**CODIGOS USADOS NO PROJETO EXPOTECH   
24/05/25  
  
SAFE GUARD**

**SUPABASE SQL:**

**/\***

**CREATE TABLE Setores (**

**id SERIAL PRIMARY KEY,**

**nome VARCHAR(100) NOT NULL**

**);**

**CREATE TABLE Funcionarios (**

**id SERIAL PRIMARY KEY,**

**nome VARCHAR(100) NOT NULL,**

**cargo VARCHAR(100),**

**data\_admissao DATE,**

**setor\_id INTEGER REFERENCES Setores(id)**

**);**

**CREATE TABLE EPIs (**

**id SERIAL PRIMARY KEY,**

**nome VARCHAR(100) NOT NULL,**

**descricao TEXT,**

**tempo\_troca\_meses INTEGER NOT NULL**

**);**

**CREATE TABLE Retiradas (**

**id SERIAL PRIMARY KEY,**

**funcionario\_id INTEGER REFERENCES Funcionarios(id),**

**epi\_id INTEGER REFERENCES EPIs(id),**

**data\_retirada DATE NOT NULL,**

**data\_proxima\_troca DATE NOT NULL**

**);**

**INSERT INTO Setores (nome) VALUES**

**('Manutenção'),**

**('Produção'),**

**('Qualidade'),**

**('Segurança'),**

**('Administração');**

**INSERT INTO Funcionarios (nome, cargo, data\_admissao, setor\_id) VALUES**

**('Maria Oliveira', 'Técnico de Manutenção', '2022-05-20', (SELECT id FROM Setores WHERE nome = 'Manutenção')),**

**('Carlos Souza', 'Supervisor', '2021-03-10', (SELECT id FROM Setores WHERE nome = 'Produção')),**

**('Ana Silva', 'Operador', '2023-07-05', (SELECT id FROM Setores WHERE nome = 'Produção')),**

**('Pedro Almeida', 'Engenheiro de Segurança', '2020-11-30', (SELECT id FROM Setores WHERE nome = 'Segurança')),**

**('João Santos', 'Analista de Qualidade', '2022-09-15', (SELECT id FROM Setores WHERE nome = 'Qualidade')),**

**('Luiza Costa', 'Técnico de Segurança', '2021-06-25', (SELECT id FROM Setores WHERE nome = 'Segurança')),**

**('Fernanda Lima', 'Operador', '2023-02-10', (SELECT id FROM Setores WHERE nome = 'Produção')),**

**('Ricardo Pereira', 'Engenheiro de Produção', '2020-12-05', (SELECT id FROM Setores WHERE nome = 'Produção')),**

**('Camila Rocha', 'Supervisor', '2022-04-30', (SELECT id FROM Setores WHERE nome = 'Produção')),**

**('Gustavo Fernandes', 'Técnico de Manutenção', '2021-08-18', (SELECT id FROM Setores WHERE nome = 'Manutenção')),**

**('Beatriz Martins', 'Analista de Qualidade', '2023-01-22', (SELECT id FROM Setores WHERE nome = 'Qualidade'));**

**INSERT INTO EPIs (nome, descricao, tempo\_troca\_meses) VALUES**

**('Luva de Proteção', 'Luva para proteção das mãos', 6),**

**('Capacete', 'Capacete de segurança para proteção da cabeça', 12),**

**('Protetor Auricular', 'Protetor para redução de ruído', 8),**

**('Máscara Respiradora', 'Máscara para proteção respiratória', 4),**

**('Óculos de Proteção', 'Óculos para proteção ocular', 10),**

**('Calçado de Segurança', 'Calçado para proteção dos pés', 18),**

**('Cinto de Segurança', 'Cinto para proteção contra quedas', 24),**

**('Avental de Proteção', 'Avental para proteção corporal', 12),**

**('Luvas de Procedimento', 'Luvas descartáveis para procedimentos', 1),**

**('Roupa de Alta Visibilidade', 'Roupa para aumentar a visibilidade', 12),**

**('Protetor Facial', 'Protetor para proteção do rosto', 6);**

**INSERT INTO Retiradas (funcionario\_id, epi\_id, data\_retirada, data\_proxima\_troca) VALUES**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Maria Oliveira'),**

**(SELECT id FROM EPIs WHERE nome = 'Luva de Proteção'),**

**'2024-01-15',**

**'2024-07-15'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Carlos Souza'),**

**(SELECT id FROM EPIs WHERE nome = 'Capacete'),**

**'2024-02-01',**

**'2025-02-01'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Ana Silva'),**

**(SELECT id FROM EPIs WHERE nome = 'Protetor Auricular'),**

**'2024-03-10',**

**'2024-11-10'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Pedro Almeida'),**

**(SELECT id FROM EPIs WHERE nome = 'Máscara Respiradora'),**

**'2024-04-05',**

**'2024-08-05'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'João Santos'),**

**(SELECT id FROM EPIs WHERE nome = 'Óculos de Proteção'),**

**'2024-05-20',**

**'2025-03-20'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Luiza Costa'),**

**(SELECT id FROM EPIs WHERE nome = 'Calçado de Segurança'),**

**'2024-06-01',**

**'2025-12-01'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Fernanda Lima'),**

**(SELECT id FROM EPIs WHERE nome = 'Cinto de Segurança'),**

**'2024-07-15',**

**'2026-07-15'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Ricardo Pereira'),**

**(SELECT id FROM EPIs WHERE nome = 'Avental de Proteção'),**

**'2024-08-20',**

**'2025-08-20'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Camila Rocha'),**

**(SELECT id FROM EPIs WHERE nome = 'Luvas de Procedimento'),**

**'2024-09-25',**

**'2024-10-25'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Gustavo Fernandes'),**

**(SELECT id FROM EPIs WHERE nome = 'Roupa de Alta Visibilidade'),**

**'2024-10-10',**

**'2025-10-10'**

**),**

**(**

**(SELECT id FROM Funcionarios WHERE nome = 'Beatriz Martins'),**

**(SELECT id FROM EPIs WHERE nome = 'Protetor Facial'),**

**'2024-11-05',**

**'2025-05-05'**

**);**

**CREATE POLICY "Inserir funcionarios"**

**ON "funcionarios"**

**FOR INSERT**

**WITH CHECK (auth.role() = 'anon');**

**CREATE POLICY "Inserir setores"**

**ON "setores"**

**FOR INSERT**

**WITH CHECK (auth.role() = 'anon');**

**ALTER TABLE setores**

**ADD CONSTRAINT setores\_pkey PRIMARY KEY (id);**

**ALTER TABLE funcionarios**

**ADD COLUMN setor\_id INTEGER REFERENCES setores(id);**

**UPDATE funcionarios SET setor\_id = (SELECT id FROM setores WHERE nome = 'Manutenção') WHERE nome = 'Maria Oliveira';**

**UPDATE funcionarios SET setor\_id = (SELECT id FROM setores WHERE nome = 'Produção') WHERE nome IN ('Carlos Souza', 'Ana Silva', 'Fernanda Lima', 'Ricardo Pereira', 'Camila Rocha');**

**UPDATE funcionarios SET setor\_id = (SELECT id FROM setores WHERE nome = 'Segurança') WHERE nome IN ('Pedro Almeida', 'Luiza Costa');**

**UPDATE funcionarios SET setor\_id = (SELECT id FROM setores WHERE nome = 'Qualidade') WHERE nome IN ('João Santos', 'Beatriz Martins');**

**SELECT f.nome AS funcionario, f.cargo, s.nome AS setor**

**FROM funcionarios f**

**JOIN setores s ON f.setor\_id = s.id;**

**-- Atualizar capacetes para 16 meses**

**UPDATE epis SET tempo\_troca\_meses = 16 WHERE nome ILIKE '%capacete%';**

**-- Atualizar luvas (geralmente 3 meses)**

**UPDATE epis SET tempo\_troca\_meses = 3 WHERE nome ILIKE '%luva%';**

**-- Atualizar óculos de proteção (geralmente 12 meses)**

**UPDATE epis SET tempo\_troca\_meses = 12 WHERE nome ILIKE '%óculos%' OR nome ILIKE '%oculos%';**

**-- Atualizar protetores auriculares (geralmente 6 meses)**

**UPDATE epis SET tempo\_troca\_meses = 6 WHERE nome ILIKE '%protetor%auricular%';**

**-- Atualizar máscaras (geralmente 3 meses)**

**UPDATE epis SET tempo\_troca\_meses = 3 WHERE nome ILIKE '%máscara%' OR nome ILIKE '%mascara%';**

**-- Criar tabela de motivos**

**CREATE TABLE motivos (**

**id SERIAL PRIMARY KEY,**

**nome VARCHAR(100) NOT NULL**

**);**

**-- Inserir os motivos**

**INSERT INTO motivos (nome) VALUES**

**('Prevenção'),**

**('Substituição'),**

**('Distribuição'),**

**('Visitante'),**

**('Validade'),**

**('Outro');**

**-- Primeiro, vamos verificar se a tabela retiradas tem todos os campos necessários**

**ALTER TABLE retiradas**

**ADD COLUMN IF NOT EXISTS motivo VARCHAR(100);**

**-- Certificar que as colunas têm os tipos corretos**

**ALTER TABLE retiradas**

**ALTER COLUMN funcionario\_id TYPE INTEGER,**

**ALTER COLUMN epi\_id TYPE INTEGER,**

**ALTER COLUMN data\_retirada TYPE DATE,**

**ALTER COLUMN data\_proxima\_troca TYPE DATE;**

**-- Garantir que as chaves estrangeiras estão corretas**

**ALTER TABLE retiradas**

**ADD CONSTRAINT fk\_funcionario**

**FOREIGN KEY (funcionario\_id)**

**REFERENCES funcionarios(id);**

**ALTER TABLE retiradas**

**ADD CONSTRAINT fk\_epi**

**FOREIGN KEY (epi\_id)**

**REFERENCES epis(id);\*/**

**CODIGOS USADOS NO SITE REPLIT  
  
HTML DA PAGINA PRINCIPAL:**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>Safe Guard - Controle de EPIs</title>

<link rel="stylesheet" href="style.css">

<script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js"></script>

</head>

<body>

<button class="theme-toggle" onclick="toggleTheme()">🌞 Modo Claro</button>

<header>

<div class="logo-container">

<img src="attached\_assets/Escudo e mão do \_Safe Guard\_.png" alt="Safe Guard Logo" class="logo" loading="eager" decoding="async">

<a href="index.html" class="nav-link">Voltar para Início</a>

</div>

</header>

<div id="dashboard">

<button id="btnCarregarFuncionarios">Carregar Funcionários</button>

<button id="btnCarregarRetiradas">Carregar Retiradas de EPIs</button>

<h2>Registrar Novo Funcionário</h2>

<form id="formFuncionario">

<input type="text" id="nome" placeholder="Nome" required>

<input type="text" id="cargo" placeholder="Cargo" required>

<select id="setor" required>

<option value="">Selecione o Setor</option>

</select>

<input type="date" id="dataAdmissao" required>

<button type="submit">Adicionar Funcionário</button>

</form>

<h2>Funcionários</h2>

<input type="text" id="pesquisaFuncionarios" placeholder="Pesquisar funcionários..." style="margin-bottom: 1rem;">

<table id="tabelaFuncionarios">

<thead>

<tr>

<th>Nome</th>

<th>Cargo</th>

<th>Setor</th>

<th>Data de Admissão</th>

<th>Ações</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<!-- Modal de Edição -->

<div id="modalEdicao" class="modal">

<div class="modal-content">

<h3>Editar Funcionário</h3>

<form id="formEdicao">

<input type="hidden" id="editId">

<input type="text" id="editNome" required>

<input type="text" id="editCargo" required>

<select id="editSetor" required>

<option value="">Selecione o Setor</option>

</select>

<input type="date" id="editDataAdmissao" required>

<input type="password" id="editSenha" placeholder="Senha" required>

<div class="modal-buttons">

<button type="submit">Salvar</button>

<button type="button" onclick="fecharModal()">Cancelar</button>

</div>

</form>

</div>

</div>

<h2>Registrar Retirada de EPI</h2>

<form id="formRetirada">

<select id="funcionarioRetirada" required>

<option value="">Selecione o Funcionário</option>

</select>

<select id="epiRetirada" required>

<option value="">Selecione o EPI</option>

</select>

<select id="motivoRetirada" required>

<option value="">Selecione o Motivo</option>

</select>

<input type="date" id="dataRetirada" required>

<input type="date" id="dataProximaTroca" required readonly>

<button type="submit">Adicionar Retirada</button>

</form>

<h2>Retiradas de EPIs</h2>

<input type="text" id="pesquisaRetiradas" placeholder="Pesquisar retiradas..." style="margin-bottom: 1rem;">

<table id="tabelaRetiradas">

<thead>

<tr>

<th>Funcionário</th>

<th>EPI</th>

<th>Motivo</th>

<th>Data de Retirada</th>

<th>Próxima Troca</th>

<th>Ações</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<!-- Modal de Edição de Retirada -->

<div id="modalEdicaoRetirada" class="modal">

<div class="modal-content">

<h3>Editar Retirada de EPI</h3>

<form id="formEdicaoRetirada">

<input type="hidden" id="editRetiradaId">

<select id="editFuncionarioRetirada" required>

<option value="">Selecione o Funcionário</option>

</select>

<select id="editEpiRetirada" required>

<option value="">Selecione o EPI</option>

</select>

<select id="editMotivoRetirada" required>

<option value="">Selecione o Motivo</option>

</select>

<input type="date" id="editDataRetirada" required>

<input type="date" id="editDataProximaTroca" required>

<input type="password" id="editSenhaRetirada" placeholder="Senha" required>

<div class="modal-buttons">

<button type="submit">Salvar</button>

<button type="button" onclick="fecharModalRetirada()">Cancelar</button>

</div>

</form>

</div>

</div>

<h2>Dashboard</h2>

<div class="dashboard-container">

<div class="dashboard-card">

<h3>EPIs por Setor</h3>

<canvas id="episPorSetor"></canvas>

</div>

<div class="dashboard-card">

<h3>Retiradas por Mês</h3>

<canvas id="retiradasPorMes"></canvas>

</div>

<div class="dashboard-card">

<h3>Top 5 EPIs mais Retirados</h3>

<canvas id="topEpis"></canvas>

</div>

<div class="dashboard-card">

<h3>Status das Trocas</h3>

<canvas id="statusTrocas"></canvas>

</div>

<div class="stats-container">

<div class="stat-card">

<h4>Total de EPIs</h4>

<p id="totalEpis">0</p>

</div>

<div class="stat-card">

<h4>Funcionários Ativos</h4>

<p id="totalFuncionarios">0</p>

</div>

<div class="stat-card">

<h4>Trocas Pendentes</h4>

<p id="trocasPendentes">0</p>

</div>

</div>

</div>

</div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script src="script.js"></script>

</body>

</html>

**HTML DO SITE:**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>Safe Guard - Introdução</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<div class="logo-container">

<img src="attached\_assets/Escudo e mão do \_Safe Guard\_.png" alt="Safe Guard Logo" class="logo">

</div>

</header>

<main>

<section class="hero">

<h1>SAFE GUARD</h1>

<p class="subtitle">Sistema de Gestão de Equipamentos de Proteção Individual</p>

</section>

<section class="about">

<div class="container">

<h2>Sobre o Projeto</h2>

<p>O controle de EPIs é fundamental para garantir a segurança e o bem-estar dos trabalhadores em ambientes de risco. Os EPIs são equipamentos essenciais para proteger os colaboradores de acidentes e doenças ocupacionais, como capacetes, luvas, botas, protetores auditivos, entre outros.</p>

<p>Manter um controle rigoroso sobre a retirada e devolução desses itens ajuda a evitar desperdícios, garantir a conformidade com as normas de segurança e assegurar que todos os funcionários tenham os recursos necessários para desempenharem suas funções com segurança.</p>

<div class="cta-container">

<a href="dashboard.html" class="cta-button">Acessar Sistema de EPIs</a>

</div>

</div>

</section>

</main>

</body>

</html>

**JAVASCRIPT:**

// Carregar setores

async function carregarSetores() {

const { data, error } = await supabase

.from('setores')

.select('\*');

if (error) {

console.error('Erro ao carregar setores:', error);

return;

}

const seletorSetor = document.getElementById('setor');

const editSeletorSetor = document.getElementById('editSetor');

// Limpar opções existentes

seletorSetor.innerHTML = '<option value="">Selecione o Setor</option>';

editSeletorSetor.innerHTML = '<option value="">Selecione o Setor</option>';

data.forEach(setor => {

// Adicionar ao seletor principal

const option = document.createElement('option');

option.value = setor.id;

option.textContent = setor.nome;

seletorSetor.appendChild(option);

// Adicionar ao seletor do modal de edição

const editOption = option.cloneNode(true);

editSeletorSetor.appendChild(editOption);

});

}

// Conexão com o Supabase

const supabase = window.supabase.createClient(

'https://yngpiwqarastgwdcqbyp.supabase.co',

'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InluZ3Bpd3FhcmFzdGd3ZGNxYnlwIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDE5MDUxOTksImV4cCI6MjA1NzQ4MTE5OX0.7ZOYVhA1IsS4OlcETp5rETWuey0EFPlJ8tCy9dlDHwo'

);

// Botão para carregar funcionários

document.getElementById('btnCarregarFuncionarios').addEventListener('click', async () => {

const { data, error } = await supabase

.from('funcionarios')

.select(`

\*,

setores (nome)

`);

if (error) {

console.error('Erro ao carregar funcionários:', error);

return;

}

const tabela = document.getElementById('tabelaFuncionarios').getElementsByTagName('tbody')[0];

tabela.innerHTML = '';

data.forEach(funcionario => {

const tr = document.createElement('tr');

tr.innerHTML = `

<td>${funcionario.nome}</td>

<td>${funcionario.cargo}</td>

<td>${funcionario.setores?.nome || 'N/A'}</td>

<td>${funcionario.data\_admissao}</td>

<td>

<div class="action-buttons">

<button class="btn-edit" onclick="editarFuncionario(${funcionario.id})">Editar</button>

<button class="btn-delete" onclick="deletarFuncionario(${funcionario.id})">Excluir</button>

</div>

</td>

`;

tabela.appendChild(tr);

});

});

// Botão para carregar retiradas de EPIs

document.getElementById('btnCarregarRetiradas').addEventListener('click', async () => {

try {

const { data, error } = await supabase

.from('retiradas')

.select(`

id,

funcionario\_id,

epi\_id,

motivo,

data\_retirada,

data\_proxima\_troca,

funcionarios!retiradas\_funcionario\_id\_fkey (nome),

epis!retiradas\_epi\_id\_fkey (nome)

`);

if (error) {

console.error('Erro ao carregar retiradas:', error);

return;

}

const tabela = document.getElementById('tabelaRetiradas').getElementsByTagName('tbody')[0];

tabela.innerHTML = '';

data.forEach(retirada => {

const tr = document.createElement('tr');

tr.innerHTML = `

<td>${retirada.funcionarios?.nome || 'N/A'}</td>

<td>${retirada.epis?.nome || 'N/A'}</td>

<td>${retirada.motivo || 'N/A'}</td>

<td>${retirada.data\_retirada || 'N/A'}</td>

<td>${retirada.data\_proxima\_troca || 'N/A'}</td>

<td>

<div class="action-buttons">

<button class="btn-edit" onclick="editarRetirada(${retirada.id})">Editar</button>

<button class="btn-delete" onclick="deletarRetirada(${retirada.id})">Excluir</button>

</div>

</td>

`;

tabela.appendChild(tr);

});

} catch (err) {

console.error('Erro na execução:', err);

alert('Erro ao processar retiradas: ' + err.message);

}

});

// Carregar setores quando a página carregar

// Funções do Modal

function abrirModal() {

document.getElementById('modalEdicao').style.display = 'block';

}

function fecharModal() {

document.getElementById('modalEdicao').style.display = 'none';

}

async function editarFuncionario(id) {

const { data: funcionario, error } = await supabase

.from('funcionarios')

.select('\*')

.eq('id', id)

.single();

if (error) {

console.error('Erro ao carregar funcionário:', error);

return;

}

document.getElementById('editId').value = funcionario.id;

document.getElementById('editNome').value = funcionario.nome;

document.getElementById('editCargo').value = funcionario.cargo;

document.getElementById('editSetor').value = funcionario.setor\_id || '';

document.getElementById('editDataAdmissao').value = funcionario.data\_admissao;

abrirModal();

}

async function deletarFuncionario(id) {

const senha = prompt('Digite a senha para excluir o funcionário:');

if (senha !== '123456') {

alert('Senha incorreta!');

return;

}

if (confirm('Tem certeza que deseja excluir este funcionário?')) {

const { error } = await supabase

.from('funcionarios')

.delete()

.eq('id', id);

if (error) {

console.error('Erro ao deletar funcionário:', error);

alert('Erro ao deletar funcionário!');

} else {

alert('Funcionário deletado com sucesso!');

document.getElementById('btnCarregarFuncionarios').click();

}

}

}

// Form de Edição

document.getElementById('formEdicao').addEventListener('submit', async (e) => {

e.preventDefault();

const senha = document.getElementById('editSenha').value;

if (senha !== '123456') {

alert('Senha incorreta!');

return;

}

const id = document.getElementById('editId').value;

const nome = document.getElementById('editNome').value;

const cargo = document.getElementById('editCargo').value;

const setor\_id = document.getElementById('editSetor').value;

const dataAdmissao = document.getElementById('editDataAdmissao').value;

const { error } = await supabase

.from('funcionarios')

.update({

nome,

cargo,

setor\_id,

data\_admissao: dataAdmissao

})

.eq('id', id);

if (error) {

console.error('Erro ao atualizar funcionário:', error);

alert('Erro ao atualizar funcionário!');

} else {

alert('Funcionário atualizado com sucesso!');

fecharModal();

document.getElementById('btnCarregarFuncionarios').click();

}

});

// Funções para o Modal de Retirada

function abrirModalRetirada() {

document.getElementById('modalEdicaoRetirada').style.display = 'block';

}

function fecharModalRetirada() {

document.getElementById('modalEdicaoRetirada').style.display = 'none';

}

async function carregarFuncionariosSelect() {

const { data, error } = await supabase

.from('funcionarios')

.select('id, nome, setor\_id');

if (error) {

console.error('Erro ao carregar funcionários:', error);

return;

}

const selects = ['funcionarioRetirada', 'editFuncionarioRetirada'];

selects.forEach(selectId => {

const select = document.getElementById(selectId);

select.innerHTML = '<option value="">Selecione o Funcionário</option>';

if (data) {

data.forEach(funcionario => {

const option = document.createElement('option');

option.value = funcionario.id;

option.textContent = funcionario.nome;

select.appendChild(option);

});

}

});

}

// Formulário para adicionar nova retirada

document.getElementById('formRetirada').addEventListener('submit', async (e) => {

e.preventDefault();

const funcionario\_id = parseInt(document.getElementById('funcionarioRetirada').value);

const epi\_id = parseInt(document.getElementById('epiRetirada').value);

const motivoSelect = document.getElementById('motivoRetirada');

const motivo = motivoSelect.options[motivoSelect.selectedIndex].text;

const data\_retirada = document.getElementById('dataRetirada').value;

const data\_proxima\_troca = document.getElementById('dataProximaTroca').value;

const { error } = await supabase

.from('retiradas')

.insert([{

funcionario\_id,

epi\_id,

motivo,

data\_retirada,

data\_proxima\_troca

}]);

if (error) {

console.error('Erro ao adicionar retirada:', error);

alert('Erro ao adicionar retirada!');

} else {

alert('Retirada adicionada com sucesso!');

document.getElementById('formRetirada').reset();

document.getElementById('btnCarregarRetiradas').click();

}

});

async function editarRetirada(id) {

const { data: retirada, error } = await supabase

.from('retiradas')

.select('\*')

.eq('id', id)

.single();

if (error) {

console.error('Erro ao carregar retirada:', error);

return;

}

document.getElementById('editRetiradaId').value = retirada.id;

document.getElementById('editFuncionarioRetirada').value = retirada.funcionario\_id;

document.getElementById('editEpiRetirada').value = retirada.epi\_id;

document.getElementById('editDataRetirada').value = retirada.data\_retirada;

document.getElementById('editDataProximaTroca').value = retirada.data\_proxima\_troca;

abrirModalRetirada();

}

async function deletarRetirada(id) {

const senha = prompt('Digite a senha para excluir a retirada:');

if (senha !== '123456') {

alert('Senha incorreta!');

return;

}

if (confirm('Tem certeza que deseja excluir esta retirada?')) {

const { error } = await supabase

.from('retiradas')

.delete()

.eq('id', id);

if (error) {

console.error('Erro ao deletar retirada:', error);

alert('Erro ao deletar retirada!');

} else {

alert('Retirada deletada com sucesso!');

document.getElementById('btnCarregarRetiradas').click();

}

}

}

// Form de Edição de Retirada

document.getElementById('formEdicaoRetirada').addEventListener('submit', async (e) => {

e.preventDefault();

const senha = document.getElementById('editSenhaRetirada').value;

if (senha !== '123456') {

alert('Senha incorreta!');

return;

}

const id = document.getElementById('editRetiradaId').value;

const funcionario\_id = parseInt(document.getElementById('editFuncionarioRetirada').value);

const epi\_id = parseInt(document.getElementById('editEpiRetirada').value);

const motivoSelect = document.getElementById('editMotivoRetirada');

const motivo = motivoSelect.value;

const data\_retirada = document.getElementById('editDataRetirada').value;

const data\_proxima\_troca = document.getElementById('editDataProximaTroca').value;

const { error } = await supabase

.from('retiradas')

.update({

funcionario\_id,

epi\_id,

motivo,

data\_retirada,

data\_proxima\_troca

})

.eq('id', id);

if (error) {

console.error('Erro ao atualizar retirada:', error);

alert('Erro ao atualizar retirada!');

} else {

alert('Retirada atualizada com sucesso!');

fecharModalRetirada();

document.getElementById('btnCarregarRetiradas').click();

}

});

async function carregarMotivosSelect() {

const { data, error } = await supabase

.from('motivos')

.select('id, nome');

if (error) {

console.error('Erro ao carregar motivos:', error);

return;

}

const selects = ['motivoRetirada', 'editMotivoRetirada'];

selects.forEach(selectId => {

const select = document.getElementById(selectId);

select.innerHTML = '<option value="">Selecione o Motivo</option>';

if (data) {

data.forEach(motivo => {

const option = document.createElement('option');

option.value = motivo.nome;

option.textContent = motivo.nome;

select.appendChild(option);

});

}

});

}

async function carregarEPIsSelect() {

const { data, error } = await supabase

.from('epis')

.select('id, nome, tempo\_troca\_meses');

if (error) {

console.error('Erro ao carregar EPIs:', error);

return;

}

// Armazenar os meses de validade para cada EPI

window.episValidade = {};

if (data) {

data.forEach(epi => {

window.episValidade[epi.id] = epi.tempo\_troca\_meses;

});

}

const selects = ['epiRetirada', 'editEpiRetirada'];

selects.forEach(selectId => {

const select = document.getElementById(selectId);

select.innerHTML = '<option value="">Selecione o EPI</option>';

if (data) {

data.forEach(epi => {

const option = document.createElement('option');

option.value = epi.id;

option.textContent = epi.nome;

select.appendChild(option);

});

}

});

}

// Função de pesquisa para funcionários

function pesquisarFuncionarios() {

const termo = document.getElementById('pesquisaFuncionarios').value.toLowerCase();

const linhas = document.getElementById('tabelaFuncionarios').getElementsByTagName('tbody')[0].getElementsByTagName('tr');

for (let linha of linhas) {

const texto = linha.textContent.toLowerCase();

linha.style.display = texto.includes(termo) ? '' : 'none';

}

}

// Função de pesquisa para retiradas

function pesquisarRetiradas() {

const termo = document.getElementById('pesquisaRetiradas').value.toLowerCase();

const linhas = document.getElementById('tabelaRetiradas').getElementsByTagName('tbody')[0].getElementsByTagName('tr');

for (let linha of linhas) {

const texto = linha.textContent.toLowerCase();

linha.style.display = texto.includes(termo) ? '' : 'none';

}

}

async function atualizarDashboard() {

// Buscar dados

const { data: retiradas } = await supabase.from('retiradas').select('\*');

const { data: epis } = await supabase.from('epis').select('\*');

const { data: funcionarios } = await supabase.from('funcionarios').select('\*');

// Atualizar estatísticas

document.getElementById('totalEpis').textContent = epis?.length || 0;

document.getElementById('totalFuncionarios').textContent = funcionarios?.length || 0;

// Calcular trocas pendentes

const hoje = new Date();

const trocasPendentes = retiradas?.filter(r => new Date(r.data\_proxima\_troca) < hoje).length || 0;

document.getElementById('trocasPendentes').textContent = trocasPendentes;

// Gráfico de EPIs por Setor

new Chart(document.getElementById('episPorSetor'), {

type: 'pie',

data: {

labels: ['Produção', 'Manutenção', 'Logística'],

datasets: [{

data: [30, 20, 15],

backgroundColor: ['#FF6B00', '#002B4D', '#4CAF50']

}]

}

});

// Gráfico de Retiradas por Mês

new Chart(document.getElementById('retiradasPorMes'), {

type: 'line',

data: {

labels: ['Jan', 'Fev', 'Mar', 'Abr', 'Mai', 'Jun'],

datasets: [{

label: 'Retiradas',

data: [12, 19, 3, 5, 2, 3],

borderColor: '#FF6B00'

}]

}

});

// Gráfico Top EPIs

new Chart(document.getElementById('topEpis'), {

type: 'bar',

data: {

labels: ['Capacete', 'Luvas', 'Óculos', 'Protetor', 'Máscara'],

datasets: [{

label: 'Quantidade',

data: [25, 20, 15, 10, 8],

backgroundColor: '#002B4D'

}]

}

});

// Gráfico Status das Trocas

new Chart(document.getElementById('statusTrocas'), {

type: 'doughnut',

data: {

labels: ['Em Dia', 'Pendentes', 'Próximos'],

datasets: [{

data: [70, 15, 15],

backgroundColor: ['#4CAF50', '#FF6B00', '#FFC107']

}]

}

});

}

function toggleTheme() {

const button = document.querySelector('.theme-toggle');

if (document.body.getAttribute('data-theme') === 'dark') {

document.body.removeAttribute('data-theme');

button.textContent = '🌞 Modo Claro';

localStorage.setItem('theme', 'light');

} else {

document.body.setAttribute('data-theme', 'dark');

button.textContent = '🌙 Modo Escuro';

localStorage.setItem('theme', 'dark');

}

}

document.addEventListener('DOMContentLoaded', () => {

// Carregar tema salvo

const savedTheme = localStorage.getItem('theme');

if (savedTheme === 'dark') {

document.body.setAttribute('data-theme', 'dark');

document.querySelector('.theme-toggle').textContent = '🌙 Modo Escuro';

}

carregarSetores();

carregarFuncionariosSelect();

carregarEPIsSelect();

carregarMotivosSelect();

atualizarDashboard();

// Adicionar eventos de pesquisa

document.getElementById('pesquisaFuncionarios').addEventListener('input', pesquisarFuncionarios);

document.getElementById('pesquisaRetiradas').addEventListener('input', pesquisarRetiradas);

// Atualizar data próxima troca automaticamente

document.getElementById('epiRetirada').addEventListener('change', (e) => {

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

if (dataRetirada && e.target.value) {

const mesesValidade = window.episValidade[e.target.value];

if (mesesValidade) {

const dataProximaTroca = calcularDataProximaTroca(dataRetirada, mesesValidade);

document.getElementById('dataProximaTroca').value = dataProximaTroca;

}

}

});

document.getElementById('dataRetirada').addEventListener('change', (e) => {

const epiId = document.getElementById('epiRetirada').value;

if (epiId && e.target.value) {

const mesesValidade = window.episValidade[epiId];

if (mesesValidade) {

const dataProximaTroca = calcularDataProximaTroca(e.target.value, mesesValidade);

document.getElementById('dataProximaTroca').value = dataProximaTroca;

}

}

});

});

function calcularDataProximaTroca(dataRetirada, mesesValidade) {

const data = new Date(dataRetirada);

data.setMonth(data.getMonth() + mesesValidade);

return data.toISOString().split('T')[0];

}

// Formulário para adicionar novo funcionário

document.getElementById('formFuncionario').addEventListener('submit', async (e) => {

e.preventDefault();

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

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

const setor\_id = document.getElementById('setor').value;

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

const { data, error } = await supabase

.from('funcionarios')

.insert([

{

nome: nome,

cargo: cargo,

setor\_id: parseInt(setor\_id),

data\_admissao: dataAdmissao

}

]);

if (error) {

console.error('Erro ao adicionar funcionário:', error);

alert('Erro ao adicionar funcionário!');

} else {

alert('Funcionário adicionado com sucesso!');

document.getElementById('formFuncionario').reset();

document.getElementById('btnCarregarFuncionarios').click(); // Atualiza a tabela

}

});

**CSS:**@import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap');

:root {

--bg-color: #f8fafc;

--text-color: #1e293b;

--header-bg: #002B4D;

--card-bg: #ffffff;

--hover-bg: #f8fafc;

--border-color: #e2e8f0;

}

[data-theme="dark"] {

--bg-color: #1a1a1a;

--text-color: #e2e8f0;

--header-bg: #001a2e;

--card-bg: #2d2d2d;

--hover-bg: #333333;

--border-color: #404040;

}

body {

font-family: 'Inter', sans-serif;

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

color: var(--text-color);

margin: 0;

padding: 0;

font-size: 16px;

line-height: 1.6;

transition: background-color 0.3s, color 0.3s;

}

header {

background-color: var(--header-bg);

padding: 1rem 0;

text-align: center;

}

.theme-toggle {

position: fixed;

top: 1rem;

right: 1rem;

padding: 0.5rem 1rem;

border-radius: 20px;

background: var(--card-bg);

color: var(--text-color);

border: 1px solid var(--border-color);

cursor: pointer;

z-index: 1000;

}

.logo-container {

max-width: 1200px;

margin: 0 auto;

padding: 0 1rem;

}

.logo {

height: 120px;

width: auto;

will-change: transform;

max-width: 100%;

}

.logo-container img {

display: block;

}

.hero {

background-color: #002B4D;

color: white;

text-align: center;

padding: 3rem 1rem;

}

.hero h1 {

font-size: 3rem;

margin: 0;

color: white;

}

.subtitle {

font-size: 1.2rem;

margin-top: 1rem;

opacity: 0.9;

}

.about {

background-color: white;

padding: 4rem 1rem;

}

.container {

max-width: 800px;

margin: 0 auto;

}

main {

min-height: 100vh;

}

h1 {

font-size: 2.5rem;

font-weight: 700;

color: #1e293b;

margin: 2rem 0;

text-align: center;

}

h2 {

font-size: 1.5rem;

font-weight: 600;

color: #334155;

margin: 1.5rem 0;

}

#dashboard {

width: 95%;

max-width: 1200px;

margin: 2rem auto;

background-color: #ffffff;

border-radius: 16px;

padding: 2rem;

box-shadow: 0 4px 6px -1px rgb(0 0 0 / 0.1), 0 2px 4px -2px rgb(0 0 0 / 0.1);

}

form {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 1rem;

margin-bottom: 2rem;

background-color: #f1f5f9;

padding: 1.5rem;

border-radius: 12px;

}

input, select {

padding: 0.75rem 1rem;

font-size: 0.95rem;

border-radius: 8px;

border: 1px solid #e2e8f0;

background-color: #ffffff;

color: #1e293b;

width: 100%;

transition: all 0.2s ease;

}

input:focus, select:focus {

border-color: #970700;

box-shadow: 0 0 0 3px rgba(151, 7, 0, 0.1);

outline: none;

}

button {

padding: 0.75rem 1.5rem;

font-size: 0.95rem;

font-weight: 500;

border-radius: 8px;

border: none;

cursor: pointer;

transition: all 0.2s ease;

color: #ffffff;

background: #ff6b00;

display: inline-flex;

align-items: center;

justify-content: center;

gap: 0.5rem;

}

button:hover {

background: #ff8533;

transform: translateY(-1px);

}

button:active {

transform: translateY(0);

}

table {

width: 100%;

margin-top: 1.5rem;

border-collapse: separate;

border-spacing: 0;

background-color: #ffffff;

border-radius: 12px;

overflow: hidden;

box-shadow: 0 1px 3px 0 rgb(0 0 0 / 0.1);

}

th, td {

padding: 1rem;

text-align: left;

border-bottom: 1px solid #e2e8f0;

}

th {

background-color: var(--header-bg);

font-weight: 600;

color: var(--text-color);

text-transform: uppercase;

font-size: 0.85rem;

letter-spacing: 0.05em;

}

tr:last-child td {

border-bottom: none;

}

tr:hover {

background-color: #f8fafc;

}

#btnCarregarFuncionarios, #btnCarregarRetiradas {

margin: 0 0.5rem 1rem 0;

}

.cta-container {

text-align: center;

margin-top: 2rem;

}

.cta-button {

display: inline-block;

padding: 1rem 2rem;

font-size: 1.2rem;

font-weight: 600;

color: white;

background-color: #ff6b00;

text-decoration: none;

border-radius: 8px;

transition: all 0.2s ease;

}

.cta-button:hover {

background-color: #ff8533;

transform: translateY(-2px);

}

.nav-link {

color: white;

text-decoration: none;

margin-left: 1rem;

font-weight: 500;

}

.nav-link:hover {

text-decoration: underline;

}

.modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.5);

z-index: 1000;

}

.modal-content {

background-color: white;

padding: 2rem;

border-radius: 12px;

width: 90%;

max-width: 500px;

margin: 50px auto;

}

.modal-buttons {

display: flex;

gap: 1rem;

justify-content: flex-end;

margin-top: 1rem;

}

.action-buttons {

display: flex;

gap: 0.5rem;

}

.btn-edit, .btn-delete {

padding: 0.5rem;

font-size: 0.9rem;

}

.btn-delete {

background-color: #dc2626;

}

.btn-delete:hover {

background-color: #b91c1c;

}

.dashboard-container {

display: grid;

grid-template-columns: repeat(2, 1fr);

gap: 1.5rem;

margin-top: 2rem;

}

.dashboard-card {

background: var(--card-bg);

padding: 1.5rem;

border-radius: 12px;

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

}

.stats-container {

grid-column: 1 / -1;

display: grid;

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

gap: 1rem;

margin-top: 1rem;

}

.stat-card {

background: #002B4D;

color: white;

padding: 1.5rem;

border-radius: 12px;

text-align: center;

}

.stat-card h4 {

margin: 0;

font-size: 1rem;

opacity: 0.9;

}

.stat-card p {

margin: 0.5rem 0 0;

font-size: 2rem;

font-weight: bold;

}

@media (max-width: 768px) {

#dashboard {

padding: 1rem;

}

form {

grid-template-columns: 1fr;

}

h1 {

font-size: 2rem;

}

}