<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

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

<title>Mapeamento Cultural Brasil Infra</title>

<script src="https://cdn.tailwindcss.com"></script>

</head>

<body class="bg-gradient-to-br from-blue-50 to-orange-50">

<div id="app"></div>

<script>

const words = [

*// COMPROMETIMENTO - Positivas*

{ word: 'Responsável', value: 'comprometimento', positive: true },

{ word: 'Confiável', value: 'comprometimento', positive: true },

{ word: 'Dedicado', value: 'comprometimento', positive: true },

{ word: 'Pontual', value: 'comprometimento', positive: true },

{ word: 'Leal', value: 'comprometimento', positive: true },

{ word: 'Persistente', value: 'comprometimento', positive: true },

{ word: 'Palavra é palavra', value: 'comprometimento', positive: true },

{ word: 'Teimoso', value: 'comprometimento', positive: false },

{ word: 'Inflexível', value: 'comprometimento', positive: false },

{ word: 'Centralizador', value: 'comprometimento', positive: false },

{ word: 'Autocrítico demais', value: 'comprometimento', positive: false },

*// TRABALHO SEGURO - Positivas*

{ word: 'Cauteloso', value: 'seguranca', positive: true },

{ word: 'Atento aos detalhes', value: 'seguranca', positive: true },

{ word: 'Previdente', value: 'seguranca', positive: true },

{ word: 'Organizado', value: 'seguranca', positive: true },

{ word: 'Disciplinado', value: 'seguranca', positive: true },

{ word: 'Metódico', value: 'seguranca', positive: true },

{ word: 'Cuidadoso', value: 'seguranca', positive: true },

{ word: 'Lento', value: 'seguranca', positive: false },

{ word: 'Burocrático', value: 'seguranca', positive: false },

{ word: 'Perfeccionista', value: 'seguranca', positive: false },

{ word: 'Resistente a mudanças', value: 'seguranca', positive: false },

*// AMBIENTE SAUDÁVEL - Positivas*

{ word: 'Empático', value: 'ambiente', positive: true },

{ word: 'Colaborativo', value: 'ambiente', positive: true },

{ word: 'Bom ouvinte', value: 'ambiente', positive: true },

{ word: 'Amigável', value: 'ambiente', positive: true },

{ word: 'Paciente', value: 'ambiente', positive: true },

{ word: 'Acolhedor', value: 'ambiente', positive: true },

{ word: 'Companheiro', value: 'ambiente', positive: true },

{ word: 'Evito conflitos', value: 'ambiente', positive: false },

{ word: 'Sensível demais', value: 'ambiente', positive: false },

{ word: 'Dependente de aprovação', value: 'ambiente', positive: false },

{ word: 'Indeciso', value: 'ambiente', positive: false },

*// FOCO NO RESULTADO - Positivas*

{ word: 'Objetivo', value: 'resultado', positive: true },

{ word: 'Prático', value: 'resultado', positive: true },

{ word: 'Competitivo', value: 'resultado', positive: true },

{ word: 'Determinado', value: 'resultado', positive: true },

{ word: 'Eficiente', value: 'resultado', positive: true },

{ word: 'Direto', value: 'resultado', positive: true },

{ word: 'Produtivo', value: 'resultado', positive: true },

{ word: 'Impaciente', value: 'resultado', positive: false },

{ word: 'Impulsivo', value: 'resultado', positive: false },

{ word: 'Insensível às vezes', value: 'resultado', positive: false },

{ word: 'Ansioso', value: 'resultado', positive: false },

*// MENTALIDADE DE CRESCIMENTO - Positivas*

{ word: 'Curioso', value: 'crescimento', positive: true },

{ word: 'Criativo', value: 'crescimento', positive: true },

{ word: 'Aberto a feedbacks', value: 'crescimento', positive: true },

{ word: 'Inovador', value: 'crescimento', positive: true },

{ word: 'Adaptável', value: 'crescimento', positive: true },

{ word: 'Estudioso', value: 'crescimento', positive: true },

{ word: 'Questionador', value: 'crescimento', positive: true },

{ word: 'Inquieto', value: 'crescimento', positive: false },

{ word: 'Disperso', value: 'crescimento', positive: false },

{ word: 'Inconformado', value: 'crescimento', positive: false },

{ word: 'Muitas ideias ao mesmo tempo', value: 'crescimento', positive: false },

];

const valueNames = {

comprometimento: 'Comprometimento',

seguranca: 'Trabalho Seguro',

ambiente: 'Ambiente Saudável',

resultado: 'Foco no Resultado',

crescimento: 'Mentalidade de Crescimento'

};

const valueColors = {

comprometimento: 'bg-blue-500',

seguranca: 'bg-red-500',

ambiente: 'bg-green-500',

resultado: 'bg-orange-500',

crescimento: 'bg-purple-500'

};

let state = {

step: 'intro',

phase: 1,

selectedWords1: [],

selectedWords2: [],

results: null,

userName: '',

userEmail: '',

shuffledWords: []

};

function shuffleArray(array) {

const newArray = [...array];

for (let i = newArray.length - 1; i > 0; i--) {

const j = Math.floor(Math.random() \* (i + 1));

[newArray[i], newArray[j]] = [newArray[j], newArray[i]];

}

return newArray;

}

state.shuffledWords = shuffleArray(words);

function toggleWord(word) {

if (state.phase === 1) {

if (state.selectedWords1.includes(word)) {

state.selectedWords1 = state.selectedWords1.filter(w => w !== word);

} else {

state.selectedWords1 = [...state.selectedWords1, word];

}

} else {

if (state.selectedWords2.includes(word)) {

state.selectedWords2 = state.selectedWords2.filter(w => w !== word);

} else {

state.selectedWords2 = [...state.selectedWords2, word];

}

}

render();

}

function calculateResults() {

const scores1 = { comprometimento: 0, seguranca: 0, ambiente: 0, resultado: 0, crescimento: 0 };

const scores2 = { comprometimento: 0, seguranca: 0, ambiente: 0, resultado: 0, crescimento: 0 };

state.selectedWords1.forEach(wordText => {

const word = words.find(w => w.word === wordText);

scores1[word.value] += word.positive ? 3 : 1;

});

state.selectedWords2.forEach(wordText => {

const word = words.find(w => w.word === wordText);

scores2[word.value] += word.positive ? 3 : 1;

});

const total1 = Object.values(scores1).reduce((a, b) => a + b, 0) || 1;

const total2 = Object.values(scores2).reduce((a, b) => a + b, 0) || 1;

const percentages1 = {};

const percentages2 = {};

Object.keys(scores1).forEach(key => {

percentages1[key] = Math.round((scores1[key] / total1) \* 100);

percentages2[key] = Math.round((scores2[key] / total2) \* 100);

});

const aderencia = calculateAdherence(percentages1, percentages2);

const energia = calculateEnergy(percentages1, percentages2);

state.results = {

natural: percentages1,

adaptado: percentages2,

aderencia,

energia

};

state.step = 'email';

render();

}

function calculateAdherence(natural, adaptado) {

const idealProfile = {

comprometimento: 22,

seguranca: 22,

resultado: 22,

crescimento: 18,

ambiente: 16

};

let totalDiff = 0;

Object.keys(idealProfile).forEach(key => {

const avgScore = (natural[key] + adaptado[key]) / 2;

totalDiff += Math.abs(avgScore - idealProfile[key]);

});

const aderencia = Math.max(0, 100 - (totalDiff / 2));

return Math.round(aderencia);

}

function calculateEnergy(natural, adaptado) {

let totalDiff = 0;

Object.keys(natural).forEach(key => {

totalDiff += Math.abs(natural[key] - adaptado[key]);

});

const energia = Math.max(0, 100 - totalDiff);

return Math.round(energia);

}

function nextPhase() {

if (state.phase === 1) {

state.phase = 2;

state.step = 'phase2';

} else {

calculateResults();

}

render();

}

async function sendResults() {

const formData = new FormData();

formData.append('name', state.userName);

formData.append('email', state.userEmail);

formData.append('message', generateEmailContent());

try {

const response = await fetch('https://formsubmit.co/sulyfarinazzo@brainfra.com.br', {

method: 'POST',

body: formData

});

state.step = 'results';

render();

} catch (error) {

alert('Erro ao enviar resultados. Tente novamente.');

}

}

function generateEmailContent() {

const { natural, adaptado, aderencia, energia } = state.results;

let content = `RESULTADOS DO MAPEAMENTO CULTURAL - Brasil Infraestrutura\n\n`;

content += `Nome: ${state.userName}\n`;

content += `Email: ${state.userEmail}\n`;

content += `Data: ${new Date().toLocaleDateString('pt-BR')}\n\n`;

content += `=== MÉTRICAS PRINCIPAIS ===\n`;

content += `Aderência Cultural: ${aderencia}%\n`;

content += `Nível de Energia: ${energia}%\n\n`;

content += `=== PERFIL NATURAL ===\n`;

Object.entries(natural).sort(([,a], [,b]) => b - a).forEach(([value, score]) => {

content += `${valueNames[value]}: ${score}%\n`;

});

content += `\n=== PERFIL ADAPTADO ===\n`;

Object.entries(adaptado).sort(([,a], [,b]) => b - a).forEach(([value, score]) => {

content += `${valueNames[value]}: ${score}%\n`;

});

content += `\n=== PALAVRAS SELECIONADAS ===\n`;

content += `Perfil Natural: ${state.selectedWords1.join(', ')}\n\n`;

content += `Perfil Adaptado: ${state.selectedWords2.join(', ')}\n`;

return content;

}

function renderIntro() {

return `

<div class="min-h-screen p-4 md:p-8 flex items-center justify-center">

<div class="max-w-2xl bg-white rounded-2xl shadow-2xl p-8">

<div class="text-center mb-8">

<div class="inline-flex items-center justify-center w-20 h-20 bg-gradient-to-r from-blue-600 to-orange-500 rounded-full mb-4">

<span class="text-5xl">🧠</span>

</div>

<h1 class="text-3xl font-bold text-gray-800 mb-2">Mapeamento de Perfil Cultural</h1>

<p class="text-xl text-gray-600">Brasil Infraestrutura</p>

</div>

<div class="bg-blue-50 rounded-xl p-6 mb-6">

<h2 class="font-bold text-lg text-gray-800 mb-3">Como funciona:</h2>

<ul class="space-y-2 text-gray-700">

<li class="flex items-start gap-2">

<span class="text-blue-600 font-bold">✓</span>

<span>São apenas 2 perguntas simples que levam cerca de 5 minutos</span>

</li>

<li class="flex items-start gap-2">

<span class="text-blue-600 font-bold">✓</span>

<span>Você verá uma lista de palavras e características</span>

</li>

<li class="flex items-start gap-2">

<span class="text-blue-600 font-bold">✓</span>

<span>Marque aquelas que realmente descrevem VOCÊ</span>

</li>

<li class="flex items-start gap-2">

<span class="text-blue-600 font-bold">✓</span>

<span>Seja honesto! Não há respostas certas ou erradas</span>

</li>

</ul>

</div>

<div class="bg-orange-50 rounded-xl p-6 mb-8">

<h2 class="font-bold text-lg text-gray-800 mb-3">⚠️ Importante:</h2>

<ul class="space-y-2 text-gray-700">

<li>• Não marque o que você <strong>gostaria</strong> de ser, mas sim o que você <strong>realmente é</strong></li>

<li>• Considere tanto pontos fortes quanto pontos de desenvolvimento</li>

<li>• Seja sincero - isso garantirá um resultado preciso e útil</li>

</ul>

</div>

<button onclick="startTest()" class="w-full bg-gradient-to-r from-blue-600 to-orange-500 text-white py-4 rounded-xl font-semibold hover:shadow-lg transition-all flex items-center justify-center gap-2">

Iniciar Mapeamento

<span>→</span>

</button>

</div>

</div>

`;

}

function renderPhase() {

const currentWords = state.phase === 1 ? state.selectedWords1 : state.selectedWords2;

const minWords = 10;

const maxWords = 35;

return `

<div class="min-h-screen p-4 md:p-8">

<div class="max-w-4xl mx-auto">

<div class="bg-white rounded-2xl shadow-2xl p-6 md:p-8">

<div class="mb-6">

<div class="flex justify-between items-center mb-4">

<span class="text-sm font-semibold text-gray-600">

Pergunta ${state.phase} de 2

</span>

<span class="text-sm font-semibold ${currentWords.length >= minWords ? 'text-green-600' : 'text-orange-600'}">

${currentWords.length} selecionadas ${currentWords.length >= minWords ? '✓' : `(mínimo: ${minWords})`}

</span>

</div>

</div>

<div class="bg-gradient-to-r from-blue-600 to-orange-500 rounded-xl p-6 text-white mb-6">

<h2 class="text-2xl font-bold mb-2">

${state.phase === 1 ? '1) Quem você REALMENTE é:' : '2) Como o TRABALHO te exige:'}

</h2>

<p class="text-lg opacity-90">

${state.phase === 1

? 'Marque as palavras que descrevem seu jeito de ser naturalmente, seu perfil genuíno.'

: 'Agora marque como você precisa se comportar no ambiente de trabalho, como a função te exige.'}

</p>

</div>

<div class="mb-6">

<p class="text-sm text-gray-600 bg-gray-50 p-4 rounded-lg">

💡 <strong>Dica:</strong> Seja honesto! Algumas palavras podem parecer "negativas", mas todos temos pontos fortes e fracos.

Aceitar suas características reais é fundamental para um resultado preciso.

${state.phase === 1 ? 'Selecione entre 10 e 35 palavras.' : ''}

</p>

</div>

<div class="grid grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-3 mb-8">

${state.shuffledWords.map((wordObj, index) => `

<button

onclick="toggleWord('${wordObj.word.replace(/'/g, "\\'")}')"

${!currentWords.includes(wordObj.word) && currentWords.length >= maxWords ? 'disabled' : ''}

class="p-3 rounded-lg text-sm font-medium transition-all ${

currentWords.includes(wordObj.word)

? 'bg-blue-600 text-white shadow-lg scale-105'

: currentWords.length >= maxWords

? 'bg-gray-100 text-gray-400 cursor-not-allowed'

: 'bg-gray-100 text-gray-700 hover:bg-gray-200 hover:scale-105'

}"

>

${wordObj.word}

</button>

`).join('')}

</div>

<button

onclick="nextPhase()"

${currentWords.length < minWords ? 'disabled' : ''}

class="w-full py-4 rounded-xl font-semibold flex items-center justify-center gap-2 transition-all ${

currentWords.length >= minWords

? 'bg-gradient-to-r from-blue-600 to-orange-500 text-white hover:shadow-lg'

: 'bg-gray-300 text-gray-500 cursor-not-allowed'

}"

>

${state.phase === 1 ? 'Próxima Pergunta' : 'Ver Resultados'}

<span>→</span>

</button>

</div>

</div>

</div>

`;

}

function renderEmail() {

return `

<div class="min-h-screen p-4 md:p-8 flex items-center justify-center">

<div class="max-w-2xl bg-white rounded-2xl shadow-2xl p-8 w-full">

<div class="text-center mb-8">

<div class="inline-flex items-center justify-center w-20 h-20 bg-gradient-to-r from-blue-600 to-orange-500 rounded-full mb-4">

<span class="text-5xl">📧</span>

</div>

<h1 class="text-3xl font-bold text-gray-800 mb-2">Quase lá!</h1>

<p class="text-gray-600">Informe seus dados para receber os resultados</p>

</div>

<div class="space-y-4 mb-6">

<div>

<label class="block text-sm font-medium text-gray-700 mb-2">Nome completo</label>

<input

type="text"

id="userName"

value="${state.userName}"

onchange="state.userName = this.value"

class="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-blue-500 focus:border-transparent"

placeholder="Digite seu nome completo"

required

/>

</div>

<div>

<label class="block text-sm font-medium text-gray-700 mb-2">Email</label>

<input

type="email"

id="userEmail"

value="${state.userEmail}"

onchange="state.userEmail = this.value"

class="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-blue-500 focus:border-transparent"

placeholder="seu@email.com"

required

/>

</div>

</div>

<button

onclick="validateAndSend()"

class="w-full bg-gradient-to-r from-blue-600 to-orange-500 text-white py-4 rounded-xl font-semibold hover:shadow-lg transition-all"

>

Enviar Resultados

</button>

</div>

</div>

`;

}

function renderResults() {

const sortedNatural = Object.entries(state.results.natural).sort(([,a], [,b]) => b - a);

const sortedAdaptado = Object.entries(state.results.adaptado).sort(([,a], [,b]) => b - a);

return `

<div class="min-h-screen p-4 md:p-8">

<div class="max-w-5xl mx-auto bg-white rounded-2xl shadow-2xl p-8">

<div class="text-center mb-8">

<div class="inline-flex items-center justify-center w-20 h-20 bg-gradient-to-r from-blue-600 to-orange-500 rounded-full mb-4">

<span class="text-5xl">🏆</span>

</div>

<h1 class="text-3xl font-bold text-gray-800 mb-2">Seu Mapeamento Cultural Brasil Infra</h1>

<p class="text-gray-600">Análise de aderência aos valores da empresa</p>

</div>

<div class="grid md:grid-cols-2 gap-6 mb-8">

<div class="bg-gradient-to-r from-blue-600 to-orange-500 rounded-xl p-6 text-white text-center">

<div class="text-5xl font-bold mb-2">${state.results.aderencia}%</div>

<div class="text-lg font-semibold mb-2">Aderência Cultural</div>

<div class="text-sm opacity-90">

${state.results.aderencia >= 85 ? "Excelente fit cultural!" :

state.results.aderencia >= 70 ? "Bom alinhamento cultural" :

state.results.aderencia >= 50 ? "Aderência moderada" : "Perfil cultural diferente"}

</div>

</div>

<div class="rounded-xl p-6 text-white text-center ${

state.results.energia >= 80 ? 'bg-green-500' :

state.results.energia >= 60 ? 'bg-yellow-500' : 'bg-red-500'

}">

<div class="text-5xl font-bold mb-2">${state.results.energia}%</div>

<div class="text-lg font-semibold mb-2">Nível de Energia</div>

<div class="text-sm opacity-90">

${state.results.energia >= 80 ? "Perfil natural alinhado" :

state.results.energia >= 60 ? "Adaptação moderada" : "Alto esforço de adaptação"}

</div>

</div>

</div>

<div class="grid md:grid-cols-2 gap-8 mb-8">

<div>

<h2 class="text-xl font-bold text-gray-800 mb-4 flex items-center gap-2">

<div class="w-3 h-3 bg-blue-500 rounded-full"></div>

Perfil Natural (Quem você é)

</h2>

${sortedNatural.map(([value, score]) => `

<div class="mb-3">

<div class="flex items-center justify-between mb-1">

<span class="text-sm font-medium text-gray-700">${valueNames[value]}</span>

<span class="text-sm font-bold text-gray-800">${score}%</span>

</div>

<div class="w-full bg-gray-200 rounded-full h-2">

<div class="${valueColors[value]} h-2 rounded-full" style="width: ${score}%"></div>

</div>

</div>

`).join('')}

</div>

<div>

<h2 class="text-xl font-bold text-gray-800 mb-4 flex items-center gap-2">

<div class="w-3 h-3 bg-orange-500 rounded-full"></div>

Perfil Adaptado (Como você age)

</h2>

${sortedAdaptado.map(([value, score]) => `

<div class="mb-3">

<div class="flex items-center justify-between mb-1">

<span class="text-sm font-medium text-gray-700">${valueNames[value]}</span>

<span class="text-sm font-bold text-gray-800">${score}%</span>

</div>

<div class="w-full bg-gray-200 rounded-full h-2">

<div class="${valueColors[value]} h-2 rounded-full" style="width: ${score}%"></div>

</div>

</div>

`).join('')}

</div>

</div>

<div class="bg-blue-50 rounded-xl p-6">

<h3 class="font-bold text-lg text-gray-800 mb-3">💡 Interpretação dos Resultados:</h3>

<div class="space-y-2 text-gray-700 text-sm">

<p><strong>Aderência Cultural:</strong> Mede o quanto seu perfil se alinha com os valores da Brasil Infra. Quanto maior, melhor o fit cultural.</p>

<p><strong>Nível de Energia:</strong> Indica o esforço necessário para você se adaptar ao ambiente. Quando natural e adaptado são muito diferentes, há maior desgaste.</p>

<p><strong>Perfil Natural:</strong> Quem você realmente é, suas características genuínas.</p>

<p><strong>Perfil Adaptado:</strong> Como você se comporta no ambiente de trabalho, suas adaptações.</p>

${state.results.energia < 60 ? `

<p class="mt-4 p-3 bg-yellow-100 rounded-lg">

⚠️ <strong>Atenção:</strong> Seu perfil natural e adaptado são muito diferentes. Isso pode gerar desgaste a longo prazo. Considere buscar funções onde você possa ser mais autêntico.

</p>

` : ''}

</div>

</div>

<div class="text-center mt-8">

<button

onclick="location.reload()"

class="bg-gradient-to-r from-blue-600 to-orange-500 text-white px-8 py-3 rounded-lg font-semibold hover:shadow-lg transition-all"

>

Refazer Teste

</button>

</div>

</div>

</div>

`;

}

function render() {

let html = '';

if (state.step === 'intro') {

html = renderIntro();

} else if (state.step === 'phase1' || state.step === 'phase2') {

html = renderPhase();

} else if (state.step === 'email') {

html = renderEmail();

} else if (state.step === 'results') {

html = renderResults();

}

document.getElementById('app').innerHTML = html;

}

function startTest() {

state.step = 'phase1';

render();

}

function validateAndSend() {

const name = document.getElementById('userName').value;

const email = document.getElementById('userEmail').value;

if (!name || !email) {

alert('Por favor, preencha todos os campos.');

return;

}

if (!email.includes('@')) {

alert('Por favor, insira um email válido.');

return;

}

state.userName = name;

state.userEmail = email;

sendResults();

}

*// Inicializar*

render();

</script>

</body>

</html>