: {*cristal*} | Demerara: {*demerara*} | Total: {*total*}")

        else:

            st.error(f"ERRO! Cristal: {*cristal*-cristal\_restante}/{*cristal*} | Demerara: {demerara\_distribuido}/{*demerara*}")

        df = pd.DataFrame(dados, *columns*=["Lote", "Cristal", "Demerara", "Total"])

    else:  # Caso apenas Fardos Cristal (nesse caso so pra calculo do fardo cristal)

        if not tem\_meio\_lote:

            dados.append(["Lote 1", f"12\*{(primeiro\_lote//12)}+{primeiro\_lote%12}", primeiro\_lote])

            for i in range(1, lotes\_inteiros):

                dados.append([f"Lote {i+1}", f"12\*{outros\_lotes//12}+{outros\_lotes%12}", outros\_lotes])

        else:

            for i in range(lotes\_inteiros):

                dados.append([f"Lote {i+1}", f"12\*{outros\_lotes//12}+{outros\_lotes%12}", outros\_lotes])

            dados.append(["Meio lote", f"6\*{resto//6}+{resto%6}", resto])

        df = pd.DataFrame(dados, *columns*=["Lote", "Fardos (12)", "Total"])

    st.write(df.to\_html(*index*=False), *unsafe\_allow\_html*=True)

def calcular\_sacos(*total\_sacos*, *num\_lotes\_sacos*):

    lotes\_inteiros = int(*num\_lotes\_sacos*)

    tem\_meio\_lote = *num\_lotes\_sacos* % 1 != 0

    if tem\_meio\_lote:

        tamanho\_lote = int(*total\_sacos* // (lotes\_inteiros + 0.5))

        resto = *total\_sacos* - (lotes\_inteiros \* tamanho\_lote)

    else:

        tamanho\_lote = *total\_sacos* // lotes\_inteiros

        resto = *total\_sacos* % lotes\_inteiros

    primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

    outros\_lotes = tamanho\_lote

    dados = []

    if not tem\_meio\_lote:

        dados.append(["Lote 1", f"5\*{(primeiro\_lote//5)}+{primeiro\_lote%5}", primeiro\_lote])

        for i in range(1, lotes\_inteiros):

            dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

    else:

        for i in range(lotes\_inteiros):

            dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

        dados.append(["Meio lote", f"5\*{resto//5}+{resto%5}", resto])

    df = pd.DataFrame(dados, *columns*=["Lote", "Sacos (5)", "Total"])

    st.write(df.to\_html(*index*=False), *unsafe\_allow\_html*=True)

# qui vou da inicio com o titalo ( talvez eu mude depois )

st.title("🚚 CALCULADORA DE CARGAS ARMAZÉM 2025")

tab1, tab2, tab3 = st.tabs(["Fardo Cristal + Demerara", "Apenas Fardos Cristal", "Apenas Sacos"])

with tab1:

    st.header("Cálculo Fardo Cristal + Demerara")

    total\_fardo = st.number\_input("Total:", *min\_value*=1, *step*=1, *value*=1070, *key*="total\_fardo")

    lotes\_fardo = st.number\_input("Nº Lotes:", *min\_value*=0.5, *step*=0.5, *value*=10.5, *key*="lotes\_fardo")

    cristal = st.number\_input("Cristal:", *min\_value*=0, *step*=1, *value*=500, *key*="cristal")

    demerara = st.number\_input("Demerara:", *min\_value*=0, *step*=1, *value*=570, *key*="demerara")

    if st.button("Calcular Fardo", *key*="btn\_fardo"):

        if (cristal + demerara) != total\_fardo:

            st.error("Soma incorreta!")

        else:

            calcular\_distribuicao\_fardos(total\_fardo, lotes\_fardo, cristal, demerara)

with tab2:

    st.header("Cálculo Apenas Fardos Cristal")

    total\_cristal = st.number\_input("Total Fardos:", *min\_value*=1, *step*=1, *value*=1070, *key*="total\_cristal")

    lotes\_cristal = st.number\_input("Nº Lotes:", *min\_value*=0.5, *step*=0.5, *value*=10.5, *key*="lotes\_cristal")

    if st.button("Calcular Fardos Cristal", *key*="btn\_cristal"):

        calcular\_distribuicao\_fardos(total\_cristal, lotes\_cristal)

with tab3:

    st.header("Cálculo Apenas Sacos")

    total\_sacos = st.number\_input("Total Sacos:", *min\_value*=1, *step*=1, *value*=300, *key*="total\_sacos")

    lotes\_sacos = st.number\_input("Nº Lotes Sacos:", *min\_value*=0.5, *step*=0.5, *value*=11.0, *key*="lotes\_sacos")

    if st.button("Calcular Sacos", *key*="btn\_sacos"):

        calcular\_sacos(total\_sacos, lotes\_sacos)

st.markdown("---")

st.markdown("\*\*Desenvolvido por Lucas Cardoso dia 15/04/2025\*\*")

st.markdown("Sistema de calculo de cargas armazém agrovale 2025/2026")

#sistema simples concluido por mim lucas cardoso de freitas na data do dia 15 de abril de 2025 vou considerara esse app como a versao 1.0 depois posso estar fazendo alguns ajustes e melhorando o codigo. lembrando que desemvolvi todo em streamlit execuçao= streamlit run app.py

as somas não estava batendo então fiz ajustes

import streamlit as st

import pandas as pd

# nesta aba iniciarei as configuraçoes de pagina

st.set\_page\_config(*page\_title*="calculadora armazem", *layout*="centered")

# Funções de cálculo ( parte de distribuição)

def calcular\_distribuicao\_fardos(*total*, *num\_lotes*, *cristal*=None, *demerara*=None):

    lotes\_inteiros = int(*num\_lotes*)

    tem\_meio\_lote = *num\_lotes* % 1 != 0

    if tem\_meio\_lote:

        tamanho\_lote = int(*total* // (lotes\_inteiros + 0.5))

        resto = *total* - (lotes\_inteiros \* tamanho\_lote)

    else:

        # aqui e para somas de lotes inteiros

        tamanho\_lote = *total* // lotes\_inteiros

        resto = *total* % lotes\_inteiros

    dados = []

    if *cristal* is not None:  # Caso Cristal + Demerara

        cristal\_restante = *cristal*

        demerara\_restante = *demerara* if *demerara* is not None else 0

        if not tem\_meio\_lote:

            #(distribuição para lotes inteiros)

            valor\_base = *total* // lotes\_inteiros

            resto = *total* % lotes\_inteiros

            # Distribui primeiro todo o cristal

            for i in range(lotes\_inteiros):

                # Primeiro lote leva o resto de fardos , a sobra

                if i == 0:

                    lote\_atual = valor\_base + resto

                else:

                    lote\_atual = valor\_base

                if cristal\_restante > 0:

                    if cristal\_restante >= lote\_atual:

                        dados.append([f"Lote {i+1}", f"12\*{lote\_atual//12}+{lote\_atual%12}", "0", lote\_atual])

                        cristal\_restante -= lote\_atual

                    else:

                        demerara\_no\_lote = lote\_atual - cristal\_restante

                        dados.append([f"Lote {i+1}",

                                    f"12\*{cristal\_restante//12}+{cristal\_restante%12}",

                                    f"12\*{demerara\_no\_lote//12}+{demerara\_no\_lote%12}",

                                    lote\_atual])

                        cristal\_restante = 0

                        demerara\_restante -= demerara\_no\_lote

                else:

                    dados.append([f"Lote {i+1}", "0", f"12\*{lote\_atual//12}+{lote\_atual%12}", lote\_atual])

                    demerara\_restante -= lote\_atual

        else:

            # Lógica para meia ( vou chamar de meio lote)

            for i in range(lotes\_inteiros + 1):

                if i < lotes\_inteiros:

                    lote\_atual = tamanho\_lote

                    lote\_nome = f"Lote {i+1}"

                else:

                    lote\_atual = resto

                    lote\_nome = "Meio lote"

                if cristal\_restante > 0:

                    if cristal\_restante >= lote\_atual:

                        dados.append([lote\_nome, f"12\*{lote\_atual//12}+{lote\_atual%12}", "0", lote\_atual])

                        cristal\_restante -= lote\_atual

                    else:

                        demerara\_no\_lote = lote\_atual - cristal\_restante

                        dados.append([lote\_nome,

                                    f"12\*{cristal\_restante//12}+{cristal\_restante%12}",

                                    f"12\*{demerara\_no\_lote//12}+{demerara\_no\_lote%12}" if i < lotes\_inteiros else f"6\*{demerara\_no\_lote//6}+{demerara\_no\_lote%6}",

                                    lote\_atual])

                        cristal\_restante = 0

                        demerara\_restante -= demerara\_no\_lote

                else:

                    dados.append([lote\_nome, "0",

                                f"12\*{lote\_atual//12}+{lote\_atual%12}" if i < lotes\_inteiros else f"6\*{lote\_atual//6}+{lote\_atual%6}",

                                lote\_atual])

                    demerara\_restante -= lote\_atual

        # verificando as informaçoes finais

        total\_calculado = sum([row[3] for row in dados])

        cristal\_distribuido = *cristal* - cristal\_restante

        demerara\_distribuido = (*demerara* - demerara\_restante) if *demerara* is not None else 0

        if total\_calculado == *total* and cristal\_distribuido == *cristal* and (*demerara* is None or demerara\_distribuido == *demerara*):

            st.success(f"✓ Distribuição correta! Total: {total\_calculado}")

        else:

            st.error(f"ERRO! Faltou distribuir: Cristal: {cristal\_restante}/{*cristal*} | Demerara: {demerara\_restante}/{*demerara*}")

        df = pd.DataFrame(dados, *columns*=["Lote", "Cristal", "Demerara", "Total"])

    else:  # Caso de soma para apenas fardo cristal de 30kg

        if not tem\_meio\_lote:

            dados.append(["Lote 1", f"12\*{(tamanho\_lote + resto)//12}+{(tamanho\_lote + resto)%12}", tamanho\_lote + resto])

            for i in range(1, lotes\_inteiros):

                dados.append([f"Lote {i+1}", f"12\*{tamanho\_lote//12}+{tamanho\_lote%12}", tamanho\_lote])

        else:

            for i in range(lotes\_inteiros):

                dados.append([f"Lote {i+1}", f"12\*{tamanho\_lote//12}+{tamanho\_lote%12}", tamanho\_lote])

            dados.append(["Meio lote", f"6\*{resto//6}+{resto%6}", resto])

        df = pd.DataFrame(dados, *columns*=["Lote", "Fardos (12)", "Total"])

    st.write(df.to\_html(*index*=False), *unsafe\_allow\_html*=True)

# funçoes

def calcular\_sacos(*total\_sacos*, *num\_lotes\_sacos*):

    lotes\_inteiros = int(*num\_lotes\_sacos*)

    tem\_meio\_lote = *num\_lotes\_sacos* % 1 != 0

    if tem\_meio\_lote:

        tamanho\_lote = int(*total\_sacos* // (lotes\_inteiros + 0.5))

        resto = *total\_sacos* - (lotes\_inteiros \* tamanho\_lote)

    else:

        tamanho\_lote = *total\_sacos* // lotes\_inteiros

        resto = *total\_sacos* % lotes\_inteiros

    primeiro\_lote = tamanho\_lote + (resto if not tem\_meio\_lote else 0)

    outros\_lotes = tamanho\_lote

    dados = []

    if not tem\_meio\_lote:

        dados.append(["Lote 1", f"5\*{(primeiro\_lote//5)}+{primeiro\_lote%5}", primeiro\_lote])

        for i in range(1, lotes\_inteiros):

            dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

    else:

        for i in range(lotes\_inteiros):

            dados.append([f"Lote {i+1}", f"5\*{outros\_lotes//5}+{outros\_lotes%5}", outros\_lotes])

        dados.append(["Meio lote", f"5\*{resto//5}+{resto%5}", resto])

    df = pd.DataFrame(dados, *columns*=["Lote", "Sacos (5)", "Total"])

    st.write(df.to\_html(*index*=False), *unsafe\_allow\_html*=True)

# interface e titalo do app calculadora armazém

st.title("🚚 CALCULADORA DE CARGAS ARMAZÉM 2025")

tab1, tab2, tab3 = st.tabs(["Fardo Cristal + Demerara", "Apenas Fardos Cristal", "Apenas Sacos"])

with tab1:

    st.header("Cálculo Fardo Cristal + Demerara")

    total\_fardo = st.number\_input("Total:", *min\_value*=1, *step*=1, *value*=1070, *key*="total\_fardo")

    lotes\_fardo = st.number\_input("Nº Lotes:", *min\_value*=0.5, *step*=0.5, *value*=10.5, *key*="lotes\_fardo")

    cristal = st.number\_input("Cristal:", *min\_value*=0, *step*=1, *value*=500, *key*="cristal")

    demerara = st.number\_input("Demerara:", *min\_value*=0, *step*=1, *value*=570, *key*="demerara")

    if st.button("Calcular Fardo", *key*="btn\_fardo"):

        if (cristal + demerara) != total\_fardo:

            st.error("Soma incorreta!")

        else:

            calcular\_distribuicao\_fardos(total\_fardo, lotes\_fardo, cristal, demerara)

with tab2:

    st.header("Cálculo Apenas Fardos Cristal")

    total\_cristal = st.number\_input("Total Fardos:", *min\_value*=1, *step*=1, *value*=1070, *key*="total\_cristal")

    lotes\_cristal = st.number\_input("Nº Lotes:", *min\_value*=0.5, *step*=0.5, *value*=10.5, *key*="lotes\_cristal")

    if st.button("Calcular Fardos Cristal", *key*="btn\_cristal"):

        calcular\_distribuicao\_fardos(total\_cristal, lotes\_cristal)

with tab3:

    st.header("Cálculo Apenas Sacos")

    total\_sacos = st.number\_input("Total Sacos:", *min\_value*=1, *step*=1, *value*=300, *key*="total\_sacos")

    lotes\_sacos = st.number\_input("Nº Lotes Sacos:", *min\_value*=0.5, *step*=0.5, *value*=11.0, *key*="lotes\_sacos")

    if st.button("Calcular Sacos", *key*="btn\_sacos"):

        calcular\_sacos(total\_sacos, lotes\_sacos)

st.markdown("---")

st.markdown("\*\*Desenvolvido por Lucas Cardoso dia 15/04/2025\*\*")

st.markdown("Sistema de calculo de cargas armazém agrovale 2025/2026")

# codigo alterado, corrigindo erros de calculos, vou considerar ainda essa como a versão 1.0 , data da ultima alteração 16/04/2025. para executar : streamlit run app.py. esse sistema ja esta no meu repositorio da git telefone (11 9 99454425 )