# **👨‍🍳 30 RECEITAS DE MODIFICAÇÕES**

## **Copie, Cole e Personalize!**

# **🎨 RECEITAS DE CORES**

## **Receita 1: Tema Verde Empresarial**

/\* Substitua as variáveis no início do CSS \*/  
:root {  
 --primary-color: #10b981; /\* Verde principal \*/  
 --secondary-color: #059669; /\* Verde escuro \*/  
 --success-color: #22c55e; /\* Verde sucesso \*/  
 --danger-color: #ef4444; /\* Vermelho \*/  
 --warning-color: #f59e0b; /\* Amarelo \*/  
}

## **Receita 2: Tema Roxo Moderno**

:root {  
 --primary-color: #8b5cf6; /\* Roxo principal \*/  
 --secondary-color: #7c3aed; /\* Roxo escuro \*/  
 --success-color: #10b981; /\* Verde \*/  
 --danger-color: #ef4444; /\* Vermelho \*/  
 --warning-color: #f59e0b; /\* Amarelo \*/  
}

## **Receita 3: Tema Azul Escuro Profissional**

:root {  
 --primary-color: #1e40af; /\* Azul escuro \*/  
 --secondary-color: #1e3a8a; /\* Azul mais escuro \*/  
 --success-color: #059669; /\* Verde escuro \*/  
 --danger-color: #dc2626; /\* Vermelho escuro \*/  
 --warning-color: #d97706; /\* Laranja escuro \*/  
}

# **📝 RECEITAS DE NOVOS CAMPOS**

## **Receita 4: Adicionar Campo CPF**

**No HTML (dentro do form-cadastro):**

<div class="form-group">  
 <label for="cpf">CPF:</label>  
 <input type="text" id="cpf" placeholder="000.000.000-00" maxlength="14">  
</div>

**No JavaScript (função cadastrarPessoa):**

// Adicione após pegar outros valores  
const cpf = document.getElementById('cpf').value;  
  
// No objeto novaPessoa, adicione:  
const novaPessoa = {  
 // ... outros campos  
 cpf: cpf,  
 // ... resto  
};

**Formatação automática do CPF:**

document.getElementById('cpf').addEventListener('input', function(e) {  
 let value = e.target.value.replace(/\D/g, '');  
 if (value.length <= 11) {  
 value = value.replace(/(\d{3})(\d{3})(\d{3})(\d{2})/, '$1.$2.$3-$4');  
 }  
 e.target.value = value;  
});

## **Receita 5: Adicionar Campo Empresa**

**No HTML:**

<div class="form-group">  
 <label for="empresa">Empresa:</label>  
 <input type="text" id="empresa" placeholder="Nome da empresa">  
</div>

**No JavaScript:**

const empresa = document.getElementById('empresa').value;  
  
const novaPessoa = {  
 // ... outros campos  
 empresa: empresa || 'Não informado',  
 // ... resto  
};

## **Receita 6: Adicionar Campo Foto**

**No HTML:**

<div class="form-group">  
 <label for="foto">Foto:</label>  
 <input type="file" id="foto" accept="image/\*">  
 <img id="preview-foto" style="max-width: 200px; margin-top: 10px; display: none;">  
</div>

**Preview da foto:**

document.getElementById('foto').addEventListener('change', function(e) {  
 const file = e.target.files[0];  
 if (file) {  
 const reader = new FileReader();  
 reader.onload = function(e) {  
 document.getElementById('preview-foto').src = e.target.result;  
 document.getElementById('preview-foto').style.display = 'block';  
 };  
 reader.readAsDataURL(file);  
 }  
});

# **🏢 RECEITAS DE SALAS**

## **Receita 7: Adicionar Nova Sala**

**No HTML (lista de salas):**

<div class="room-card" onclick="selecionarSala('sala-roxa', this)">  
 <h3>🟣 Sala Roxa</h3>  
 <p><strong>Capacidade:</strong> 15 pessoas</p>  
 <p><strong>Recursos:</strong> Projetor 4K, Som surround, Ar condicionado</p>  
 <p><strong>Ideal para:</strong> Apresentações e palestras</p>  
</div>

**No select do agendamento:**

<option value="sala-roxa">🟣 Sala Roxa (15 pessoas)</option>

**No JavaScript (função renderizarVisualizacaoDia):**

const salas = {  
 // ... outras salas  
 'sala-roxa': { nome: '🟣 Sala Roxa', agendamentos: [] }  
};

## **Receita 8: Adicionar Auditório**

<div class="room-card" onclick="selecionarSala('auditorio', this)">  
 <h3>🎭 Auditório Principal</h3>  
 <p><strong>Capacidade:</strong> 100 pessoas</p>  
 <p><strong>Recursos:</strong> Palco, Sistema de som, Iluminação profissional</p>  
 <p><strong>Ideal para:</strong> Eventos e palestras grandes</p>  
</div>

# **⏰ RECEITAS DE HORÁRIOS**

## **Receita 9: Mudar para 24 horas**

// Encontre a função gerarHorarios e substitua:  
function gerarHorarios() {  
 const select = document.getElementById('horario-inicio');  
 select.innerHTML = '<option value="">Selecione</option>';  
   
 for (let hora = 0; hora < 24; hora++) { // 0 a 23  
 for (let minuto = 0; minuto < 60; minuto += 30) {  
 const h = hora.toString().padStart(2, '0');  
 const m = minuto.toString().padStart(2, '0');  
 const value = `${h}:${m}`;  
 const option = document.createElement('option');  
 option.value = value;  
 option.textContent = value;  
 select.appendChild(option);  
 }  
 }  
}

## **Receita 10: Intervalos de 15 minutos**

// Mude apenas esta linha:  
for (let minuto = 0; minuto < 60; minuto += 15) { // De 15 em 15

## **Receita 11: Adicionar Período (Manhã/Tarde/Noite)**

**No HTML:**

<div class="form-group">  
 <label for="periodo">Período:</label>  
 <select id="periodo">  
 <option value="manha">Manhã (8h-12h)</option>  
 <option value="tarde">Tarde (12h-18h)</option>  
 <option value="noite">Noite (18h-22h)</option>  
 </select>  
</div>

# **🔔 RECEITAS DE NOTIFICAÇÕES**

## **Receita 12: Alerta Bonito de Sucesso**

function showNotification(message, type = 'success') {  
 const notification = document.createElement('div');  
 notification.className = `notification ${type}`;  
 notification.innerHTML = `  
 <div style="  
 position: fixed;  
 top: 20px;  
 right: 20px;  
 background: ${type === 'success' ? '#10b981' : '#ef4444'};  
 color: white;  
 padding: 20px;  
 border-radius: 10px;  
 box-shadow: 0 5px 15px rgba(0,0,0,0.3);  
 z-index: 9999;  
 animation: slideIn 0.3s ease;  
 ">  
 ${type === 'success' ? '✅' : '❌'} ${message}  
 </div>  
 `;  
 document.body.appendChild(notification);  
   
 setTimeout(() => {  
 notification.remove();  
 }, 3000);  
}

## **Receita 13: Som de Notificação**

function playSound(type = 'success') {  
 const audio = new Audio();  
 audio.src = type === 'success'   
 ? 'data:audio/wav;base64,UklGRnoGAABXQVZFZm10IBAAAAABAAEAQB8AAEAfAAABAAgAZGF0YQoGAACBhYqFbF1fdJivrJBhNjVgodDbq2EcBj+a2/LDciUFLIHO8tiJNwgZaLvt559NEAxQp+PwtmMcBjiR1/LMeSwFJHfH8N2QQAoUXrTp66hVFApGn+DyvmwhBSuBzvLZiTYIG2m98OScTgwOUarm7blmFgU7k9n1unEiBC13yO/eizEIHWq+8+OWT'   
 : 'data:audio/wav;base64,UklGRjwGAABXQVZFZm10IBAAAAABAAEARKwAAIhYAQACABAAZGF0YRgGAACBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGBgYGB';  
 audio.play();  
}  
  
// Use assim:  
showNotification('Cadastro realizado!', 'success');  
playSound('success');

# **📊 RECEITAS DE VALIDAÇÕES**

## **Receita 14: Validar E-mail Melhorado**

function validarEmail(email) {  
 const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
 if (!regex.test(email)) {  
 showAlert('E-mail inválido! Use formato: [exemplo@email.com](mailto:exemplo@email.com)', 'error');  
 return false;  
 }  
 return true;  
}  
  
// Use antes de cadastrar:  
if (!validarEmail(email)) return;

## **Receita 15: Validar Telefone**

function validarTelefone(telefone) {  
 const numeros = telefone.replace(/\D/g, '');  
 if (numeros.length !== 11) {  
 showAlert('Telefone deve ter 11 dígitos!', 'error');  
 return false;  
 }  
 return true;  
}

## **Receita 16: Limite de Agendamentos por Dia**

function verificarLimiteAgendamentos(data) {  
 const agendamentosDia = agendamentos.filter(a => a.data === data);  
 const LIMITE = 20; // Máximo por dia  
   
 if (agendamentosDia.length >= LIMITE) {  
 showAlert(`Limite de ${LIMITE} agendamentos por dia atingido!`, 'error');  
 return false;  
 }  
 return true;  
}

# **🎯 RECEITAS DE RECURSOS EXTRAS**

## **Receita 17: Botão WhatsApp**

<button class="btn btn-success" onclick="enviarWhatsApp()">  
 📱 Enviar por WhatsApp  
</button>  
 function enviarWhatsApp() {  
 const pessoa = pessoas.find(p => p.id == document.getElementById('pessoa-agendamento').value);  
 const data = document.getElementById('data-agendamento').value;  
 const hora = document.getElementById('horario-inicio').value;  
   
 const mensagem = `Olá ${pessoa.nome}! Confirmando seu agendamento para ${data} às ${hora}.`;  
 const numeroEmpresa = '5511999999999'; // Coloque o número da empresa  
   
 window.open(`https://wa.me/${numeroEmpresa}?text=${encodeURIComponent(mensagem)}`);  
}

## **Receita 18: Imprimir Crachá**

function imprimirCracha(pessoaId) {  
 const pessoa = pessoas.find(p => p.id === pessoaId);  
   
 const janela = window.open('', '', 'width=400,height=300');  
 janela.document.write(`  
 <html>  
 <head>  
 <title>Crachá - ${pessoa.nome}</title>  
 <style>  
 body {  
 font-family: Arial;  
 text-align: center;  
 padding: 50px;  
 }  
 .cracha {  
 border: 3px solid #2563eb;  
 padding: 30px;  
 border-radius: 15px;  
 }  
 h1 { color: #2563eb; }  
 </style>  
 </head>  
 <body>  
 <div class="cracha">  
 <h1>VISITANTE</h1>  
 <h2>${pessoa.nome}</h2>  
 <p>${pessoa.empresa || 'Visitante'}</p>  
 <p>${new Date().toLocaleDateString('pt-BR')}</p>  
 <img src="https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=${pessoa.id}">  
 </div>  
 </body>  
 </html>  
 `);  
   
 setTimeout(() => {  
 janela.print();  
 janela.close();  
 }, 500);  
}

## **Receita 19: Contador Regressivo**

function mostrarProximoAgendamento() {  
 const agora = new Date();  
 const proximoAgendamento = agendamentos  
 .filter(a => new Date(a.data + ' ' + a.horario) > agora)  
 .sort((a, b) => new Date(a.data + ' ' + a.horario) - new Date(b.data + ' ' + b.horario))[0];  
   
 if (proximoAgendamento) {  
 const tempo = new Date(proximoAgendamento.data + ' ' + proximoAgendamento.horario) - agora;  
 const horas = Math.floor(tempo / (1000 \* 60 \* 60));  
 const minutos = Math.floor((tempo % (1000 \* 60 \* 60)) / (1000 \* 60));  
   
 document.getElementById('contador').innerHTML = `  
 ⏰ Próximo agendamento em: ${horas}h ${minutos}min  
 `;  
 }  
}  
  
// Atualizar a cada minuto  
setInterval(mostrarProximoAgendamento, 60000);

# **📈 RECEITAS DE RELATÓRIOS**

## **Receita 20: Relatório Mensal**

function gerarRelatorioMensal(mes, ano) {  
 const agendamentosMes = agendamentos.filter(a => {  
 const data = new Date(a.data);  
 return data.getMonth() === mes && data.getFullYear() === ano;  
 });  
   
 const relatorio = {  
 total: agendamentosMes.length,  
 porSala: {},  
 porTipo: { visitante: 0, colaborador: 0 }  
 };  
   
 // Contar por sala  
 agendamentosMes.forEach(a => {  
 relatorio.porSala[a.sala] = (relatorio.porSala[a.sala] || 0) + 1;  
   
 const pessoa = pessoas.find(p => p.id === a.pessoaId);  
 if (pessoa) {  
 relatorio.porTipo[pessoa.tipo]++;  
 }  
 });  
   
 console.log('Relatório:', relatorio);  
 return relatorio;  
}

## **Receita 21: Gráfico de Pizza**

<!-- Adicione no HTML -->  
<canvas id="graficoPizza" width="400" height="400"></canvas>  
 function desenharGraficoPizza() {  
 const canvas = document.getElementById('graficoPizza');  
 const ctx = canvas.getContext('2d');  
 const centerX = canvas.width / 2;  
 const centerY = canvas.height / 2;  
 const radius = 150;  
   
 // Dados  
 const visitantes = pessoas.filter(p => p.tipo === 'visitante').length;  
 const colaboradores = pessoas.filter(p => p.tipo === 'colaborador').length;  
 const total = visitantes + colaboradores;  
   
 // Visitantes (azul)  
 ctx.beginPath();  
 ctx.arc(centerX, centerY, radius, 0, (visitantes / total) \* 2 \* Math.PI);  
 ctx.lineTo(centerX, centerY);  
 ctx.fillStyle = '#3b82f6';  
 ctx.fill();  
   
 // Colaboradores (verde)  
 ctx.beginPath();  
 ctx.arc(centerX, centerY, radius, (visitantes / total) \* 2 \* Math.PI, 2 \* Math.PI);  
 ctx.lineTo(centerX, centerY);  
 ctx.fillStyle = '#22c55e';  
 ctx.fill();  
}

# **🔐 RECEITAS DE SEGURANÇA**

## **Receita 22: Senha para Excluir**

function excluirComSenha(id) {  
 const senha = prompt('Digite a senha de administrador:');  
   
 if (senha !== '1234') { // Mude a senha!  
 showAlert('Senha incorreta!', 'error');  
 return;  
 }  
   
 // Código original de exclusão  
 excluirPessoa(id);  
}

## **Receita 23: Log de Atividades**

function registrarAtividade(acao, detalhes) {  
 const log = {  
 acao: acao,  
 detalhes: detalhes,  
 dataHora: new Date().toISOString(),  
 usuario: 'Recepção' // Poderia pegar de login  
 };  
   
 let logs = JSON.parse(localStorage.getItem('logs') || '[]');  
 logs.push(log);  
   
 // Manter apenas últimos 1000 logs  
 if (logs.length > 1000) {  
 logs = logs.slice(-1000);  
 }  
   
 localStorage.setItem('logs', JSON.stringify(logs));  
}  
  
// Use assim:  
registrarAtividade('cadastro', `Nova pessoa: ${nome}`);  
registrarAtividade('agendamento', `Sala ${sala} - ${data}`);

# **🎨 RECEITAS DE TEMAS**

## **Receita 24: Tema Alto Contraste**

.alto-contraste {  
 --primary-color: #000000;  
 --secondary-color: #333333;  
 --success-color: #00ff00;  
 --danger-color: #ff0000;  
 --dark-bg: #000000;  
 --dark-text: #ffff00;  
 --light-bg: #ffffff;  
 --light-text: #000000;  
}  
  
body.alto-contraste \* {  
 font-weight: bold !important;  
}

## **Receita 25: Tema Infantil**

.tema-infantil {  
 --primary-color: #ff6b6b;  
 --secondary-color: #4ecdc4;  
 --success-color: #95e1d3;  
 --danger-color: #f38181;  
 font-family: 'Comic Sans MS', cursive !important;  
}  
  
.tema-infantil .btn {  
 border-radius: 30px;  
 font-size: 20px;  
}

# **📱 RECEITAS MOBILE**

## **Receita 26: Menu Hambúrguer**

<button class="menu-mobile" onclick="toggleMenu()">☰</button>  
 .menu-mobile {  
 display: none;  
 font-size: 30px;  
 background: none;  
 border: none;  
 cursor: pointer;  
}  
  
@media (max-width: 768px) {  
 .menu-mobile { display: block; }  
 .tabs {   
 display: none;  
 position: absolute;  
 top: 60px;  
 left: 0;  
 right: 0;  
 flex-direction: column;  
 background: white;  
 box-shadow: 0 5px 15px rgba(0,0,0,0.3);  
 }  
 .tabs.show { display: flex; }  
}  
 function toggleMenu() {  
 document.querySelector('.tabs').classList.toggle('show');  
}

# **🚀 RECEITAS AVANÇADAS**

## **Receita 27: Busca Inteligente**

function buscaInteligente(termo) {  
 termo = termo.toLowerCase();  
   
 return pessoas.filter(pessoa => {  
 // Busca em todos os campos  
 return Object.values(pessoa).some(valor =>   
 String(valor).toLowerCase().includes(termo)  
 );  
 });  
}

## **Receita 28: Exportar para PDF**

function exportarPDF() {  
 // Requer biblioteca jsPDF  
 const doc = new jsPDF();  
   
 doc.text('Relatório de Agendamentos', 20, 20);  
   
 let y = 40;  
 agendamentos.forEach(a => {  
 doc.text(`${a.data} - ${a.pessoaNome} - ${a.sala}`, 20, y);  
 y += 10;  
   
 if (y > 270) {  
 doc.addPage();  
 y = 20;  
 }  
 });  
   
 doc.save('agendamentos.pdf');  
}

## **Receita 29: Modo Offline**

// Detectar conexão  
window.addEventListener('online', () => {  
 showNotification('Conexão restaurada! Sincronizando...', 'success');  
 salvarNoGoogleSheets();  
});  
  
window.addEventListener('offline', () => {  
 showNotification('Sem conexão. Dados salvos localmente.', 'warning');  
});

## **Receita 30: Atalhos de Teclado**

document.addEventListener('keydown', function(e) {  
 // Ctrl + N = Novo cadastro  
 if (e.ctrlKey && e.key === 'n') {  
 e.preventDefault();  
 showTab('cadastro');  
 document.querySelector('.tab[onclick\*="cadastro"]').click();  
 }  
   
 // Ctrl + A = Agendar  
 if (e.ctrlKey && e.key === 'a') {  
 e.preventDefault();  
 showTab('agendamento');  
 document.querySelector('.tab[onclick\*="agendamento"]').click();  
 }  
   
 // Esc = Voltar ao dashboard  
 if (e.key === 'Escape') {  
 showTab('dashboard');  
 document.querySelector('.tab[onclick\*="dashboard"]').click();  
 }  
});

# **📝 COMO USAR ESTAS RECEITAS**

1. **Escolha a receita** que precisa
2. **Copie o código** correspondente
3. **Cole no local indicado** (HTML, CSS ou JavaScript)
4. **Personalize** valores, textos e cores
5. **Teste** a funcionalidade
6. **Ajuste** conforme necessário

## **Dicas:**

* 💡 Sempre faça backup antes
* 💡 Teste uma receita por vez
* 💡 Leia os comentários no código
* 💡 Adapte para suas necessidades

# **🎯 BÔNUS: COMBINAÇÕES PODEROSAS**

## **Combo 1: Sistema Completo de Visitantes**

* Receita 4 (CPF) +
* Receita 5 (Empresa) +
* Receita 18 (Crachá) +
* Receita 17 (WhatsApp)

## **Combo 2: Dashboard Gerencial**

* Receita 20 (Relatório) +
* Receita 21 (Gráfico) +
* Receita 19 (Contador) +
* Receita 28 (PDF)

## **Combo 3: Segurança Total**

* Receita 22 (Senha) +
* Receita 23 (Log) +
* Receita 29 (Offline) +
* Receita 16 (Limites)

*Use estas receitas para transformar o sistema!*