# Project Structure:

MegaSenaConferidor/

├── requirements.txt

├── setup.ps1

├── static/

│ ├── css/

│ │ └── style.css

│ └── js/

│ └── main.js

├── templates/

│ └── index.html

└── app.py

# requirements.txt

flask==2.0.3

Werkzeug==2.0.3

requests==2.26.0

python-dotenv==0.19.0

# app.py

from flask import Flask, render\_template, request, jsonify

import requests

from datetime import datetime

app = Flask(\_\_name\_\_)

API\_BASE\_URL = "https://loteriascaixa-api.herokuapp.com/api"

def get\_latest\_result():

try:

response = requests.get(f"{API\_BASE\_URL}/megasena/latest")

if response.status\_code == 200:

return response.json()

return None

except Exception as e:

print(f"Erro ao buscar último resultado: {str(e)}")

return None

@app.route('/')

def index():

latest = get\_latest\_result()

ultimo\_concurso = latest['concurso'] if latest else 2680

return render\_template('index.html', ultimo\_concurso=ultimo\_concurso)

@app.route('/gerar\_numeros')

def gerar\_numeros():

numeros = random.sample(range(1, 61), 6)

return jsonify({'numeros': sorted(numeros)})

@app.route('/conferir', methods=['POST'])

def conferir():

try:

data = request.get\_json()

inicio = int(data['inicio'])

fim = int(data['fim'])

jogos = data['jogos']

resultados = {

'acertos': [],

'resumo': {

'quatro': 0,

'cinco': 0,

'seis': 0

}

}

# Buscar resultados para cada concurso

for concurso in range(inicio, fim + 1):

try:

# Fazer requisição para a API

response = requests.get(f"{API\_BASE\_URL}/megasena/{concurso}", timeout=10)

# Verificar se a requisição foi bem sucedida

if response.status\_code != 200:

print(f"Erro ao buscar concurso {concurso}: Status {response.status\_code}")

continue

resultado = response.json()

# Converter dezenas para inteiros

dezenas = [int(d) for d in resultado['dezenas']]

# Verificar cada jogo contra este resultado

for jogo in jogos:

acertos = len(set(jogo) & set(dezenas))

if acertos >= 4:

# Buscar informações de premiação

premio = 0

for premiacao in resultado['premiacoes']:

if (acertos == 6 and premiacao['descricao'] == '6 acertos') or \

(acertos == 5 and premiacao['descricao'] == '5 acertos') or \

(acertos == 4 and premiacao['descricao'] == '4 acertos'):

premio = premiacao['valorPremio']

if acertos == 4:

resultados['resumo']['quatro'] += 1

elif acertos == 5:

resultados['resumo']['cinco'] += 1

elif acertos == 6:

resultados['resumo']['seis'] += 1

# Adicionar resultado detalhado

resultados['acertos'].append({

'concurso': resultado['concurso'],

'data': resultado['data'],

'local': resultado['local'],

'numeros\_sorteados': dezenas,

'numeros\_ordem\_sorteio': [int(d) for d in resultado['dezenasOrdemSorteio']],

'seus\_numeros': jogo,

'acertos': acertos,

'premio': premio,

'premiacoes': resultado['premiacoes'],

'acumulou': resultado['acumulou'],

'local\_ganhadores': resultado.get('localGanhadores', [])

})

except requests.exceptions.RequestException as e:

print(f"Erro na requisição para concurso {concurso}: {str(e)}")

continue

except Exception as e:

print(f"Erro ao processar concurso {concurso}: {str(e)}")

continue

return jsonify(resultados)

except Exception as e:

print(f"Erro geral na conferência: {str(e)}")

return jsonify({'error': str(e)}), 500

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>MegaSena Conferidor</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

</head>

<body>

<div class="container">

<h1>Conferidor MegaSena</h1>

<div class="volante">

<h2>Selecione 6 números</h2>

<div class="numeros">

{% for i in range(1, 61) %}

<button type="button" class="numero" data-numero="{{ i }}">{{ '%02d' % i }}</button>

{% endfor %}

</div>

</div>

<div class="controles">

<div class="input-group">

<label for="inicio">Concurso Inicial:</label>

<input type="number" id="inicio" min="1">

</div>

<div class="input-group">

<label for="fim">Concurso Final:</label>

<input type="number" id="fim" min="1" value="{{ ultimo\_concurso }}">

</div>

<div class="checkbox-group">

<input type="checkbox" id="somente-premiados">

<label for="somente-premiados">Somente premiados</label>

</div>

</div>

<div class="acoes">

<button id="incluir" class="btn btn-azul">Incluir Jogo</button>

<button id="limpar" class="btn btn-vermelho">Limpar</button>

<button id="sugestao" class="btn btn-roxo">Palpite</button>

<button id="conferir" class="btn btn-verde">Conferir</button>

</div>

<div class="jogos-incluidos">

<h3>Jogos Incluídos</h3>

<div id="lista-jogos" class="lista-jogos"></div>

</div>

<div id="overlay" class="overlay" style="display: none;">

<div class="progress">

<div class="progress-bar"></div>

<div class="progress-text">Conferindo jogos...</div>

</div>

</div>

<div class="resultados">

<h2>Resumo de acertos</h2>

<div class="cards">

<div class="card">

<h3>4 Acertos</h3>

<p class="contagem" id="quatro-acertos">0</p>

</div>

<div class="card">

<h3>5 Acertos</h3>

<p class="contagem" id="cinco-acertos">0</p>

</div>

<div class="card">

<h3>6 Acertos</h3>

<p class="contagem" id="seis-acertos">0</p>

</div>

</div>

<div id="detalhes-resultados" class="detalhes-resultados"></div>

</div>

</div>

<script src="{{ url\_for('static', filename='js/main.js') }}"></script>

</body>

</html>

// static/js/main.js

document.addEventListener('DOMContentLoaded', function() {

const numeros = document.querySelectorAll('.numero');

const limparBtn = document.getElementById('limpar');

const sugestaoBtn = document.getElementById('sugestao');

const conferirBtn = document.getElementById('conferir');

const incluirBtn = document.getElementById('incluir');

const listaJogos = document.getElementById('lista-jogos');

const overlay = document.getElementById('overlay');

const numerosSelecionados = new Set();

const jogosIncluidos = [];

// Funcionalidade de seleção de números

numeros.forEach(numero => {

numero.addEventListener('click', () => {

const num = parseInt(numero.dataset.numero);

if (numero.classList.contains('selecionado')) {

numero.classList.remove('selecionado');

numerosSelecionados.delete(num);

} else if (numerosSelecionados.size < 6) {

numero.classList.add('selecionado');

numerosSelecionados.add(num);

}

});

});

// Botão Limpar

limparBtn.addEventListener('click', () => {

numeros.forEach(numero => numero.classList.remove('selecionado'));

numerosSelecionados.clear();

});

// Botão Sugestão

sugestaoBtn.addEventListener('click', async () => {

const response = await fetch('/gerar\_numeros');

const data = await response.json();

limparBtn.click();

data.numeros.forEach(num => {

const numero = document.querySelector(`[data-numero="${num}"]`);

numero.classList.add('selecionado');

numerosSelecionados.add(num);

});

});

// Botão Incluir

incluirBtn.addEventListener('click', () => {

if (numerosSelecionados.size !== 6) {

alert('Selecione 6 números antes de incluir o jogo!');

return;

}

const numerosArray = Array.from(numerosSelecionados).sort((a, b) => a - b);

jogosIncluidos.push(numerosArray);

const jogoItem = document.createElement('div');

jogoItem.className = 'jogo-item';

const jogoNumeros = document.createElement('div');

jogoNumeros.className = 'jogo-numeros';

numerosArray.forEach(num => {

const numeroSpan = document.createElement('span');

numeroSpan.className = 'jogo-numero';

numeroSpan.textContent = String(num).padStart(2, '0');

jogoNumeros.appendChild(numeroSpan);

});

const btnRemover = document.createElement('button');

btnRemover.className = 'btn-remover';

btnRemover.textContent = 'Remover';

btnRemover.onclick = () => {

jogoItem.remove();

const index = jogosIncluidos.findIndex(jogo =>

JSON.stringify(jogo) === JSON.stringify(numerosArray)

);

if (index !== -1) {

jogosIncluidos.splice(index, 1);

}

};

jogoItem.appendChild(jogoNumeros);

jogoItem.appendChild(btnRemover);

listaJogos.appendChild(jogoItem);

limparBtn.click();

});

// Botão Conferir

//# static/js/main.js - Parte da conferência

conferirBtn.addEventListener('click', async () => {

if (jogosIncluidos.length === 0) {

alert('Inclua pelo menos um jogo antes de conferir!');

return;

}

const inicio = document.getElementById('inicio').value;

const fim = document.getElementById('fim').value;

if (!inicio || !fim || parseInt(inicio) > parseInt(fim)) {

alert('Verifique os números dos concursos!');

return;

}

// Mostrar overlay

overlay.style.display = 'flex';

document.querySelector('.progress-text').textContent = 'Conferindo jogos...';

try {

const response = await fetch('/conferir', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

inicio: parseInt(inicio),

fim: parseInt(fim),

jogos: jogosIncluidos

})

});

if (!response.ok) {

throw new Error(`Erro na requisição: ${response.status}`);

}

const resultados = await response.json();

if (resultados.error) {

throw new Error(resultados.error);

}

document.querySelector('.progress-text').textContent = 'Atualizando resultados...';

// Atualizar contagens

document.getElementById('quatro-acertos').textContent = resultados.resumo.quatro || 0;

document.getElementById('cinco-acertos').textContent = resultados.resumo.cinco || 0;

document.getElementById('seis-acertos').textContent = resultados.resumo.seis || 0;

// Atualizar detalhes

const detalhesDiv = document.getElementById('detalhes-resultados');

detalhesDiv.innerHTML = '';

const somentePremiados = document.getElementById('somente-premiados').checked;

if (resultados.acertos.length === 0) {

detalhesDiv.innerHTML = '<p class="sem-resultados">Nenhum prêmio encontrado para os jogos conferidos.</p>';

} else {

resultados.acertos

.sort((a, b) => b.concurso - a.concurso)

.forEach(resultado => {

if (!somentePremiados || resultado.acertos >= 4) {

const resultadoDiv = document.createElement('div');

resultadoDiv.className = 'resultado-item';

resultadoDiv.innerHTML = `

<div class="resultado-header">

<h3>Concurso ${resultado.concurso} - ${resultado.data}</h3>

<p>${resultado.local}</p>

</div>

<div class="resultado-numeros">

<div class="numeros-grupo">

<h4>Números Sorteados:</h4>

<div class="numeros-lista">

${resultado.numeros\_sorteados.map(n => `

<span class="numero-sorteado">${String(n).padStart(2, '0')}</span>

`).join('')}

</div>

</div>

<div class="numeros-grupo">

<h4>Seu Jogo:</h4>

<div class="numeros-lista">

${resultado.seus\_numeros.map(n => `

<span class="numero-jogado ${resultado.numeros\_sorteados.includes(n) ? 'acertou' : ''}">${String(n).padStart(2, '0')}</span>

`).join('')}

</div>

</div>

</div>

<div class="resultado-detalhes">

<p>Acertos: <strong>${resultado.acertos}</strong></p>

${resultado.premio > 0 ? `<p>Prêmio: <strong>R$ ${resultado.premio.toLocaleString('pt-BR', {minimumFractionDigits: 2})}</strong></p>` : ''}

<p>${resultado.acumulou ? 'ACUMULOU!' : 'Não Acumulou'}</p>

</div>

`;

detalhesDiv.appendChild(resultadoDiv);

}

});

}

} catch (error) {

console.error('Erro ao conferir jogos:', error);

alert(`Erro ao conferir jogos: ${error.message}`);

} finally {

overlay.style.display = 'none';

}

});

});

:root {

--primary-color: #008751;

--secondary-color: #005c36;

--danger-color: #dc3545;

--success-color: #28a745;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

line-height: 1.6;

padding: 20px;

background-color: #f4f4f4;

}

.container {

max-width: 1200px;

margin: 0 auto;

background-color: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.volante {

margin: 20px 0;

padding: 20px;

background-color: #f8f9fa;

border-radius: 8px;

text-align: center;

}

.volante h2 {

color: #495057;

margin-bottom: 20px;

font-size: 1.2rem;

}

.numeros {

display: grid;

grid-template-columns: repeat(10, 1fr);

gap: 8px;

max-width: 800px;

margin: 0 auto;

padding: 15px;

}

.numero {

width: 40px;

height: 40px;

border-radius: 50%;

border: 2px solid #ced4da;

background-color: white;

color: #495057;

font-weight: bold;

font-size: 16px;

cursor: pointer;

transition: all 0.2s ease;

display: flex;

align-items: center;

justify-content: center;

padding: 0;

margin: 0 auto;

}

.numero:hover {

background-color: #e9ecef;

border-color: #adb5bd;

}

.numero.selecionado {

background-color: #28a745;

color: white;

border-color: #28a745;

}

.controles {

display: flex;

gap: 20px;

margin: 20px 0;

padding: 20px;

background-color: #f8f9fa;

border-radius: 8px;

align-items: center;

}

.input-group {

display: flex;

flex-direction: column;

gap: 5px;

}

.input-group label {

font-weight: bold;

color: #495057;

}

.input-group input {

padding: 8px;

border: 1px solid #ced4da;

border-radius: 4px;

width: 120px;

}

.checkbox-group {

display: flex;

align-items: center;

gap: 5px;

}

.acoes {

display: flex;

gap: 10px;

margin: 20px 0;

justify-content: center;

}

.btn {

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

font-weight: bold;

transition: all 0.3s ease;

}

.btn:hover {

opacity: 0.9;

transform: translateY(-1px);

}

.btn-vermelho {

background-color: var(--danger-color);

color: white;

}

.btn-azul {

background-color: #007bff;

color: white;

}

.btn-roxo {

background-color: #6f42c1;

color: white;

}

.btn-verde {

background-color: var(--success-color);

color: white;

}

.cards {

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 20px;

margin-top: 20px;

}

.card {

padding: 20px;

border-radius: 8px;

background-color: #f8f9fa;

text-align: center;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);

}

.card h3 {

color: #495057;

margin-bottom: 10px;

}

.contagem {

font-size: 24px;

font-weight: bold;

color: var(--primary-color);

}

.lista-jogos {

margin-top: 20px;

border: 1px solid #ddd;

border-radius: 4px;

padding: 10px;

min-height: 100px;

background-color: #fff;

}

.jogo-item {

background-color: #f8f9fa;

padding: 10px;

margin: 5px 0;

border-radius: 4px;

display: flex;

justify-content: space-between;

align-items: center;

box-shadow: 0 1px 3px rgba(0, 0, 0, 0.1);

}

.jogo-numeros {

display: flex;

gap: 5px;

}

.jogo-numero {

background-color: var(--primary-color);

color: white;

width: 30px;

height: 30px;

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

font-size: 14px;

font-weight: bold;

}

.btn-remover {

background-color: var(--danger-color);

color: white;

border: none;

border-radius: 4px;

padding: 5px 10px;

cursor: pointer;

font-size: 12px;

}

.btn-remover:hover {

opacity: 0.9;

}

.overlay {

position: fixed;

top: 0;

left: 0;

right: 0;

bottom: 0;

background-color: rgba(0, 0, 0, 0.7);

display: flex;

justify-content: center;

align-items: center;

z-index: 1000;

backdrop-filter: blur(5px);

}

.progress {

background-color: white;

padding: 20px;

border-radius: 8px;

width: 300px;

text-align: center;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

.progress-bar {

height: 6px;

background-color: var(--primary-color);

border-radius: 3px;

margin: 10px 0;

position: relative;

overflow: hidden;

}

.progress-bar::after {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background: linear-gradient(

90deg,

transparent,

rgba(255, 255, 255, 0.4),

transparent

);

animation: shimmer 1s infinite;

}

.progress-text {

margin-top: 10px;

color: #495057;

font-weight: 500;

}

.detalhes-resultados {

margin-top: 20px;

}

@keyframes shimmer {

0% {

transform: translateX(-100%);

}

100% {

transform: translateX(100%);

}

}

/\*

@keyframes progress {

0% { width: 0%; }

50% { width: 100%; }

100% { width: 0%; }

}

\*/

.resultado-item {

background-color: #fff;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0,0,0,0.1);

margin-bottom: 20px;

padding: 20px;

}

.resultado-header {

border-bottom: 1px solid #eee;

margin-bottom: 15px;

padding-bottom: 10px;

}

.resultado-header h3 {

color: var(--primary-color);

margin-bottom: 5px;

}

.numeros-grupo {

margin: 15px 0;

}

.numeros-lista {

display: flex;

gap: 10px;

flex-wrap: wrap;

margin-top: 5px;

}

.numero-sorteado, .numero-jogado {

width: 40px;

height: 40px;

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

font-weight: bold;

background-color: #f0f0f0;

}

.numero-jogado.acertou {

background-color: var(--success-color);

color: white;

}

.premiacao-item {

display: grid;

grid-template-columns: 1fr 1fr 1fr;

gap: 10px;

padding: 10px;

background-color: #f8f9fa;

border-radius: 4px;

margin: 5px 0;

}

.ganhador-item {

padding: 10px;

background-color: #f8f9fa;

border-radius: 4px;

margin: 5px 0;

}

.resultado-detalhes {

margin: 15px 0;

padding: 15px;

background-color: #f8f9fa;

border-radius: 4px;

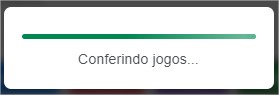
}

.resultado-detalhes p {

margin: 5px 0;

}

O PRBLEMA ESTA AQUI.. FICA PARADO E NÃO CONFERE NADA.. A CONFERÊNCIA PRECISA ACONTECER EM SEGUNDOS..



AQUI ESTÃO OS DADOS DA API

**Exemplos de Retorno**

Atualmente o banco de dados contém os jogos das loterias ...

<https://loteriascaixa-api.herokuapp.com/api>

[

"maismilionaria",

"megasena",

"lotofacil",

"quina",

"lotomania",

"timemania",

"duplasena",

"federal",

"diadesorte",

"supersete"

]

**Obtendo o Resultado Mais Recente**

URL BASE: https://loteriascaixa-api.herokuapp.com/api/<loteria>/latest

Apenas substitua <loteria> pelo nome da loteria desejada. Por exemplo, consultando pelo resultado da Mega-Sena (megasena) mais recente:

<https://loteriascaixa-api.herokuapp.com/api/megasena/latest>

{

"loteria": "megasena",

"concurso": 2620,

"data": "12/08/2023",

"local": "ESPAÇO DA SORTE em SÃO PAULO, SP",

"dezenasOrdemSorteio": [

"26",

"21",

"13",

"04",

"06",

"28"

],

"dezenas": [

"04",

"06",

"13",

"21",

"26",

"28"

],

"trevos": [

],

"timeCoracao": null,

"mesSorte": null,

"premiacoes": [

{

"descricao": "6 acertos",

"faixa": 1,

"ganhadores": 4,

"valorPremio": 29058128.28

},

{

"descricao": "5 acertos",

"faixa": 2,

"ganhadores": 404,

"valorPremio": 23042.04

},

{

"descricao": "4 acertos",

"faixa": 3,

"ganhadores": 21667,

"valorPremio": 613.76

}

],

"estadosPremiados": [

],

"observacao": "",

"acumulou": false,

"proximoConcurso": 2621,

"dataProximoConcurso": "16/08/2023",

"localGanhadores": [

{

"ganhadores": 1,

"municipio": "CANAL ELETRONICO",

"nomeFatansiaUL": "",

"serie": "",

"posicao": 1,

"uf": "--"

},

{

"ganhadores": 1,

"municipio": "UBERABA",

"nomeFatansiaUL": "",

"serie": "",

"posicao": 1,

"uf": "MG"

},

{

"ganhadores": 2,

"municipio": "SINOP",

"nomeFatansiaUL": "",

"serie": "",

"posicao": 1,

"uf": "MT"

}

],

"valorArrecadado": 161458740,

"valorAcumuladoConcurso\_0\_5": 10778824.03,

"valorAcumuladoConcursoEspecial": 72866210.44,

"valorAcumuladoProximoConcurso": 0.0,

"valorEstimadoProximoConcurso": 3500000.0

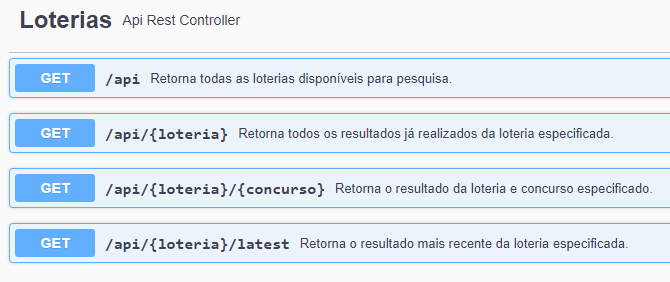
}

* **Observações**: Os campos ***timeCoracao*** e ***mesSorte*** só terão algum valor quando a loteria pesquisada for Timemania (timemania) ou Dia de Sorte (diadesorte) respectivamente.

**Documentação da API**

Para mais informações sobre todas as operações da API acesse:

<https://loteriascaixa-api.herokuapp.com/swagger-ui/#/Loterias>

[](https://user-images.githubusercontent.com/48946749/144352143-7140d64d-43a9-465c-b12c-7d5d3514ccd5.png)

VEJA O QUE VC PODE FAZER PRA ME AJUDAR..