# Arquivo: .env

EMAIL\_USER=oilsmartoficiall@gmail.com

EMAIL\_PASS=cyog lhar hnpm plgw

PORT=3000

# Arquivo: database/db.js

const mysql = require('mysql2');

const db = mysql.createConnection({

host: 'localhost',

user: 'root',

password: '277353Hp', // coloque sua senha aqui

database: 'Oil'

});

// const db = mysql.createConnection({

// host: 'localhost',

// user: 'root',

// password: 'Root@123', // coloque sua senha aqui

// database: 'Oil'

// });

db.connect((err) => {

if (err) {

console.error('Erro ao conectar no banco:', err);

} else {

console.log('MySQL conectado com sucesso!');

}

});

module.exports = db;

# Arquivo: database/oil.sql

-- Criação do banco de dados OilSmart

CREATE DATABASE IF NOT EXISTS Oil;

USE Oil;

-- Tabela de usuários

CREATE TABLE usuario (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

senha VARCHAR(255) NOT NULL,

tipo ENUM('cliente', 'oficina', 'funcionario') NOT NULL,

telefone VARCHAR(20),

cpf VARCHAR(14) UNIQUE NULL,

cnpj VARCHAR(18) UNIQUE NULL,

endereco VARCHAR(255),

cep VARCHAR(9),

cidade VARCHAR(100),

estado VARCHAR(2),

data\_cadastro DATETIME DEFAULT CURRENT\_TIMESTAMP

);

-- Tabela de marcas de veículos

CREATE TABLE marca (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL UNIQUE

);

-- Tabela de modelos de veículos

CREATE TABLE modelo (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

marca\_id INT NOT NULL,

tipo ENUM('carro', 'moto') NOT NULL,

FOREIGN KEY (marca\_id) REFERENCES marca(id) ON DELETE CASCADE

);

-- Tabela de anos dos modelos

CREATE TABLE modelo\_ano (

id INT AUTO\_INCREMENT PRIMARY KEY,

modelo\_id INT NOT NULL,

ano INT NOT NULL,

FOREIGN KEY (modelo\_id) REFERENCES modelo(id) ON DELETE CASCADE

);

-- Tabela de veículos dos clientes

CREATE TABLE veiculo (

id INT AUTO\_INCREMENT PRIMARY KEY,

usuario\_id INT NOT NULL,

modelo\_ano\_id INT NOT NULL,

placa VARCHAR(10),

quilometragem INT,

data\_cadastro DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (usuario\_id) REFERENCES usuario(id) ON DELETE CASCADE,

FOREIGN KEY (modelo\_ano\_id) REFERENCES modelo\_ano(id) ON DELETE CASCADE

);

-- Tabela de oficinas

CREATE TABLE oficina (

id INT AUTO\_INCREMENT PRIMARY KEY,

usuario\_id INT NOT NULL,

nome VARCHAR(100) NOT NULL,

cep VARCHAR(9) NOT NULL,

endereco VARCHAR(255),

cidade VARCHAR(100),

estado VARCHAR(2),

horario\_abertura TIME,

horario\_fechamento TIME,

dias\_funcionamento VARCHAR(50),

lat DECIMAL(10,8),

lng DECIMAL(11,8),

FOREIGN KEY (usuario\_id) REFERENCES usuario(id) ON DELETE CASCADE

);

-- Tabela de funcionários

CREATE TABLE funcionario (

id INT AUTO\_INCREMENT PRIMARY KEY,

usuario\_id INT NOT NULL,

oficina\_id INT NOT NULL,

cargo VARCHAR(50),

FOREIGN KEY (usuario\_id) REFERENCES usuario(id) ON DELETE CASCADE,

FOREIGN KEY (oficina\_id) REFERENCES oficina(id) ON DELETE CASCADE

);

-- Tabela de produtos (óleos)

CREATE TABLE produto\_oleo (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

tipo ENUM('carro', 'moto') NOT NULL,

viscosidade VARCHAR(20),

especificacao VARCHAR(255),

marca VARCHAR(100),

preco DECIMAL(10,2) DEFAULT 0

);

-- Tabela de produtos (filtros)

CREATE TABLE produto\_filtro (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

tipo ENUM('carro', 'moto') NOT NULL,

compatibilidade\_modelo VARCHAR(255),

preco DECIMAL(10,2) DEFAULT 0

);

-- Tabela de estoque

CREATE TABLE estoque (

id INT AUTO\_INCREMENT PRIMARY KEY,

oficina\_id INT NOT NULL,

produto\_id INT NOT NULL,

tipo\_produto ENUM('oleo', 'filtro') NOT NULL,

quantidade INT NOT NULL DEFAULT 0,

FOREIGN KEY (oficina\_id) REFERENCES oficina(id) ON DELETE CASCADE

);

-- Tabela de recomendações

CREATE TABLE recomendacao (

id INT AUTO\_INCREMENT PRIMARY KEY,

modelo\_ano\_id INT NOT NULL,

oleo\_id INT,

filtro\_id INT,

FOREIGN KEY (modelo\_ano\_id) REFERENCES modelo\_ano(id) ON DELETE CASCADE,

FOREIGN KEY (oleo\_id) REFERENCES produto\_oleo(id) ON DELETE SET NULL,

FOREIGN KEY (filtro\_id) REFERENCES produto\_filtro(id) ON DELETE SET NULL

);

-- Tabela de agendamento simples

CREATE TABLE agendamento\_simples (

id INT AUTO\_INCREMENT PRIMARY KEY,

protocolo VARCHAR(20) NOT NULL UNIQUE,

data\_hora DATETIME,

oficina\_nome VARCHAR(100),

oficina\_endereco VARCHAR(255),

oficina\_telefone VARCHAR(20),

veiculo VARCHAR(255),

servicos TEXT,

total\_servico DECIMAL(10, 2),

cliente\_nome VARCHAR(100),

cliente\_cpf VARCHAR(14),

cliente\_telefone VARCHAR(20),

cliente\_email VARCHAR(100),

usuario\_id INT,

FOREIGN KEY (usuario\_id) REFERENCES usuario(id) ON DELETE SET NULL

);

-- Tabela de divergências

CREATE TABLE divergencia (

id INT AUTO\_INCREMENT PRIMARY KEY,

agendamento\_id INT NOT NULL,

descricao TEXT NOT NULL,

resolvido BOOLEAN DEFAULT FALSE,

data\_criacao DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (agendamento\_id) REFERENCES agendamento\_simples(id) ON DELETE CASCADE

);

-- ========================================

-- INSERÇÃO DE DADOS INICIAIS

-- ========================================

-- Inserir usuários administradores das oficinas

INSERT INTO usuario (nome, email, senha, tipo, telefone, cnpj) VALUES

("Oficina Norte", "adm\_norte@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 1234-5678", "12.345.678/0001-90"),

("Oficina Sul", "adm\_sul@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 2345-6789", "23.456.789/0001-01"),

("Oficina Leste", "adm\_leste@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 3456-7890", "34.567.890/0001-12");

-- Inserir oficinas

INSERT INTO oficina (usuario\_id, nome, cep, endereco, cidade, estado, horario\_abertura, horario\_fechamento, dias\_funcionamento, lat, lng) VALUES

(1, "Motul Oficina Norte", "02430-000", "Rua Exemplo Norte, 123", "São Paulo", "SP", "08:00:00", "17:00:00", "Seg-Sex", -23.550520, -46.633308),

(2, "Motul Oficina Sul", "04783-100", "Rua Exemplo Sul, 456", "São Paulo", "SP", "09:00:00", "18:00:00", "Seg-Sab", -23.563099, -46.654279),

(3, "Motul Oficina Leste", "03675-040", "Rua Exemplo Leste, 789", "São Paulo", "SP", "10:00:00", "20:00:00", "Seg-Sex", -23.548943, -46.638818);

-- Inserir marcas de veículos

INSERT INTO marca (nome) VALUES

("Fiat"), ("Ford"), ("Chevrolet"), ("Volkswagen"), ("Toyota"),

("Honda"), ("Hyundai"), ("Renault"), ("Peugeot"), ("Nissan"), ("Jeep"),

("Yamaha"), ("Suzuki"), ("Kawasaki"), ("Harley-Davidson"),

("BMW"), ("Mercedes-Benz"), ("Audi"), ("Porsche");

-- Inserir modelos de carros

INSERT INTO modelo (nome, marca\_id, tipo) VALUES

("Palio", 1, "carro"),

("Uno", 1, "carro"),

("Strada", 1, "carro"),

("Toro", 1, "carro"),

("Ka", 2, "carro"),

("Fiesta", 2, "carro"),

("EcoSport", 2, "carro"),

("Focus", 2, "carro"),

("Onix", 3, "carro"),

("Celta", 3, "carro"),

("Prisma", 3, "carro"),

("S10", 3, "carro"),

("Gol", 4, "carro"),

("Voyage", 4, "carro"),

("Saveiro", 4, "carro"),

("Polo", 4, "carro"),

("Corolla", 5, "carro"),

("Hilux", 5, "carro"),

("Etios", 5, "carro"),

("Yaris", 5, "carro"),

("Civic", 6, "carro"),

("Fit", 6, "carro"),

("HR-V", 6, "carro"),

("CR-V", 6, "carro"),

("HB20", 7, "carro"),

("Creta", 7, "carro"),

("Tucson", 7, "carro"),

("Santa Fe", 7, "carro"),

("Kwid", 8, "carro"),

("Sandero", 8, "carro"),

("Logan", 8, "carro"),

("Duster", 8, "carro"),

("208", 9, "carro"),

("2008", 9, "carro"),

("3008", 9, "carro"),

("Partner", 9, "carro"),

("March", 10, "carro"),

("Versa", 10, "carro"),

("Kicks", 10, "carro"),

("Frontier", 10, "carro"),

("Renegade", 11, "carro"),

("Compass", 11, "carro"),

("Commander", 11, "carro"),

("Wrangler", 11, "carro"),

("X5", 16, "carro"),

("S-Class", 17, "carro"),

("A4", 18, "carro"),

("911", 19, "carro");

-- Inserir modelos de motos

INSERT INTO modelo (nome, marca\_id, tipo) VALUES

("CG 160", 12, "moto"),

("Fazer 250", 12, "moto"),

("MT-03", 12, "moto"),

("XTZ 250", 12, "moto"),

("Bandit 125", 13, "moto"),

("Burgman 125", 13, "moto"),

("GSX-S750", 13, "moto"),

("V-Strom 650", 13, "moto"),

("Ninja 300", 14, "moto"),

("Z400", 14, "moto"),

("Versys 650", 14, "moto"),

("Z900", 14, "moto"),

("Iron 883", 15, "moto"),

("Street 750", 15, "moto"),

("Fat Boy", 15, "moto"),

("Road King", 15, "moto"),

("CBR 600RR", 6, "moto"),

("CB 500F", 6, "moto");

-- Inserir anos dos modelos (simplificado para os principais modelos)

INSERT INTO modelo\_ano (modelo\_id, ano) VALUES

-- Fiat

(1, 2020), (1, 2021), (1, 2022), (1, 2023), (1, 2024),

(2, 2020), (2, 2021), (2, 2022), (2, 2023), (2, 2024),

(3, 2021), (3, 2022), (3, 2023), (3, 2024),

(4, 2020), (4, 2021), (4, 2022), (4, 2023), (4, 2024),

-- Ford

(5, 2021), (5, 2022), (5, 2023), (5, 2024),

(6, 2020), (6, 2021), (6, 2022), (6, 2023), (6, 2024),

(7, 2021), (7, 2022), (7, 2023), (7, 2024),

(8, 2020), (8, 2021), (8, 2022), (8, 2023), (8, 2024),

-- Chevrolet

(9, 2021), (9, 2022), (9, 2023), (9, 2024),

(10, 2020), (10, 2021), (10, 2022), (10, 2023), (10, 2024),

(11, 2021), (11, 2022), (11, 2023), (11, 2024),

(12, 2020), (12, 2021), (12, 2022), (12, 2023), (12, 2024),

-- Volkswagen

(13, 2020), (13, 2021), (13, 2022), (13, 2023), (13, 2024),

(14, 2020), (14, 2021), (14, 2022), (14, 2023), (14, 2024),

(15, 2020), (15, 2021), (15, 2022), (15, 2023), (15, 2024),

(16, 2020), (16, 2021), (16, 2022), (16, 2023), (16, 2024),

-- Toyota

(17, 2020), (17, 2021), (17, 2022), (17, 2023), (17, 2024),

(18, 2020), (18, 2021), (18, 2022), (18, 2023), (18, 2024),

(19, 2020), (19, 2021), (19, 2022), (19, 2023), (19, 2024),

(20, 2020), (20, 2021), (20, 2022), (20, 2023), (20, 2024),

-- Honda carros

(21, 2020), (21, 2021), (21, 2022), (21, 2023), (21, 2024),

(22, 2020), (22, 2021), (22, 2022), (22, 2023), (22, 2024),

(23, 2020), (23, 2021), (23, 2022), (23, 2023), (23, 2024),

(24, 2020), (24, 2021), (24, 2022), (24, 2023), (24, 2024),

-- Hyundai

(25, 2020), (25, 2021), (25, 2022), (25, 2023), (25, 2024),

(26, 2020), (26, 2021), (26, 2022), (26, 2023), (26, 2024),

(27, 2020), (27, 2021), (27, 2022), (27, 2023), (27, 2024),

(28, 2020), (28, 2021), (28, 2022), (28, 2023), (28, 2024),

-- Renault

(29, 2020), (29, 2021), (29, 2022), (29, 2023), (29, 2024),

(30, 2020), (30, 2021), (30, 2022), (30, 2023), (30, 2024),

(31, 2020), (31, 2021), (31, 2022), (31, 2023), (31, 2024),

(32, 2020), (32, 2021), (32, 2022), (32, 2023), (32, 2024),

-- Peugeot

(33, 2020), (33, 2021), (33, 2022), (33, 2023), (33, 2024),

(34, 2020), (34, 2021), (34, 2022), (34, 2023), (34, 2024),

(35, 2020), (35, 2021), (35, 2022), (35, 2023), (35, 2024),

(36, 2020), (36, 2021), (36, 2022), (36, 2023), (36, 2024),

-- Nissan

(37, 2020), (37, 2021), (37, 2022), (37, 2023), (37, 2024),

(38, 2020), (38, 2021), (38, 2022), (38, 2023), (38, 2024),

(39, 2020), (39, 2021), (39, 2022), (39, 2023), (39, 2024),

(40, 2020), (40, 2021), (40, 2022), (40, 2023), (40, 2024),

-- Jeep

(41, 2020), (41, 2021), (41, 2022), (41, 2023), (41, 2024),

(42, 2020), (42, 2021), (42, 2022), (42, 2023), (42, 2024),

(43, 2020), (43, 2021), (43, 2022), (43, 2023), (43, 2024),

(44, 2020), (44, 2021), (44, 2022), (44, 2023), (44, 2024),

-- BMW, Mercedes, Audi, Porsche

(45, 2020), (45, 2021), (45, 2022), (45, 2023), (45, 2024),

(46, 2020), (46, 2021), (46, 2022), (46, 2023), (46, 2024),

(47, 2020), (47, 2021), (47, 2022), (47, 2023), (47, 2024),

(48, 2020), (48, 2021), (48, 2022), (48, 2023), (48, 2024),

-- Motos Yamaha

(49, 2020), (49, 2021), (49, 2022), (49, 2023), (49, 2024),

(50, 2020), (50, 2021), (50, 2022), (50, 2023), (50, 2024),

(51, 2020), (51, 2021), (51, 2022), (51, 2023), (51, 2024),

(52, 2020), (52, 2021), (52, 2022), (52, 2023), (52, 2024),

-- Motos Suzuki

(53, 2020), (53, 2021), (53, 2022), (53, 2023), (53, 2024),

(54, 2020), (54, 2021), (54, 2022), (54, 2023), (54, 2024),

(55, 2020), (55, 2021), (55, 2022), (55, 2023), (55, 2024),

(56, 2020), (56, 2021), (56, 2022), (56, 2023), (56, 2024),

-- Motos Kawasaki

(57, 2020), (57, 2021), (57, 2022), (57, 2023), (57, 2024),

(58, 2020), (58, 2021), (58, 2022), (58, 2023), (58, 2024),

(59, 2020), (59, 2021), (59, 2022), (59, 2023), (59, 2024),

(60, 2020), (60, 2021), (60, 2022), (60, 2023), (60, 2024),

-- Motos Harley-Davidson

(61, 2020), (61, 2021), (61, 2022), (61, 2023), (61, 2024),

(62, 2020), (62, 2021), (62, 2022), (62, 2023), (62, 2024),

(63, 2020), (63, 2021), (63, 2022), (63, 2023), (63, 2024),

(64, 2020), (64, 2021), (64, 2022), (64, 2023), (64, 2024),

-- Motos Honda

(65, 2020), (65, 2021), (65, 2022), (65, 2023), (65, 2024),

(66, 2020), (66, 2021), (66, 2022), (66, 2023), (66, 2024);

-- Inserção de óleos (DEVE VIR ANTES DAS RECOMENDAÇÕES)

INSERT INTO produto\_oleo (nome, tipo, viscosidade, especificacao, marca, preco) VALUES

("Óleo Sintético 5W-30", "carro", "5W-30", "API SN/CF", "Motul", 85.00),

("Óleo Semissintético 10W-40", "carro", "10W-40", "API SL/CF", "Castrol", 60.00),

("Óleo Sintético 10W-30 Moto", "moto", "10W-30", "JASO MA2", "Mobil", 75.00),

("Óleo Mineral 20W-50", "carro", "20W-50", "API SJ", "Petronas", 45.00),

("Óleo Sintético 5W-40", "carro", "5W-40", "API SN/CF", "Shell", 90.00),

("Óleo Sintético 15W-50 Moto", "moto", "15W-50", "JASO MA2", "Motul", 80.00),

("Óleo Sintético 0W-20", "carro", "0W-20", "API SP", "Idemitsu", 110.00),

("Óleo Sintético 5W-20", "carro", "5W-20", "API SN", "Valvoline", 80.00),

("Óleo Sintético 5W-30 Diesel", "carro", "5W-30", "ACEA C3", "Total", 95.00),

("Óleo Sintético 10W-50 Moto", "moto", "10W-50", "JASO MA2", "Castrol", 85.00),

("Óleo Sintético 0W-30", "carro", "0W-30", "API SP", "Motul", 120.00),

("Óleo Sintético 5W-50", "carro", "5W-50", "API SN", "Shell", 100.00),

("Óleo Semissintético 15W-40", "carro", "15W-40", "API SL", "Petronas", 55.00),

("Óleo Sintético 10W-40 Moto", "moto", "10W-40", "JASO MA2", "Yamaha", 70.00),

("Óleo Sintético 20W-50 Moto", "moto", "20W-50", "JASO MA2", "Honda", 65.00);

-- Inserção de filtros (DEVE VIR ANTES DAS RECOMENDAÇÕES)

INSERT INTO produto\_filtro (nome, tipo, compatibilidade\_modelo, preco) VALUES

("Filtro de Óleo Universal Carro Pequeno", "carro", "Compatível com Fiat Palio, Uno, VW Gol, Chevrolet Onix", 30.00),

("Filtro de Óleo Universal Carro Médio", "carro", "Compatível com Ford Focus, Honda Civic, Toyota Corolla", 35.00),

("Filtro de Óleo Universal Carro Grande/SUV", "carro", "Compatível com Jeep Compass, Toyota Hilux, Chevrolet S10", 40.00),

("Filtro de Óleo Universal Moto Pequena", "moto", "Compatível com Honda CG 160, Yamaha Fazer 250", 25.00),

("Filtro de Ar Universal Carro Pequeno", "carro", "Compatível com Fiat Palio, Uno, VW Gol, Chevrolet Onix", 45.00),

("Filtro de Ar Universal Carro Médio", "carro", "Compatível com Ford Focus, Honda Civic, Toyota Corolla", 50.00),

("Filtro de Ar Universal Carro Grande/SUV", "carro", "Compatível com Jeep Compass, Toyota Hilux, Chevrolet S10", 55.00),

("Filtro de Ar Universal Moto", "moto", "Compatível com Honda CG 160, Yamaha Fazer 250", 30.00),

("Filtro de Combustível Universal Carro", "carro", "Compatível com a maioria dos carros a gasolina/etanol", 20.00),

("Filtro de Combustível Universal Moto", "moto", "Compatível com a maioria das motos", 18.00),

("Filtro de Cabine Universal Carro", "carro", "Compatível com a maioria dos carros", 60.00),

("Filtro de Óleo Esportivo Carro", "carro", "Compatível com carros esportivos", 70.00),

("Filtro de Ar Esportivo Carro", "carro", "Compatível com carros esportivos", 80.00),

("Filtro de Óleo para Diesel", "carro", "Compatível com veículos a diesel", 45.00),

("Filtro de Ar para Diesel", "carro", "Compatível com veículos a diesel", 50.00),

("Filtro de Óleo Premium Carro", "carro", "Compatível com carros premium", 65.00);

-- Inserção de usuários clientes

INSERT INTO usuario (nome, email, senha, tipo, telefone, cpf, endereco, cep, cidade, estado) VALUES

("João Silva", "joao.silva@email.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "cliente", "(11) 98765-4321", "111.111.111-11", "Rua A, 1", "01000-001", "São Paulo", "SP"),

("Maria Santos", "maria.santos@email.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "cliente", "(21) 98765-6789", "222.222.222-22", "Rua B, 2", "20000-002", "Rio de Janeiro", "RJ"),

("Lucas Santos", "lucas.santos@email.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "cliente", "(31) 98765-9012", "333.333.333-33", "Rua C, 3", "30000-003", "Belo Horizonte", "MG"),

("Julia Costa", "julia.costa@email.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "cliente", "(41) 98765-3456", "444.444.444-44", "Rua D, 4", "80000-004", "Curitiba", "PR"),

("Gabriel Almeida", "gabriel.almeida@email.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "cliente", "(51) 98765-7890", "555.555.555-55", "Rua E, 5", "90000-005", "Porto Alegre", "RS");

-- Mais usuários de exemplo (oficinas)

INSERT INTO usuario (nome, email, senha, tipo, telefone, cnpj) VALUES

("Oficina Centro", "adm\_centro@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 4567-8901", "45.678.901/0001-23"),

("Oficina Oeste", "adm\_oeste@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 5678-9012", "56.789.012/0001-34");

-- Mais oficinas (associando aos usuários recém-criados)

INSERT INTO oficina (usuario\_id, nome, cep, endereco, cidade, estado, horario\_abertura, horario\_fechamento, dias\_funcionamento, lat, lng) VALUES

(9, "Motul Oficina Centro", "01000-000", "Rua do Centro, 100", "São Paulo", "SP", "08:30:00", "17:30:00", "Seg-Sex", -23.550520, -46.633308),

(10, "Motul Oficina Oeste", "05000-000", "Av. Oeste, 200", "São Paulo", "SP", "09:00:00", "18:00:00", "Seg-Sab", -23.563099, -46.654279);

-- Mais usuários de exemplo (funcionários)

INSERT INTO usuario (nome, email, senha, tipo, telefone, cpf) VALUES

("Roberto Carlos", "roberto.carlos@oficinacentro.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "funcionario", "(11) 99887-7766", "666.777.888-99"),

("Patricia Souza", "patricia.souza@oficinaoeste.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "funcionario", "(11) 99776-6655", "777.888.999-00");

-- Mais funcionários (associando aos usuários recém-criados e oficinas)

INSERT INTO funcionario (usuario\_id, oficina\_id, cargo) VALUES

(11, 4, "Mecânico"),

(12, 5, "Atendente");

-- RECOMENDAÇÕES (AGORA COM IDs CORRETOS)

-- Fiat (marca\_id = 1) - usando óleo 2 e filtro 1

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 2, 1 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 1 AND m.tipo = "carro";

-- Ford (marca\_id = 2) - usando óleo 1 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 1, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 2 AND m.tipo = "carro";

-- Chevrolet (marca\_id = 3) - usando óleo 2 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 2, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 3 AND m.tipo = "carro";

-- Volkswagen (marca\_id = 4) - usando óleo 1 e filtro 1

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 1, 1 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 4 AND m.tipo = "carro";

-- Toyota (marca\_id = 5) - usando óleo 5 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 5, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 5 AND m.tipo = "carro";

-- Honda carros (marca\_id = 6) - usando óleo 5 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 5, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 6 AND m.tipo = "carro";

-- Honda motos (marca\_id = 6) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 6 AND m.tipo = "moto";

-- Hyundai (marca\_id = 7) - usando óleo 1 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 1, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 7 AND m.tipo = "carro";

-- Renault (marca\_id = 8) - usando óleo 2 e filtro 1

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 2, 1 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 8 AND m.tipo = "carro";

-- Peugeot (marca\_id = 9) - usando óleo 5 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 5, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 9 AND m.tipo = "carro";

-- Nissan (marca\_id = 10) - usando óleo 5 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 5, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 10 AND m.tipo = "carro";

-- Jeep (marca\_id = 11) - usando óleo 5 e filtro 3

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 5, 3 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 11 AND m.tipo = "carro";

-- Yamaha (marca\_id = 12) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 12 AND m.tipo = "moto";

-- Suzuki (marca\_id = 13) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 13 AND m.tipo = "moto";

-- Kawasaki (marca\_id = 14) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 14 AND m.tipo = "moto";

-- Harley-Davidson (marca\_id = 15) - usando óleo 6 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 6, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 15 AND m.tipo = "moto";

-- BMW (marca\_id = 16) - usando óleo 11 e filtro 12

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 11, 12 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 16 AND m.tipo = "carro";

-- Mercedes-Benz (marca\_id = 17) - usando óleo 11 e filtro 12

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 11, 12 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 17 AND m.tipo = "carro";

-- Audi (marca\_id = 18) - usando óleo 11 e filtro 12

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 11, 12 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 18 AND m.tipo = "carro";

-- Porsche (marca\_id = 19) - usando óleo 12 e filtro 12

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 12, 12 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 19 AND m.tipo = "carro";

-- Estoque para Motul Oficina Norte (id 1)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(1, 1, 'oleo', 50),

(1, 2, 'oleo', 40),

(1, 1, 'filtro', 60),

(1, 6, 'filtro', 30),

(1, 7, 'oleo', 25),

(1, 9, 'filtro', 45);

-- Estoque para Motul Oficina Sul (id 2)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(2, 4, 'oleo', 35),

(2, 3, 'oleo', 30),

(2, 2, 'filtro', 50),

(2, 4, 'filtro', 40),

(2, 5, 'oleo', 20),

(2, 11, 'filtro', 35);

-- Estoque para Motul Oficina Leste (id 3)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(3, 1, 'oleo', 45),

(3, 6, 'oleo', 25),

(3, 3, 'filtro', 55),

(3, 8, 'filtro', 30),

(3, 8, 'oleo', 30),

(3, 12, 'filtro', 20);

-- Estoque para Motul Oficina Centro (id 4)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(4, 2, 'oleo', 30),

(4, 7, 'oleo', 20),

(4, 1, 'filtro', 40),

(4, 5, 'filtro', 25),

(4, 9, 'oleo', 15),

(4, 10, 'filtro', 30);

-- Estoque para Motul Oficina Oeste (id 5)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(5, 4, 'oleo', 25),

(5, 10, 'oleo', 15),

(5, 3, 'filtro', 30),

(5, 13, 'filtro', 10),

(5, 5, 'oleo', 20),

(5, 14, 'filtro', 25);

-- Adicione a coluna telefone na tabela oficina

ALTER TABLE oficina ADD COLUMN telefone VARCHAR(20) AFTER estado;

-- Atualize os telefones das oficinas existentes

UPDATE oficina SET telefone = '(11) 3333-0001' WHERE id = 1;

UPDATE oficina SET telefone = '(11) 3333-0002' WHERE id = 2;

UPDATE oficina SET telefone = '(11) 3333-0003' WHERE id = 3;

UPDATE oficina SET telefone = '(11) 3333-0004' WHERE id = 4;

UPDATE oficina SET telefone = '(11) 3333-0005' WHERE id = 5;

-- aaaaaaaaaa

-- Inserir 10 novos usuários para as oficinas

INSERT INTO usuario (nome, email, senha, tipo, telefone, cnpj) VALUES

("Oficina Paulista", "adm\_paulista@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 6789-0123", "67.890.123/0001-45"),

("Oficina Ipiranga", "adm\_ipiranga@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 7890-1234", "78.901.234/0001-56"),

("Oficina Tatuapé", "adm\_tatuape@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 8901-2345", "89.012.345/0001-67"),

("Oficina Mooca", "adm\_mooca@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9012-3456", "90.123.456/0001-78"),

("Oficina Santana", "adm\_santana@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9123-4567", "91.234.567/0001-89"),

("Oficina Pinheiros", "adm\_pinheiros@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9234-5678", "92.345.678/0001-90"),

("Oficina Vila Madalena", "adm\_vilamada@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9345-6789", "93.456.789/0001-01"),

("Oficina Itaim Bibi", "adm\_itaim@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9456-7890", "94.567.890/0001-12"),

("Oficina Jardins", "adm\_jardins@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9567-8901", "95.678.901/0001-23"),

("Oficina Morumbi", "adm\_morumbi@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9678-9012", "96.789.012/0001-34");

-- Inserir as 10 novas oficinas

INSERT INTO oficina (usuario\_id, nome, cep, endereco, cidade, estado, telefone, horario\_abertura, horario\_fechamento, dias\_funcionamento, lat, lng) VALUES

(13, "Motul Oficina Paulista", "01310-000", "Av. Paulista, 1000", "São Paulo", "SP", "(11) 3333-0006", "08:00:00", "18:00:00", "Seg-Sab", -23.563099, -46.654279),

(14, "Motul Oficina Ipiranga", "04280-000", "Av. Ipiranga, 200", "São Paulo", "SP", "(11) 3333-0007", "08:30:00", "17:30:00", "Seg-Sex", -23.580000, -46.640000),

(15, "Motul Oficina Tatuapé", "03310-000", "Rua Tuiuti, 500", "São Paulo", "SP", "(11) 3333-0008", "09:00:00", "18:00:00", "Seg-Sex", -23.540000, -46.575000),

(16, "Motul Oficina Mooca", "03110-000", "Rua da Mooca, 300", "São Paulo", "SP", "(11) 3333-0009", "08:00:00", "17:00:00", "Seg-Sex", -23.550000, -46.600000),

(17, "Motul Oficina Santana", "02030-000", "Av. Eng. Caetano Álvares, 400", "São Paulo", "SP", "(11) 3333-0010", "08:30:00", "17:30:00", "Seg-Sex", -23.500000, -46.625000),

(18, "Motul Oficina Pinheiros", "05410-000", "Rua dos Pinheiros, 600", "São Paulo", "SP", "(11) 3333-0011", "09:00:00", "19:00:00", "Seg-Sab", -23.565000, -46.690000),

(19, "Motul Oficina Vila Madalena", "05435-000", "Rua Aspicuelta, 700", "São Paulo", "SP", "(11) 3333-0012", "10:00:00", "20:00:00", "Seg-Sab", -23.550000, -46.690000),

(20, "Motul Oficina Itaim Bibi", "04551-000", "Rua João Cachoeira, 800", "São Paulo", "SP", "(11) 3333-0013", "08:00:00", "18:00:00", "Seg-Sex", -23.585000, -46.680000),

(21, "Motul Oficina Jardins", "01410-000", "Alameda Santos, 900", "São Paulo", "SP", "(11) 3333-0014", "08:30:00", "17:30:00", "Seg-Sex", -23.560000, -46.655000),

(22, "Motul Oficina Morumbi", "05710-000", "Av. Morumbi, 1000", "São Paulo", "SP", "(11) 3333-0015", "09:00:00", "18:00:00", "Seg-Sex", -23.600000, -46.700000);

-- Inserir estoque para a Oficina Paulista (id 6)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(6, 1, 'oleo', 40),

(6, 5, 'oleo', 30),

(6, 2, 'filtro', 50),

(6, 6, 'filtro', 35),

(6, 7, 'oleo', 20),

(6, 9, 'filtro', 40);

-- Inserir estoque para a Oficina Ipiranga (id 7)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(7, 3, 'oleo', 35),

(7, 8, 'oleo', 25),

(7, 4, 'filtro', 45),

(7, 7, 'filtro', 30),

(7, 10, 'oleo', 15),

(7, 11, 'filtro', 35);

-- Inserir estoque para a Oficina Tatuapé (id 8)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(8, 2, 'oleo', 45),

(8, 6, 'oleo', 20),

(8, 1, 'filtro', 55),

(8, 5, 'filtro', 25),

(8, 9, 'oleo', 30),

(8, 12, 'filtro', 20);

-- Inserir estoque para a Oficina Mooca (id 9)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(9, 4, 'oleo', 30),

(9, 7, 'oleo', 25),

(9, 3, 'filtro', 40),

(9, 8, 'filtro', 30),

(9, 11, 'oleo', 20),

(9, 13, 'filtro', 25);

-- Inserir estoque para a Oficina Santana (id 10)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(10, 1, 'oleo', 50),

(10, 5, 'oleo', 35),

(10, 2, 'filtro', 60),

(10, 6, 'filtro', 40),

(10, 8, 'oleo', 25),

(10, 10, 'filtro', 45);

-- Inserir estoque para a Oficina Pinheiros (id 11)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(11, 3, 'oleo', 40),

(11, 9, 'oleo', 30),

(11, 4, 'filtro', 50),

(11, 7, 'filtro', 35),

(11, 12, 'oleo', 20),

(11, 14, 'filtro', 30);

-- Inserir estoque para a Oficina Vila Madalena (id 12)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(12, 2, 'oleo', 35),

(12, 10, 'oleo', 25),

(12, 1, 'filtro', 45),

(12, 5, 'filtro', 30),

(12, 13, 'oleo', 15),

(12, 15, 'filtro', 25);

-- Inserir estoque para a Oficina Itaim Bibi (id 13)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(13, 5, 'oleo', 45),

(13, 11, 'oleo', 20),

(13, 6, 'filtro', 55),

(13, 12, 'filtro', 25),

(13, 14, 'oleo', 30),

(13, 16, 'filtro', 20);

-- Inserir estoque para a Oficina Jardins (id 14)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(14, 1, 'oleo', 40),

(14, 7, 'oleo', 30),

(14, 2, 'filtro', 50),

(14, 8, 'filtro', 35),

(14, 15, 'oleo', 20),

(14, 9, 'filtro', 40);

-- Inserir estoque para a Oficina Morumbi (id 15)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(15, 4, 'oleo', 35),

(15, 8, 'oleo', 25),

(15, 3, 'filtro', 45),

(15, 10, 'filtro', 30),

(15, 16, 'oleo', 15),

(15, 11, 'filtro', 35);

-- segunda leva

-- Inserir 20 novos usuários para as oficinas de regiões periféricas

INSERT INTO usuario (nome, email, senha, tipo, telefone, cnpj) VALUES

("Oficina Heliópolis", "adm\_heliopolis@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0741", "10.074.100/0001-01"),

("Oficina Paraisópolis", "adm\_paraisopolis@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0742", "10.074.200/0001-02"),

("Oficina Cidade Tiradentes", "adm\_tiradentes@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0743", "10.074.300/0001-03"),

("Oficina Brasilândia", "adm\_brasilandia@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0744", "10.074.400/0001-04"),

("Oficina Capão Redondo", "adm\_capao@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0745", "10.074.500/0001-05"),

("Oficina Grajaú", "adm\_grajau@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0746", "10.074.600/0001-06"),

("Oficina Jardim Angela", "adm\_jangela@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0747", "10.074.700/0001-07"),

("Oficina Cidade Dutra", "adm\_dutra@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0748", "10.074.800/0001-08"),

("Oficina Vila Prudente", "adm\_vprudente@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0749", "10.074.900/0001-09"),

("Oficina Sapopemba", "adm\_sapopemba@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0750", "10.075.000/0001-10"),

("Oficina Itaquera", "adm\_itaquera@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0751", "10.075.100/0001-11"),

("Oficina Cidade Ademar", "adm\_ademar@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0752", "10.075.200/0001-12"),

("Oficina Pedreira", "adm\_pedreira@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0753", "10.075.300/0001-13"),

("Oficina Jardim São Luís", "adm\_jsluis@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0754", "10.075.400/0001-14"),

("Oficina Parelheiros", "adm\_parelheiros@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0755", "10.075.500/0001-15"),

("Oficina Perus", "adm\_perus@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0756", "10.075.600/0001-16"),

("Oficina Jaraguá", "adm\_jaragua@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0757", "10.075.700/0001-17"),

("Oficina São Miguel", "adm\_smiguel@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0758", "10.075.800/0001-18"),

("Oficina Ermelino Matarazzo", "adm\_ermelino@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0759", "10.075.900/0001-19"),

("Oficina Guaianases", "adm\_guaianases@motul.com", "$2a$10$xJwL5v5z5U6ZU5Z5Z5Z5Zu", "oficina", "(11) 9685-0760", "10.076.000/0001-20");

-- Inserir as 20 novas oficinas em regiões periféricas

INSERT INTO oficina (usuario\_id, nome, cep, endereco, cidade, estado, telefone, horario\_abertura, horario\_fechamento, dias\_funcionamento, lat, lng) VALUES

(23, "Motul Oficina Heliópolis", "04250-000", "Estrada das Lágrimas, 123", "São Paulo", "SP", "(11) 3333-0016", "08:00:00", "18:00:00", "Seg-Sab", -23.610000, -46.625000),

(24, "Motul Oficina Paraisópolis", "05660-000", "Rua Itamotinga, 456", "São Paulo", "SP", "(11) 3333-0017", "08:30:00", "17:30:00", "Seg-Sex", -23.620000, -46.720000),

(25, "Motul Oficina Cidade Tiradentes", "08450-000", "Av. dos Metalúrgicos, 789", "São Paulo", "SP", "(11) 3333-0018", "09:00:00", "18:00:00", "Seg-Sex", -23.580000, -46.410000),

(26, "Motul Oficina Brasilândia", "02850-000", "Av. Brasilândia, 321", "São Paulo", "SP", "(11) 3333-0019", "08:00:00", "17:00:00", "Seg-Sex", -23.470000, -46.680000),

(27, "Motul Oficina Capão Redondo", "05880-000", "Av. Carlos Caldeira, 654", "São Paulo", "SP", "(11) 3333-0020", "08:30:00", "17:30:00", "Seg-Sex", -23.660000, -46.770000),

(28, "Motul Oficina Grajaú", "04840-000", "Av. Dona Belmira Marin, 987", "São Paulo", "SP", "(11) 3333-0021", "09:00:00", "19:00:00", "Seg-Sab", -23.720000, -46.700000),

(29, "Motul Oficina Jardim Angela", "04950-000", "Av. João Dias, 147", "São Paulo", "SP", "(11) 3333-0022", "10:00:00", "20:00:00", "Seg-Sab", -23.690000, -46.760000),

(30, "Motul Oficina Cidade Dutra", "04810-000", "Av. Senador Teotônio Vilela, 258", "São Paulo", "SP", "(11) 3333-0023", "08:00:00", "18:00:00", "Seg-Sex", -23.700000, -46.680000),

(31, "Motul Oficina Vila Prudente", "03130-000", "Rua Siqueira Bueno, 369", "São Paulo", "SP", "(11) 3333-0024", "08:30:00", "17:30:00", "Seg-Sex", -23.580000, -46.580000),

(32, "Motul Oficina Sapopemba", "03980-000", "Av. Sapopemba, 741", "São Paulo", "SP", "(11) 3333-0025", "09:00:00", "18:00:00", "Seg-Sex", -23.540000, -46.480000),

(33, "Motul Oficina Itaquera", "08250-000", "Av. Itaquera, 852", "São Paulo", "SP", "(11) 3333-0026", "08:00:00", "17:00:00", "Seg-Sex", -23.540000, -46.470000),

(34, "Motul Oficina Cidade Ademar", "04450-000", "Av. Cupecê, 963", "São Paulo", "SP", "(11) 3333-0027", "08:30:00", "17:30:00", "Seg-Sex", -23.690000, -46.700000),

(35, "Motul Oficina Pedreira", "04455-000", "Av. Robert Kennedy, 159", "São Paulo", "SP", "(11) 3333-0028", "09:00:00", "18:00:00", "Seg-Sex", -23.730000, -46.690000),

(36, "Motul Oficina Jardim São Luís", "05840-000", "Av. Washington Luís, 357", "São Paulo", "SP", "(11) 3333-0029", "08:00:00", "18:00:00", "Seg-Sab", -23.650000, -46.750000),

(37, "Motul Oficina Parelheiros", "04890-000", "Estrada de Parelheiros, 486", "São Paulo", "SP", "(11) 3333-0030", "08:30:00", "17:30:00", "Seg-Sex", -23.820000, -46.700000),

(38, "Motul Oficina Perus", "05200-000", "Av. Raimundo Pereira de Magalhães, 753", "São Paulo", "SP", "(11) 3333-0031", "09:00:00", "18:00:00", "Seg-Sex", -23.400000, -46.760000),

(39, "Motul Oficina Jaraguá", "05180-000", "Av. Elísio Teixeira Leite, 264", "São Paulo", "SP", "(11) 3333-0032", "08:00:00", "17:00:00", "Seg-Sex", -23.460000, -46.740000),

(40, "Motul Oficina São Miguel", "08010-000", "Av. Nordestina, 975", "São Paulo", "SP", "(11) 3333-0033", "08:30:00", "17:30:00", "Seg-Sex", -23.500000, -46.440000),

(41, "Motul Oficina Ermelino Matarazzo", "03810-000", "Av. Paranaguá, 681", "São Paulo", "SP", "(11) 3333-0034", "09:00:00", "18:00:00", "Seg-Sex", -23.490000, -46.480000),

(42, "Motul Oficina Guaianases", "08440-000", "Av. Dr. Assis Ribeiro, 294", "São Paulo", "SP", "(11) 3333-0035", "08:00:00", "17:00:00", "Seg-Sex", -23.540000, -46.410000);

-- Inserir estoque para as 20 novas oficinas (cada uma com 6 produtos)

-- Oficina Heliópolis (id 16)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(16, 2, 'oleo', 35), (16, 4, 'oleo', 40), (16, 1, 'filtro', 50),

(16, 3, 'filtro', 30), (16, 5, 'oleo', 25), (16, 6, 'filtro', 45);

-- Oficina Paraisópolis (id 17)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(17, 1, 'oleo', 40), (17, 3, 'oleo', 35), (17, 2, 'filtro', 45),

(17, 4, 'filtro', 30), (17, 6, 'oleo', 20), (17, 5, 'filtro', 40);

-- Oficina Cidade Tiradentes (id 18)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(18, 4, 'oleo', 30), (18, 6, 'oleo', 25), (18, 3, 'filtro', 40),

(18, 5, 'filtro', 35), (18, 7, 'oleo', 20), (18, 8, 'filtro', 30);

-- Oficina Brasilândia (id 19)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(19, 3, 'oleo', 35), (19, 5, 'oleo', 30), (19, 4, 'filtro', 45),

(19, 6, 'filtro', 40), (19, 8, 'oleo', 25), (19, 7, 'filtro', 35);

-- Oficina Capão Redondo (id 20)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(20, 5, 'oleo', 40), (20, 7, 'oleo', 35), (20, 6, 'filtro', 50),

(20, 8, 'filtro', 30), (20, 9, 'oleo', 20), (20, 10, 'filtro', 40);

-- Oficina Grajaú (id 21)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(21, 6, 'oleo', 35), (21, 8, 'oleo', 30), (21, 7, 'filtro', 45),

(21, 9, 'filtro', 40), (21, 10, 'oleo', 25), (21, 11, 'filtro', 35);

-- Oficina Jardim Angela (id 22)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(22, 7, 'oleo', 40), (22, 9, 'oleo', 35), (22, 8, 'filtro', 50),

(22, 10, 'filtro', 30), (22, 11, 'oleo', 20), (22, 12, 'filtro', 40);

-- Oficina Cidade Dutra (id 23)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(23, 8, 'oleo', 35), (23, 10, 'oleo', 30), (23, 9, 'filtro', 45),

(23, 11, 'filtro', 40), (23, 12, 'oleo', 25), (23, 13, 'filtro', 35);

-- Oficina Vila Prudente (id 24)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(24, 9, 'oleo', 40), (24, 11, 'oleo', 35), (24, 10, 'filtro', 50),

(24, 12, 'filtro', 30), (24, 13, 'oleo', 20), (24, 14, 'filtro', 40);

-- Oficina Sapopemba (id 25)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(25, 10, 'oleo', 35), (25, 12, 'oleo', 30), (25, 11, 'filtro', 45),

(25, 13, 'filtro', 40), (25, 14, 'oleo', 25), (25, 15, 'filtro', 35);

-- Oficina Itaquera (id 26)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(26, 11, 'oleo', 40), (26, 13, 'oleo', 35), (26, 12, 'filtro', 50),

(26, 14, 'filtro', 30), (26, 15, 'oleo', 20), (26, 16, 'filtro', 40);

-- Oficina Cidade Ademar (id 27)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(27, 12, 'oleo', 35), (27, 14, 'oleo', 30), (27, 13, 'filtro', 45),

(27, 15, 'filtro', 40), (27, 16, 'oleo', 25), (27, 1, 'filtro', 35);

-- Oficina Pedreira (id 28)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(28, 13, 'oleo', 40), (28, 15, 'oleo', 35), (28, 14, 'filtro', 50),

(28, 16, 'filtro', 30), (28, 1, 'oleo', 20), (28, 2, 'filtro', 40);

-- Oficina Jardim São Luís (id 29)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(29, 14, 'oleo', 35), (29, 16, 'oleo', 30), (29, 15, 'filtro', 45),

(29, 1, 'filtro', 40), (29, 2, 'oleo', 25), (29, 3, 'filtro', 35);

-- Oficina Parelheiros (id 30)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(30, 15, 'oleo', 40), (30, 1, 'oleo', 35), (30, 16, 'filtro', 50),

(30, 2, 'filtro', 30), (30, 3, 'oleo', 20), (30, 4, 'filtro', 40);

-- Oficina Perus (id 31)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(31, 16, 'oleo', 35), (31, 2, 'oleo', 30), (31, 1, 'filtro', 45),

(31, 3, 'filtro', 40), (31, 4, 'oleo', 25), (31, 5, 'filtro', 35);

-- Oficina Jaraguá (id 32)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(32, 1, 'oleo', 40), (32, 3, 'oleo', 35), (32, 2, 'filtro', 50),

(32, 4, 'filtro', 30), (32, 5, 'oleo', 20), (32, 6, 'filtro', 40);

-- Oficina São Miguel (id 33)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(33, 2, 'oleo', 35), (33, 4, 'oleo', 30), (33, 3, 'filtro', 45),

(33, 5, 'filtro', 40), (33, 6, 'oleo', 25), (33, 7, 'filtro', 35);

-- Oficina Ermelino Matarazzo (id 34)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(34, 3, 'oleo', 40), (34, 5, 'oleo', 35), (34, 4, 'filtro', 50),

(34, 6, 'filtro', 30), (34, 7, 'oleo', 20), (34, 8, 'filtro', 40);

-- Oficina Guaianases (id 35)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade) VALUES

(35, 4, 'oleo', 35), (35, 6, 'oleo', 30), (35, 5, 'filtro', 45),

(35, 7, 'filtro', 40), (35, 8, 'oleo', 25), (35, 9, 'filtro', 35);

-- terceiraaaaa

-- Inserir mais óleos

INSERT INTO produto\_oleo (nome, tipo, viscosidade, especificacao, marca, preco) VALUES

("Óleo Sintético 0W-40", "carro", "0W-40", "API SP/ACEA C3", "Motul", 130.00),

("Óleo Semissintético 15W-40", "carro", "15W-40", "API CH-4", "Shell", 70.00),

("Óleo Mineral 10W-30", "carro", "10W-30", "API SL", "Petrobras", 50.00),

("Óleo Sintético 5W-30 Moto", "moto", "5W-30", "JASO MA2", "Mobil", 85.00),

("Óleo Semissintético 20W-50 Moto", "moto", "20W-50", "JASO MA", "Castrol", 65.00),

("Óleo Sintético 5W-40 Diesel", "carro", "5W-40", "API CK-4", "Total", 110.00),

("Óleo Sintético 0W-16", "carro", "0W-16", "API SP", "Idemitsu", 140.00),

("Óleo Semissintético 10W-30 Moto", "moto", "10W-30", "JASO MB", "Yamalube", 75.00),

("Óleo Mineral 15W-40", "carro", "15W-40", "API SJ", "Ipiranga", 55.00),

("Óleo Sintético 5W-50", "carro", "5W-50", "API SN", "Liqui Moly", 120.00);

-- Inserir mais filtros

INSERT INTO produto\_filtro (nome, tipo, compatibilidade\_modelo, preco) VALUES

("Filtro de Óleo Específico Fiat", "carro", "Fiat Palio, Uno, Siena, Strada", 32.00),

("Filtro de Óleo Específico Volkswagen", "carro", "VW Gol, Voyage, Saveiro, Polo", 34.00),

("Filtro de Óleo Específico Honda", "carro", "Honda Civic, Fit, City, HR-V", 36.00),

("Filtro de Óleo Específico Toyota", "carro", "Toyota Corolla, Etios, Yaris, Hilux", 38.00),

("Filtro de Ar Específico Fiat", "carro", "Fiat Palio, Uno, Siena, Strada", 48.00),

("Filtro de Ar Específico Volkswagen", "carro", "VW Gol, Voyage, Saveiro, Polo", 52.00),

("Filtro de Ar Específico Honda", "carro", "Honda Civic, Fit, City, HR-V", 56.00),

("Filtro de Ar Específico Toyota", "carro", "Toyota Corolla, Etios, Yaris, Hilux", 58.00),

("Filtro de Combustível Universal Carro", "carro", "Compatível com maioria dos veículos", 22.00),

("Filtro de Cabine com Carvão Ativado", "carro", "Compatível com maioria dos veículos", 65.00),

("Filtro de Óleo Racing", "carro", "Para veículos de alta performance", 85.00),

("Filtro de Ar Esportivo", "carro", "Para veículos esportivos", 90.00),

("Filtro de Óleo para Moto Honda", "moto", "Honda CG, CB, Biz, NXR", 28.00),

("Filtro de Óleo para Moto Yamaha", "moto", "Yamaha Fazer, Factor, Lander", 30.00),

("Filtro de Ar para Moto Universal", "moto", "Compatível com maioria das motos", 35.00);

-- Inserir mais marcas de veículos

INSERT INTO marca (nome) VALUES

("Kia"), ("Citroën"), ("Mitsubishi"), ("Volvo"), ("Land Rover"),

("JAC"), ("Chery"), ("RAM"), ("Foton"), ("Iveco"), ("Scania");

-- Inserir mais modelos de carros

INSERT INTO modelo (nome, marca\_id, tipo) VALUES

("Sportage", 20, "carro"), ("Sorento", 20, "carro"), ("Cerato", 20, "carro"),

("C4 Cactus", 21, "carro"), ("C4 Lounge", 21, "carro"), ("Aircross", 21, "carro"),

("L200", 22, "carro"), ("Outlander", 22, "carro"), ("Pajero", 22, "carro"),

("XC60", 23, "carro"), ("XC90", 23, "carro"), ("S60", 23, "carro"),

("Discovery", 24, "carro"), ("Range Rover Evoque", 24, "carro"), ("Defender", 24, "carro"),

("J3", 25, "carro"), ("J5", 25, "carro"), ("T40", 25, "carro"),

("Tiggo", 26, "carro"), ("QQ", 26, "carro"), ("Arrizo", 26, "carro"),

("1500", 27, "carro"), ("2500", 27, "carro"), ("3500", 27, "carro"),

("Aumark", 28, "carro"), ("Tunland", 28, "carro"), ("View", 28, "carro"),

("Daily", 29, "carro"), ("Trakker", 29, "carro"), ("Stralis", 29, "carro"),

("P310", 30, "carro"), ("P360", 30, "carro"), ("P410", 30, "carro");

-- Inserir mais modelos de motos

INSERT INTO modelo (nome, marca\_id, tipo) VALUES

("XRE 300", 6, "moto"), ("CB 1000R", 6, "moto"), ("NC 750X", 6, "moto"),

("MT-07", 12, "moto"), ("MT-09", 12, "moto"), ("Ténéré 700", 12, "moto"),

("GSX-R1000", 13, "moto"), ("V-Strom 1000", 13, "moto"), ("Boulevard M109R", 13, "moto"),

("Ninja 650", 14, "moto"), ("Z650", 14, "moto"), ("Vulcan S", 14, "moto"),

("Street Glide", 15, "moto"), ("Road Glide", 15, "moto"), ("Sportster Iron 1200", 15, "moto"),

("S1000RR", 16, "moto"), ("R1250GS", 16, "moto"), ("F850GS", 16, "moto");

-- Inserir anos para os novos modelos

INSERT INTO modelo\_ano (modelo\_id, ano) VALUES

-- Kia

(67, 2020), (67, 2021), (67, 2022), (67, 2023), (67, 2024),

(68, 2020), (68, 2021), (68, 2022), (68, 2023), (68, 2024),

(69, 2020), (69, 2021), (69, 2022), (69, 2023), (69, 2024),

-- Citroën

(70, 2020), (70, 2021), (70, 2022), (70, 2023), (70, 2024),

(71, 2020), (71, 2021), (71, 2022), (71, 2023), (71, 2024),

(72, 2020), (72, 2021), (72, 2022), (72, 2023), (72, 2024),

-- Mitsubishi

(73, 2020), (73, 2021), (73, 2022), (73, 2023), (73, 2024),

(74, 2020), (74, 2021), (74, 2022), (74, 2023), (74, 2024),

(75, 2020), (75, 2021), (75, 2022), (75, 2023), (75, 2024),

-- Volvo

(76, 2020), (76, 2021), (76, 2022), (76, 2023), (76, 2024),

(77, 2020), (77, 2021), (77, 2022), (77, 2023), (77, 2024),

(78, 2020), (78, 2021), (78, 2022), (78, 2023), (78, 2024),

-- Land Rover

(79, 2020), (79, 2021), (79, 2022), (79, 2023), (79, 2024),

(80, 2020), (80, 2021), (80, 2022), (80, 2023), (80, 2024),

(81, 2020), (81, 2021), (81, 2022), (81, 2023), (81, 2024),

-- JAC

(82, 2020), (82, 2021), (82, 2022), (82, 2023), (82, 2024),

(83, 2020), (83, 2021), (83, 2022), (83, 2023), (83, 2024),

(84, 2020), (84, 2021), (84, 2022), (84, 2023), (84, 2024),

-- Chery

(85, 2020), (85, 2021), (85, 2022), (85, 2023), (85, 2024),

(86, 2020), (86, 2021), (86, 2022), (86, 2023), (86, 2024),

(87, 2020), (87, 2021), (87, 2022), (87, 2023), (87, 2024),

-- RAM

(88, 2020), (88, 2021), (88, 2022), (88, 2023), (88, 2024),

(89, 2020), (89, 2021), (89, 2022), (89, 2023), (89, 2024),

(90, 2020), (90, 2021), (90, 2022), (90, 2023), (90, 2024),

-- Foton

(91, 2020), (91, 2021), (91, 2022), (91, 2023), (91, 2024),

(92, 2020), (92, 2021), (92, 2022), (92, 2023), (92, 2024),

(93, 2020), (93, 2021), (93, 2022), (93, 2023), (93, 2024),

-- Iveco

(94, 2020), (94, 2021), (94, 2022), (94, 2023), (94, 2024),

(95, 2020), (95, 2021), (95, 2022), (95, 2023), (95, 2024),

(96, 2020), (96, 2021), (96, 2022), (96, 2023), (96, 2024),

-- Scania

(97, 2020), (97, 2021), (97, 2022), (97, 2023), (97, 2024),

(98, 2020), (98, 2021), (98, 2022), (98, 2023), (98, 2024),

(99, 2020), (99, 2021), (99, 2022), (99, 2023), (99, 2024),

-- Novas motos Honda

(100, 2020), (100, 2021), (100, 2022), (100, 2023), (100, 2024),

(101, 2020), (101, 2021), (101, 2022), (101, 2023), (101, 2024),

(102, 2020), (102, 2021), (102, 2022), (102, 2023), (102, 2024),

-- Novas motos Yamaha

(103, 2020), (103, 2021), (103, 2022), (103, 2023), (103, 2024),

(104, 2020), (104, 2021), (104, 2022), (104, 2023), (104, 2024),

(105, 2020), (105, 2021), (105, 2022), (105, 2023), (105, 2024),

-- Novas motos Suzuki

(106, 2020), (106, 2021), (106, 2022), (106, 2023), (106, 2024),

(107, 2020), (107, 2021), (107, 2022), (107, 2023), (107, 2024),

(108, 2020), (108, 2021), (108, 2022), (108, 2023), (108, 2024),

-- Novas motos Kawasaki

(109, 2020), (109, 2021), (109, 2022), (109, 2023), (109, 2024),

(110, 2020), (110, 2021), (110, 2022), (110, 2023), (110, 2024),

(111, 2020), (111, 2021), (111, 2022), (111, 2023), (111, 2024),

-- Novas motos Harley-Davidson

(112, 2020), (112, 2021), (112, 2022), (112, 2023), (112, 2024),

(113, 2020), (113, 2021), (113, 2022), (113, 2023), (113, 2024),

(114, 2020), (114, 2021), (114, 2022), (114, 2023), (114, 2024),

-- Novas motos BMW

(115, 2020), (115, 2021), (115, 2022), (115, 2023), (115, 2024),

(116, 2020), (116, 2021), (116, 2022), (116, 2023), (116, 2024),

(117, 2020), (117, 2021), (117, 2022), (117, 2023), (117, 2024);

-- Adicionar recomendações para os novos modelos

-- Kia (marca\_id = 20) - usando óleo 1 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 1, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 20 AND m.tipo = "carro";

-- Citroën (marca\_id = 21) - usando óleo 5 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 5, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 21 AND m.tipo = "carro";

-- Mitsubishi (marca\_id = 22) - usando óleo 1 e filtro 2

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 1, 2 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 22 AND m.tipo = "carro";

-- Volvo (marca\_id = 23) - usando óleo 11 e filtro 12

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 11, 12 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 23 AND m.tipo = "carro";

-- Land Rover (marca\_id = 24) - usando óleo 11 e filtro 12

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 11, 12 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 24 AND m.tipo = "carro";

-- JAC (marca\_id = 25) - usando óleo 2 e filtro 1

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 2, 1 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 25 AND m.tipo = "carro";

-- Chery (marca\_id = 26) - usando óleo 2 e filtro 1

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 2, 1 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 26 AND m.tipo = "carro";

-- RAM (marca\_id = 27) - usando óleo 6 e filtro 14

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 6, 14 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 27 AND m.tipo = "carro";

-- Foton (marca\_id = 28) - usando óleo 6 e filtro 14

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 6, 14 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 28 AND m.tipo = "carro";

-- Iveco (marca\_id = 29) - usando óleo 6 e filtro 14

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 6, 14 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 29 AND m.tipo = "carro";

-- Scania (marca\_id = 30) - usando óleo 6 e filtro 14

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 6, 14 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 30 AND m.tipo = "carro";

-- Novas motos Honda (marca\_id = 6) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 6 AND m.tipo = "moto" AND m.id >= 100;

-- Novas motos Yamaha (marca\_id = 12) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 12 AND m.tipo = "moto" AND m.id >= 103;

-- Novas motos Suzuki (marca\_id = 13) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 13 AND m.tipo = "moto" AND m.id >= 106;

-- Novas motos Kawasaki (marca\_id = 14) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 14 AND m.tipo = "moto" AND m.id >= 109;

-- Novas motos Harley-Davidson (marca\_id = 15) - usando óleo 6 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 6, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 15 AND m.tipo = "moto" AND m.id >= 112;

-- Novas motos BMW (marca\_id = 16) - usando óleo 3 e filtro 4

INSERT INTO recomendacao (modelo\_ano\_id, oleo\_id, filtro\_id)

SELECT ma.id, 3, 4 FROM modelo\_ano ma

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.marca\_id = 16 AND m.tipo = "moto" AND m.id >= 115;

-- Distribuir mais produtos para todas as oficinas existentes

-- Para cada oficina, adicionar 4 produtos adicionais (2 óleos e 2 filtros)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 17, 'oleo', FLOOR(RAND() \* 30) + 20 FROM oficina;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 18, 'oleo', FLOOR(RAND() \* 25) + 15 FROM oficina;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 19, 'filtro', FLOOR(RAND() \* 40) + 30 FROM oficina;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 20, 'filtro', FLOOR(RAND() \* 35) + 25 FROM oficina;

-- Adicionar produtos específicos para regiões periféricas (óleos mais acessíveis)

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 3, 'oleo', FLOOR(RAND() \* 50) + 40 FROM oficina WHERE id BETWEEN 16 AND 35;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 4, 'oleo', FLOOR(RAND() \* 45) + 35 FROM oficina WHERE id BETWEEN 16 AND 35;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 9, 'filtro', FLOOR(RAND() \* 60) + 50 FROM oficina WHERE id BETWEEN 16 AND 35;

-- Adicionar produtos premium para regiões centrais

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 11, 'oleo', FLOOR(RAND() \* 20) + 15 FROM oficina WHERE id BETWEEN 1 AND 15;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 12, 'oleo', FLOOR(RAND() \* 25) + 20 FROM oficina WHERE id BETWEEN 1 AND 15;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 16, 'filtro', FLOOR(RAND() \* 30) + 25 FROM oficina WHERE id BETWEEN 1 AND 15;

-- Adicionar produtos para motos em todas as oficinas

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 3, 'oleo', FLOOR(RAND() \* 30) + 20 FROM oficina;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 6, 'oleo', FLOOR(RAND() \* 25) + 15 FROM oficina;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 4, 'filtro', FLOOR(RAND() \* 40) + 30 FROM oficina;

INSERT INTO estoque (oficina\_id, produto\_id, tipo\_produto, quantidade)

SELECT id, 8, 'filtro', FLOOR(RAND() \* 35) + 25 FROM oficina;

SELECT \* FROM oficina;

SELECT id, nome, tipo, preco FROM produto\_filtro;

SELECT id, nome, tipo, preco FROM produto\_oleo;

-- Verificar filtros com preço zero ou nulo

SELECT id, nome, tipo, preco FROM produto\_filtro WHERE preco IS NULL OR preco = 0;

-- Verificar qual filtro está sendo retornado para motos

SELECT r.id, r.filtro\_id, f.nome, f.preco, f.compatibilidade\_modelo

FROM recomendacao r

JOIN produto\_filtro f ON r.filtro\_id = f.id

JOIN modelo\_ano ma ON r.modelo\_ano\_id = ma.id

JOIN modelo m ON ma.modelo\_id = m.id

WHERE m.tipo = 'moto'

LIMIT 5;

-- Teste direto para ver se encontra a coluna

SELECT COUNT(\*)

FROM information\_schema.columns

WHERE table\_name = 'produto\_filtro'

AND table\_schema = DATABASE()

AND column\_name = 'preco';

ALTER TABLE usuario

ADD COLUMN reset\_token VARCHAR(255) NULL,

ADD COLUMN reset\_token\_expiry BIGINT NULL;

USE Oil;

SELECT \* FROM agendamento\_simples ORDER BY id DESC;

UPDATE agendamento\_simples

SET servicos = REPLACE(REPLACE(REPLACE(servicos, '[', ''), ']', ''), '"', '')

WHERE (servicos LIKE '%[%' OR servicos LIKE '%]%' OR servicos LIKE '%"%')

AND id > 0;

ALTER TABLE agendamento\_simples

ADD COLUMN usuario\_id INT NULL AFTER cliente\_email;

-- Adicionar índice para melhor performance

CREATE INDEX idx\_agendamento\_usuario ON agendamento\_simples(usuario\_id);

ALTER TABLE agendamento\_simples

ADD COLUMN status ENUM('pendente', 'confirmado', 'concluido', 'cancelado', 'fora\_prazo') DEFAULT 'pendente',

ADD COLUMN data\_conclusao DATETIME NULL,

ADD COLUMN motivo\_cancelamento TEXT NULL;

# Arquivo: package-lock.json

{

"name": "oilsmart",

"version": "1.0.0",

"lockfileVersion": 3,

"requires": true,

"packages": {

"": {

"name": "oilsmart",

"version": "1.0.0",

"license": "ISC",

"dependencies": {

"axios": "^1.11.0",

"bcrypt": "^6.0.0",

"bcryptjs": "^3.0.2",

"cors": "^2.8.5",

"dotenv": "^17.2.1",

"ejs": "^3.1.9",

"express": "^4.21.2",

"jsonwebtoken": "^9.0.2",

"mysql2": "^3.14.1",

"nodemailer": "^7.0.5"

}

},

"node\_modules/accepts": {

"version": "1.3.8",

"resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",

"integrity": "sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACSaeXEQ+NHcVF7rONl6qcaxV3Uuemwawk+7+SJLw==",

"license": "MIT",

"dependencies": {

"mime-types": "~2.1.34",

"negotiator": "0.6.3"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/ansi-styles": {

"version": "4.3.0",

"resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.3.0.tgz",

"integrity": "sha512-zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+IDohdPlFp222wVALIheZJQSEg==",

"license": "MIT",

"dependencies": {

"color-convert": "^2.0.1"

},

"engines": {

"node": ">=8"

},

"funding": {

"url": "https://github.com/chalk/ansi-styles?sponsor=1"

}

},

"node\_modules/array-flatten": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",

"integrity": "sha512-PCVAQswWemu6UdxsDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx8nNlM++NqRcK6CxxpUafjmhIdKiHibqg==",

"license": "MIT"

},

"node\_modules/async": {

"version": "3.2.6",

"resolved": "https://registry.npmjs.org/async/-/async-3.2.6.tgz",

"integrity": "sha512-htCUDlxyyCLMgaM3xXg0C0LW2xqfuQ6p05pCEIsXuyQ+a1koYKTuBMzRNwmybfLgvJDMd0r1LTn4+E0Ti6C2AA==",

"license": "MIT"

},

"node\_modules/asynckit": {

"version": "0.4.0",

"resolved": "https://registry.npmjs.org/asynckit/-/asynckit-0.4.0.tgz",

"integrity": "sha512-Oei9OH4tRh0YqU3GxhX79dM/mwVgvbZJaSNaRk+bshkj0S5cfHcgYakreBjrHwatXKbz+IoIdYLxrKim2MjW0Q==",

"license": "MIT"

},

"node\_modules/aws-ssl-profiles": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/aws-ssl-profiles/-/aws-ssl-profiles-1.1.2.tgz",

"integrity": "sha512-NZKeq9AfyQvEeNlN0zSYAaWrmBffJh3IELMZfRpJVWgrpEbtEpnjvzqBPf+mxoI287JohRDoa+/nsfqqiZmF6g==",

"license": "MIT",

"engines": {

"node": ">= 6.0.0"

}

},

"node\_modules/axios": {

"version": "1.12.2",

"resolved": "https://registry.npmjs.org/axios/-/axios-1.12.2.tgz",

"integrity": "sha512-vMJzPewAlRyOgxV2dU0Cuz2O8zzzx9VYtbJOaBgXFeLc4IV/Eg50n4LowmehOOR61S8ZMpc2K5Sa7g6A4jfkUw==",

"license": "MIT",

"dependencies": {

"follow-redirects": "^1.15.6",

"form-data": "^4.0.4",

"proxy-from-env": "^1.1.0"

}

},

"node\_modules/balanced-match": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/balanced-match/-/balanced-match-1.0.2.tgz",

"integrity": "sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJhYU5Pw==",

"license": "MIT"

},

"node\_modules/bcrypt": {

"version": "6.0.0",

"resolved": "https://registry.npmjs.org/bcrypt/-/bcrypt-6.0.0.tgz",

"integrity": "sha512-cU8v/EGSrnH+HnxV2z0J7/blxH8gq7Xh2JFT6Aroax7UohdmiJJlxApMxtKfuI7z68NvvVcmR78k2LbT6efhRg==",

"hasInstallScript": true,

"license": "MIT",

"dependencies": {

"node-addon-api": "^8.3.0",

"node-gyp-build": "^4.8.4"

},

"engines": {

"node": ">= 18"

}

},

"node\_modules/bcryptjs": {

"version": "3.0.2",

"resolved": "https://registry.npmjs.org/bcryptjs/-/bcryptjs-3.0.2.tgz",

"integrity": "sha512-k38b3XOZKv60C4E2hVsXTolJWfkGRMbILBIe2IBITXciy5bOsTKot5kDrf3ZfufQtQOUN5mXceUEpU1rTl9Uog==",

"license": "BSD-3-Clause",

"bin": {

"bcrypt": "bin/bcrypt"

}

},

"node\_modules/body-parser": {

"version": "1.20.3",

"resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.20.3.tgz",

"integrity": "sha512-7rAxByjUMqQ3/bHJy7D6OGXvx/MMc4IqBn/X0fcM1QUcAItpZrBEYhWGem+tzXH90c+G01ypMcYJBO9Y30203g==",

"license": "MIT",

"dependencies": {

"bytes": "3.1.2",

"content-type": "~1.0.5",

"debug": "2.6.9",

"depd": "2.0.0",

"destroy": "1.2.0",

"http-errors": "2.0.0",

"iconv-lite": "0.4.24",

"on-finished": "2.4.1",

"qs": "6.13.0",

"raw-body": "2.5.2",

"type-is": "~1.6.18",

"unpipe": "1.0.0"

},

"engines": {

"node": ">= 0.8",

"npm": "1.2.8000 || >= 1.4.16"

}

},

"node\_modules/brace-expansion": {

"version": "1.1.12",

"resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-1.1.12.tgz",

"integrity": "sha512-9T9UjW3r0UW5c1Q7GTwllptXwhvYmEzFhzMfZ9H7FQWt+uZePjZPjBP/W1ZEyZ1twGWom5/56TF4lPcqjnDHcg==",

"license": "MIT",

"dependencies": {

"balanced-match": "^1.0.0",

"concat-map": "0.0.1"

}

},

"node\_modules/buffer-equal-constant-time": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/buffer-equal-constant-time/-/buffer-equal-constant-time-1.0.1.tgz",

"integrity": "sha512-zRpUiDwd/xk6ADqPMATG8vc9VPrkck7T07OIx0gnjmJAnHnTVXNQG3vfvWNuiZIkwu9KrKdA1iJKfsfTVxE6NA==",

"license": "BSD-3-Clause"

},

"node\_modules/bytes": {

"version": "3.1.2",

"resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.2.tgz",

"integrity": "sha512-/Nf7TyzTx6S3yRJObOAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8P8zGkNXStjTSi6fzO6F0pBdcYbEg==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/call-bind-apply-helpers": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/call-bind-apply-helpers/-/call-bind-apply-helpers-1.0.2.tgz",

"integrity": "sha512-Sp1ablJ0ivDkSzjcaJdxEunN5/XvksFJ2sMBFfq6x0ryhQV/2b/KwFe21cMpmHtPOSij8K99/wSfoEuTObmuMQ==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"function-bind": "^1.1.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/call-bound": {

"version": "1.0.4",

"resolved": "https://registry.npmjs.org/call-bound/-/call-bound-1.0.4.tgz",

"integrity": "sha512-+ys997U96po4Kx/ABpBCqhA9EuxJaQWDQg7295H4hBphv3IZg0boBKuwYpt4YXp6MZ5AmZQnU/tyMTlRpaSejg==",

"license": "MIT",

"dependencies": {

"call-bind-apply-helpers": "^1.0.2",

"get-intrinsic": "^1.3.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/chalk": {

"version": "4.1.2",

"resolved": "https://registry.npmjs.org/chalk/-/chalk-4.1.2.tgz",

"integrity": "sha512-oKnbhFyRIXpUuez8iBMmyEa4nbj4IOQyuhc/wy9kY7/WVPcwIO9VA668Pu8RkO7+0G76SLROeyw9CpQ061i4mA==",

"license": "MIT",

"dependencies": {

"ansi-styles": "^4.1.0",

"supports-color": "^7.1.0"

},

"engines": {

"node": ">=10"

},

"funding": {

"url": "https://github.com/chalk/chalk?sponsor=1"

}

},

"node\_modules/color-convert": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz",

"integrity": "sha512-RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+CRDpfK/U584fMg38ZHCaElKQ==",

"license": "MIT",

"dependencies": {

"color-name": "~1.1.4"

},

"engines": {

"node": ">=7.0.0"

}

},

"node\_modules/color-name": {

"version": "1.1.4",

"resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.4.tgz",

"integrity": "sha512-dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEoYo5q4Fx7dY9OgQGXgAsQA==",

"license": "MIT"

},

"node\_modules/combined-stream": {

"version": "1.0.8",

"resolved": "https://registry.npmjs.org/combined-stream/-/combined-stream-1.0.8.tgz",

"integrity": "sha512-FQN4MRfuJeHf7cBbBMJFXhKSDq+2kAArBlmRBvcvFE5BB1HZKXtSFASDhdlz9zOYwxh8lDdnvmMOe/+5cdoEdg==",

"license": "MIT",

"dependencies": {

"delayed-stream": "~1.0.0"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/concat-map": {

"version": "0.0.1",

"resolved": "https://registry.npmjs.org/concat-map/-/concat-map-0.0.1.tgz",

"integrity": "sha512-/Srv4dswyQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRljaA7EXHaXZ8wsHdodFvbkhKmqg==",

"license": "MIT"

},

"node\_modules/content-disposition": {

"version": "0.5.4",

"resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.4.tgz",

"integrity": "sha512-FveZTNuGw04cxlAiWbzi6zTAL/lhehaWbTtgluJh4/E95DqMwTmha3KZN1aAWA8cFIhHzMZUvLevkw5Rqk+tSQ==",

"license": "MIT",

"dependencies": {

"safe-buffer": "5.2.1"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/content-type": {

"version": "1.0.5",

"resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.5.tgz",

"integrity": "sha512-nTjqfcBFEipKdXCv4YDQWCfmcLZKm81ldF0pAopTvyrFGVbcR6P/VAAd5G7N+0tTr8QqiU0tFadD6FK4NtJwOA==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/cookie": {

"version": "0.7.1",

"resolved": "https://registry.npmjs.org/cookie/-/cookie-0.7.1.tgz",

"integrity": "sha512-6DnInpx7SJ2AK3+CTUE/ZM0vWTUboZCegxhC2xiIydHR9jNuTAASBrfEpHhiGOZw/nX51bHt6YQl8jsGo4y/0w==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/cookie-signature": {

"version": "1.0.6",

"resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.0.6.tgz",

"integrity": "sha512-QADzlaHc8icV8I7vbaJXJwod9HWYp8uCqf1xa4OfNu1T7JVxQIrUgOWtHdNDtPiywmFbiS12VjotIXLrKM3orQ==",

"license": "MIT"

},

"node\_modules/cors": {

"version": "2.8.5",

"resolved": "https://registry.npmjs.org/cors/-/cors-2.8.5.tgz",

"integrity": "sha512-KIHbLJqu73RGr/hnbrO9uBeixNGuvSQjul/jdFvS/KFSIH1hWVd1ng7zOHx+YrEfInLG7q4n6GHQ9cDtxv/P6g==",

"license": "MIT",

"dependencies": {

"object-assign": "^4",

"vary": "^1"

},

"engines": {

"node": ">= 0.10"

}

},

"node\_modules/debug": {

"version": "2.6.9",

"resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",

"integrity": "sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsHMA==",

"license": "MIT",

"dependencies": {

"ms": "2.0.0"

}

},

"node\_modules/delayed-stream": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/delayed-stream/-/delayed-stream-1.0.0.tgz",

"integrity": "sha512-ZySD7Nf91aLB0RxL4KGrKHBXl7Eds1DAmEdcoVawXnLD7SDhpNgtuII2aAkg7a7QS41jxPSZ17p4VdGnMHk3MQ==",

"license": "MIT",

"engines": {

"node": ">=0.4.0"

}

},

"node\_modules/denque": {

"version": "2.1.0",

"resolved": "https://registry.npmjs.org/denque/-/denque-2.1.0.tgz",

"integrity": "sha512-HVQE3AAb/pxF8fQAoiqpvg9i3evqug3hoiwakOyZAwJm+6vZehbkYXZ0l4JxS+I3QxM97v5aaRNhj8v5oBhekw==",

"license": "Apache-2.0",

"engines": {

"node": ">=0.10"

}

},

"node\_modules/depd": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/depd/-/depd-2.0.0.tgz",

"integrity": "sha512-g7nH6P6dyDioJogAAGprGpCtVImJhpPk/roCzdb3fIh61/s/nPsfR6onyMwkCAR/OlC3yBC0lESvUoQEAssIrw==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/destroy": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/destroy/-/destroy-1.2.0.tgz",

"integrity": "sha512-2sJGJTaXIIaR1w4iJSNoN0hnMY7Gpc/n8D4qSCJw8QqFWXf7cuAgnEHxBpweaVcPevC2l3KpjYCx3NypQQgaJg==",

"license": "MIT",

"engines": {

"node": ">= 0.8",

"npm": "1.2.8000 || >= 1.4.16"

}

},

"node\_modules/dotenv": {

"version": "17.2.1",

"resolved": "https://registry.npmjs.org/dotenv/-/dotenv-17.2.1.tgz",

"integrity": "sha512-kQhDYKZecqnM0fCnzI5eIv5L4cAe/iRI+HqMbO/hbRdTAeXDG+M9FjipUxNfbARuEg4iHIbhnhs78BCHNbSxEQ==",

"license": "BSD-2-Clause",

"engines": {

"node": ">=12"

},

"funding": {

"url": "https://dotenvx.com"

}

},

"node\_modules/dunder-proto": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/dunder-proto/-/dunder-proto-1.0.1.tgz",

"integrity": "sha512-KIN/nDJBQRcXw0MLVhZE9iQHmG68qAVIBg9CqmUYjmQIhgij9U5MFvrqkUL5FbtyyzZuOeOt0zdeRe4UY7ct+A==",

"license": "MIT",

"dependencies": {

"call-bind-apply-helpers": "^1.0.1",

"es-errors": "^1.3.0",

"gopd": "^1.2.0"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/ecdsa-sig-formatter": {

"version": "1.0.11",

"resolved": "https://registry.npmjs.org/ecdsa-sig-formatter/-/ecdsa-sig-formatter-1.0.11.tgz",

"integrity": "sha512-nagl3RYrbNv6kQkeJIpt6NJZy8twLB/2vtz6yN9Z4vRKHN4/QZJIEbqohALSgwKdnksuY3k5Addp5lg8sVoVcQ==",

"license": "Apache-2.0",

"dependencies": {

"safe-buffer": "^5.0.1"

}

},

"node\_modules/ee-first": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",

"integrity": "sha512-WMwm9LhRUo+WUaRN+vRuETqG89IgZphVSNkdFgeb6sS/E4OrDIN7t48CAewSHXc6C8lefD8KKfr5vY61brQlow==",

"license": "MIT"

},

"node\_modules/ejs": {

"version": "3.1.10",

"resolved": "https://registry.npmjs.org/ejs/-/ejs-3.1.10.tgz",

"integrity": "sha512-UeJmFfOrAQS8OJWPZ4qtgHyWExa088/MtK5UEyoJGFH67cDEXkZSviOiKRCZ4Xij0zxI3JECgYs3oKx+AizQBA==",

"license": "Apache-2.0",

"dependencies": {

"jake": "^10.8.5"

},

"bin": {

"ejs": "bin/cli.js"

},

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/encodeurl": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-2.0.0.tgz",

"integrity": "sha512-Q0n9HRi4m6JuGIV1eFlmvJB7ZEVxu93IrMyiMsGC0lrMJMWzRgx6WGquyfQgZVb31vhGgXnfmPNNXmxnOkRBrg==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/es-define-property": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/es-define-property/-/es-define-property-1.0.1.tgz",

"integrity": "sha512-e3nRfgfUZ4rNGL232gUgX06QNyyez04KdjFrF+LTRoOXmrOgFKDg4BCdsjW8EnT69eqdYGmRpJwiPVYNrCaW3g==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-errors": {

"version": "1.3.0",

"resolved": "https://registry.npmjs.org/es-errors/-/es-errors-1.3.0.tgz",

"integrity": "sha512-Zf5H2Kxt2xjTvbJvP2ZWLEICxA6j+hAmMzIlypy4xcBg1vKVnx89Wy0GbS+kf5cwCVFFzdCFh2XSCFNULS6csw==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-object-atoms": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/es-object-atoms/-/es-object-atoms-1.1.1.tgz",

"integrity": "sha512-FGgH2h8zKNim9ljj7dankFPcICIK9Cp5bm+c2gQSYePhpaG5+esrLODihIorn+Pe6FGJzWhXQotPv73jTaldXA==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-set-tostringtag": {

"version": "2.1.0",

"resolved": "https://registry.npmjs.org/es-set-tostringtag/-/es-set-tostringtag-2.1.0.tgz",

"integrity": "sha512-j6vWzfrGVfyXxge+O0x5sh6cvxAog0a/4Rdd2K36zCMV5eJ+/+tOAngRO8cODMNWbVRdVlmGZQL2YS3yR8bIUA==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"get-intrinsic": "^1.2.6",

"has-tostringtag": "^1.0.2",

"hasown": "^2.0.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/escape-html": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",

"integrity": "sha512-NiSupZ4OeuGwr68lGIeym/ksIZMJodUGOSCZ/FSnTxcrekbvqrgdUxlJOMpijaKZVjAJrWrGs/6Jy8OMuyj9ow==",

"license": "MIT"

},

"node\_modules/etag": {

"version": "1.8.1",

"resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",

"integrity": "sha512-aIL5Fx7mawVa300al2BnEE4iNvo1qETxLrPI/o05L7z6go7fCw1J6EQmbK4FmJ2AS7kgVF/KEZWufBfdClMcPg==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/express": {

"version": "4.21.2",

"resolved": "https://registry.npmjs.org/express/-/express-4.21.2.tgz",

"integrity": "sha512-28HqgMZAmih1Czt9ny7qr6ek2qddF4FclbMzwhCREB6OFfH+rXAnuNCwo1/wFvrtbgsQDb4kSbX9de9lFbrXnA==",

"license": "MIT",

"dependencies": {

"accepts": "~1.3.8",

"array-flatten": "1.1.1",

"body-parser": "1.20.3",

"content-disposition": "0.5.4",

"content-type": "~1.0.4",

"cookie": "0.7.1",

"cookie-signature": "1.0.6",

"debug": "2.6.9",

"depd": "2.0.0",

"encodeurl": "~2.0.0",

"escape-html": "~1.0.3",

"etag": "~1.8.1",

"finalhandler": "1.3.1",

"fresh": "0.5.2",

"http-errors": "2.0.0",

"merge-descriptors": "1.0.3",

"methods": "~1.1.2",

"on-finished": "2.4.1",

"parseurl": "~1.3.3",

"path-to-regexp": "0.1.12",

"proxy-addr": "~2.0.7",

"qs": "6.13.0",

"range-parser": "~1.2.1",

"safe-buffer": "5.2.1",

"send": "0.19.0",

"serve-static": "1.16.2",

"setprototypeof": "1.2.0",

"statuses": "2.0.1",

"type-is": "~1.6.18",

"utils-merge": "1.0.1",

"vary": "~1.1.2"

},

"engines": {

"node": ">= 0.10.0"

},

"funding": {

"type": "opencollective",

"url": "https://opencollective.com/express"

}

},

"node\_modules/filelist": {

"version": "1.0.4",

"resolved": "https://registry.npmjs.org/filelist/-/filelist-1.0.4.tgz",

"integrity": "sha512-w1cEuf3S+DrLCQL7ET6kz+gmlJdbq9J7yXCSjK/OZCPA+qEN1WyF4ZAf0YYJa4/shHJra2t/d/r8SV4Ji+x+8Q==",

"license": "Apache-2.0",

"dependencies": {

"minimatch": "^5.0.1"

}

},

"node\_modules/filelist/node\_modules/brace-expansion": {

"version": "2.0.2",

"resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-2.0.2.tgz",

"integrity": "sha512-Jt0vHyM+jmUBqojB7E1NIYadt0vI0Qxjxd2TErW94wDz+E2LAm5vKMXXwg6ZZBTHPuUlDgQHKXvjGBdfcF1ZDQ==",

"license": "MIT",

"dependencies": {

"balanced-match": "^1.0.0"

}

},

"node\_modules/filelist/node\_modules/minimatch": {

"version": "5.1.6",

"resolved": "https://registry.npmjs.org/minimatch/-/minimatch-5.1.6.tgz",

"integrity": "sha512-lKwV/1brpG6mBUFHtb7NUmtABCb2WZZmm2wNiOA5hAb8VdCS4B3dtMWyvcoViccwAW/COERjXLt0zP1zXUN26g==",

"license": "ISC",

"dependencies": {

"brace-expansion": "^2.0.1"

},

"engines": {

"node": ">=10"

}

},

"node\_modules/finalhandler": {

"version": "1.3.1",

"resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-1.3.1.tgz",

"integrity": "sha512-6BN9trH7bp3qvnrRyzsBz+g3lZxTNZTbVO2EV1CS0WIcDbawYVdYvGflME/9QP0h0pYlCDBCTjYa9nZzMDpyxQ==",

"license": "MIT",

"dependencies": {

"debug": "2.6.9",

"encodeurl": "~2.0.0",

"escape-html": "~1.0.3",

"on-finished": "2.4.1",

"parseurl": "~1.3.3",

"statuses": "2.0.1",

"unpipe": "~1.0.0"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/follow-redirects": {

"version": "1.15.11",

"resolved": "https://registry.npmjs.org/follow-redirects/-/follow-redirects-1.15.11.tgz",

"integrity": "sha512-deG2P0JfjrTxl50XGCDyfI97ZGVCxIpfKYmfyrQ54n5FO/0gfIES8C/Psl6kWVDolizcaaxZJnTS0QSMxvnsBQ==",

"funding": [

{

"type": "individual",

"url": "https://github.com/sponsors/RubenVerborgh"

}

],

"license": "MIT",

"engines": {

"node": ">=4.0"

},

"peerDependenciesMeta": {

"debug": {

"optional": true

}

}

},

"node\_modules/form-data": {

"version": "4.0.4",

"resolved": "https://registry.npmjs.org/form-data/-/form-data-4.0.4.tgz",

"integrity": "sha512-KrGhL9Q4zjj0kiUt5OO4Mr/A/jlI2jDYs5eHBpYHPcBEVSiipAvn2Ko2HnPe20rmcuuvMHNdZFp+4IlGTMF0Ow==",

"license": "MIT",

"dependencies": {

"asynckit": "^0.4.0",

"combined-stream": "^1.0.8",

"es-set-tostringtag": "^2.1.0",

"hasown": "^2.0.2",

"mime-types": "^2.1.12"

},

"engines": {

"node": ">= 6"

}

},

"node\_modules/forwarded": {

"version": "0.2.0",

"resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",

"integrity": "sha512-buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9wyablzMzOA5z9xA9L1KNjk/Nt6MT9aYow==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/fresh": {

"version": "0.5.2",

"resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",

"integrity": "sha512-zJ2mQYM18rEFOudeV4GShTGIQ7RbzA7ozbU9I/XBpm7kqgMywgmylMwXHxZJmkVoYkna9d2pVXVXPdYTP9ej8Q==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/function-bind": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/function-bind/-/function-bind-1.1.2.tgz",

"integrity": "sha512-7XHNxH7qX9xG5mIwxkhumTox/MIRNcOgDrxWsMt2pAr23WHp6MrRlN7FBSFpCpr+oVO0F744iUgR82nJMfG2SA==",

"license": "MIT",

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/generate-function": {

"version": "2.3.1",

"resolved": "https://registry.npmjs.org/generate-function/-/generate-function-2.3.1.tgz",

"integrity": "sha512-eeB5GfMNeevm/GRYq20ShmsaGcmI81kIX2K9XQx5miC8KdHaC6Jm0qQ8ZNeGOi7wYB8OsdxKs+Y2oVuTFuVwKQ==",

"license": "MIT",

"dependencies": {

"is-property": "^1.0.2"

}

},

"node\_modules/get-intrinsic": {

"version": "1.3.0",

"resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-1.3.0.tgz",

"integrity": "sha512-9fSjSaos/fRIVIp+xSJlE6lfwhES7LNtKaCBIamHsjr2na1BiABJPo0mOjjz8GJDURarmCPGqaiVg5mfjb98CQ==",

"license": "MIT",

"dependencies": {

"call-bind-apply-helpers": "^1.0.2",

"es-define-property": "^1.0.1",

"es-errors": "^1.3.0",

"es-object-atoms": "^1.1.1",

"function-bind": "^1.1.2",

"get-proto": "^1.0.1",

"gopd": "^1.2.0",

"has-symbols": "^1.1.0",

"hasown": "^2.0.2",

"math-intrinsics": "^1.1.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/get-proto": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/get-proto/-/get-proto-1.0.1.tgz",

"integrity": "sha512-sTSfBjoXBp89JvIKIefqw7U2CCebsc74kiY6awiGogKtoSGbgjYE/G/+l9sF3MWFPNc9IcoOC4ODfKHfxFmp0g==",

"license": "MIT",

"dependencies": {

"dunder-proto": "^1.0.1",

"es-object-atoms": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/gopd": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/gopd/-/gopd-1.2.0.tgz",

"integrity": "sha512-ZUKRh6/kUFoAiTAtTYPZJ3hw9wNxx+BIBOijnlG9PnrJsCcSjs1wyyD6vJpaYtgnzDrKYRSqf3OO6Rfa93xsRg==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/has-flag": {

"version": "4.0.0",

"resolved": "https://registry.npmjs.org/has-flag/-/has-flag-4.0.0.tgz",

"integrity": "sha512-EykJT/Q1KjTWctppgIAgfSO0tKVuZUjhgMr17kqTumMl6Afv3EISleU7qZUzoXDFTAHTDC4NOoG/ZxU3EvlMPQ==",

"license": "MIT",

"engines": {

"node": ">=8"

}

},

"node\_modules/has-symbols": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-1.1.0.tgz",

"integrity": "sha512-1cDNdwJ2Jaohmb3sg4OmKaMBwuC48sYni5HUw2DvsC8LjGTLK9h+eb1X6RyuOHe4hT0ULCW68iomhjUoKUqlPQ==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/has-tostringtag": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/has-tostringtag/-/has-tostringtag-1.0.2.tgz",

"integrity": "sha512-NqADB8VjPFLM2V0VvHUewwwsw0ZWBaIdgo+ieHtK3hasLz4qeCRjYcqfB6AQrBggRKppKF8L52/VqdVsO47Dlw==",

"license": "MIT",

"dependencies": {

"has-symbols": "^1.0.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/hasown": {

"version": "2.0.2",

"resolved": "https://registry.npmjs.org/hasown/-/hasown-2.0.2.tgz",

"integrity": "sha512-0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0cII9y+CyS8rG3nL48Bclra9HmKhVVUHyPWNH5Y7xDwAB7bfgSjkUMQ==",

"license": "MIT",

"dependencies": {

"function-bind": "^1.1.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/http-errors": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/http-errors/-/http-errors-2.0.0.tgz",

"integrity": "sha512-FtwrG/euBzaEjYeRqOgly7G0qviiXoJWnvEH2Z1plBdXgbyjv34pHTSb9zoeHMyDy33+DWy5Wt9Wo+TURtOYSQ==",

"license": "MIT",

"dependencies": {

"depd": "2.0.0",

"inherits": "2.0.4",

"setprototypeof": "1.2.0",

"statuses": "2.0.1",

"toidentifier": "1.0.1"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/iconv-lite": {

"version": "0.4.24",

"resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.24.tgz",

"integrity": "sha512-v3MXnZAcvnywkTUEZomIActle7RXXeedOR31wwl7VlyoXO4Qi9arvSenNQWne1TcRwhCL1HwLI21bEqdpj8/rA==",

"license": "MIT",

"dependencies": {

"safer-buffer": ">= 2.1.2 < 3"

},

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/inherits": {

"version": "2.0.4",

"resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",

"integrity": "sha512-k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSM9zGSoIFeMpXKxa4dYeZIQqewQ==",

"license": "ISC"

},

"node\_modules/ipaddr.js": {

"version": "1.9.1",

"resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",

"integrity": "sha512-0KI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ6R/w18ZlXSHBYXiYUPO3g==",

"license": "MIT",

"engines": {

"node": ">= 0.10"

}

},

"node\_modules/is-property": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/is-property/-/is-property-1.0.2.tgz",

"integrity": "sha512-Ks/IoX00TtClbGQr4TWXemAnktAQvYB7HzcCxDGqEZU6oCmb2INHuOoKxbtR+HFkmYWBKv/dOZtGRiAjDhj92g==",

"license": "MIT"

},

"node\_modules/jake": {

"version": "10.9.2",

"resolved": "https://registry.npmjs.org/jake/-/jake-10.9.2.tgz",

"integrity": "sha512-2P4SQ0HrLQ+fw6llpLnOaGAvN2Zu6778SJMrCUwns4fOoG9ayrTiZk3VV8sCPkVZF8ab0zksVpS8FDY5pRCNBA==",

"license": "Apache-2.0",

"dependencies": {

"async": "^3.2.3",

"chalk": "^4.0.2",

"filelist": "^1.0.4",

"minimatch": "^3.1.2"

},

"bin": {

"jake": "bin/cli.js"

},

"engines": {

"node": ">=10"

}

},

"node\_modules/jsonwebtoken": {

"version": "9.0.2",

"resolved": "https://registry.npmjs.org/jsonwebtoken/-/jsonwebtoken-9.0.2.tgz",

"integrity": "sha512-PRp66vJ865SSqOlgqS8hujT5U4AOgMfhrwYIuIhfKaoSCZcirrmASQr8CX7cUg+RMih+hgznrjp99o+W4pJLHQ==",

"license": "MIT",

"dependencies": {

"jws": "^3.2.2",

"lodash.includes": "^4.3.0",

"lodash.isboolean": "^3.0.3",

"lodash.isinteger": "^4.0.4",

"lodash.isnumber": "^3.0.3",

"lodash.isplainobject": "^4.0.6",

"lodash.isstring": "^4.0.1",

"lodash.once": "^4.0.0",

"ms": "^2.1.1",

"semver": "^7.5.4"

},

"engines": {

"node": ">=12",

"npm": ">=6"

}

},

"node\_modules/jsonwebtoken/node\_modules/ms": {

"version": "2.1.3",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",

"integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA==",

"license": "MIT"

},

"node\_modules/jwa": {

"version": "1.4.2",

"resolved": "https://registry.npmjs.org/jwa/-/jwa-1.4.2.tgz",

"integrity": "sha512-eeH5JO+21J78qMvTIDdBXidBd6nG2kZjg5Ohz/1fpa28Z4CcsWUzJ1ZZyFq/3z3N17aZy+ZuBoHljASbL1WfOw==",

"license": "MIT",

"dependencies": {

"buffer-equal-constant-time": "^1.0.1",

"ecdsa-sig-formatter": "1.0.11",

"safe-buffer": "^5.0.1"

}

},

"node\_modules/jws": {

"version": "3.2.2",

"resolved": "https://registry.npmjs.org/jws/-/jws-3.2.2.tgz",

"integrity": "sha512-YHlZCB6lMTllWDtSPHz/ZXTsi8S00usEV6v1tjq8tOUZzw7DpSDWVXjXDre6ed1w/pd495ODpHZYSdkRTsa0HA==",

"license": "MIT",

"dependencies": {

"jwa": "^1.4.1",

"safe-buffer": "^5.0.1"

}

},

"node\_modules/lodash.includes": {

"version": "4.3.0",

"resolved": "https://registry.npmjs.org/lodash.includes/-/lodash.includes-4.3.0.tgz",

"integrity": "sha512-W3Bx6mdkRTGtlJISOvVD/lbqjTlPPUDTMnlXZFnVwi9NKJ6tiAk6LVdlhZMm17VZisqhKcgzpO5Wz91PCt5b0w==",

"license": "MIT"

},

"node\_modules/lodash.isboolean": {

"version": "3.0.3",

"resolved": "https://registry.npmjs.org/lodash.isboolean/-/lodash.isboolean-3.0.3.tgz",

"integrity": "sha512-Bz5mupy2SVbPHURB98VAcw+aHh4vRV5IPNhILUCsOzRmsTmSQ17jIuqopAentWoehktxGd9e/hbIXq980/1QJg==",

"license": "MIT"

},

"node\_modules/lodash.isinteger": {

"version": "4.0.4",

"resolved": "https://registry.npmjs.org/lodash.isinteger/-/lodash.isinteger-4.0.4.tgz",

"integrity": "sha512-DBwtEWN2caHQ9/imiNeEA5ys1JoRtRfY3d7V9wkqtbycnAmTvRRmbHKDV4a0EYc678/dia0jrte4tjYwVBaZUA==",

"license": "MIT"

},

"node\_modules/lodash.isnumber": {

"version": "3.0.3",

"resolved": "https://registry.npmjs.org/lodash.isnumber/-/lodash.isnumber-3.0.3.tgz",

"integrity": "sha512-QYqzpfwO3/CWf3XP+Z+tkQsfaLL/EnUlXWVkIk5FUPc4sBdTehEqZONuyRt2P67PXAk+NXmTBcc97zw9t1FQrw==",

"license": "MIT"

},

"node\_modules/lodash.isplainobject": {

"version": "4.0.6",

"resolved": "https://registry.npmjs.org/lodash.isplainobject/-/lodash.isplainobject-4.0.6.tgz",

"integrity": "sha512-oSXzaWypCMHkPC3NvBEaPHf0KsA5mvPrOPgQWDsbg8n7orZ290M0BmC/jgRZ4vcJ6DTAhjrsSYgdsW/F+MFOBA==",

"license": "MIT"

},

"node\_modules/lodash.isstring": {

"version": "4.0.1",

"resolved": "https://registry.npmjs.org/lodash.isstring/-/lodash.isstring-4.0.1.tgz",

"integrity": "sha512-0wJxfxH1wgO3GrbuP+dTTk7op+6L41QCXbGINEmD+ny/G/eCqGzxyCsh7159S+mgDDcoarnBw6PC1PS5+wUGgw==",

"license": "MIT"

},

"node\_modules/lodash.once": {

"version": "4.1.1",

"resolved": "https://registry.npmjs.org/lodash.once/-/lodash.once-4.1.1.tgz",

"integrity": "sha512-Sb487aTOCr9drQVL8pIxOzVhafOjZN9UU54hiN8PU3uAiSV7lx1yYNpbNmex2PK6dSJoNTSJUUswT651yww3Mg==",

"license": "MIT"

},

"node\_modules/long": {

"version": "5.3.2",

"resolved": "https://registry.npmjs.org/long/-/long-5.3.2.tgz",

"integrity": "sha512-mNAgZ1GmyNhD7AuqnTG3/VQ26o760+ZYBPKjPvugO8+nLbYfX6TVpJPseBvopbdY+qpZ/lKUnmEc1LeZYS3QAA==",

"license": "Apache-2.0"

},

"node\_modules/lru-cache": {

"version": "7.18.3",

"resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-7.18.3.tgz",

"integrity": "sha512-jumlc0BIUrS3qJGgIkWZsyfAM7NCWiBcCDhnd+3NNM5KbBmLTgHVfWBcg6W+rLUsIpzpERPsvwUP7CckAQSOoA==",

"license": "ISC",

"engines": {

"node": ">=12"

}

},

"node\_modules/lru.min": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/lru.min/-/lru.min-1.1.2.tgz",

"integrity": "sha512-Nv9KddBcQSlQopmBHXSsZVY5xsdlZkdH/Iey0BlcBYggMd4two7cZnKOK9vmy3nY0O5RGH99z1PCeTpPqszUYg==",

"license": "MIT",

"engines": {

"bun": ">=1.0.0",

"deno": ">=1.30.0",

"node": ">=8.0.0"

},

"funding": {

"type": "github",

"url": "https://github.com/sponsors/wellwelwel"

}

},

"node\_modules/math-intrinsics": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/math-intrinsics/-/math-intrinsics-1.1.0.tgz",

"integrity": "sha512-/IXtbwEk5HTPyEwyKX6hGkYXxM9nbj64B+ilVJnC/R6B0pH5G4V3b0pVbL7DBj4tkhBAppbQUlf6F6Xl9LHu1g==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/media-typer": {

"version": "0.3.0",

"resolved": "https://registry.npmjs.org/media-typer/-/media-typer-0.3.0.tgz",

"integrity": "sha512-dq+qelQ9akHpcOl/gUVRTxVIOkAJ1wR3QAvb4RsVjS8oVoFjDGTc679wJYmUmknUF5HwMLOgb5O+a3KxfWapPQ==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/merge-descriptors": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-1.0.3.tgz",

"integrity": "sha512-gaNvAS7TZ897/rVaZ0nMtAyxNyi/pdbjbAwUpFQpN70GqnVfOiXpeUUMKRBmzXaSQ8DdTX4/0ms62r2K+hE6mQ==",

"license": "MIT",

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/methods": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",

"integrity": "sha512-iclAHeNqNm68zFtnZ0e+1L2yUIdvzNoauKU4WBA3VvH/vPFieF7qfRlwUZU+DA9P9bPXIS90ulxoUoCH23sV2w==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/mime": {

"version": "1.6.0",

"resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",

"integrity": "sha512-x0Vn8spI+wuJ1O6S7gnbaQg8Pxh4NNHb7KSINmEWKiPE4RKOplvijn+NkmYmmRgP68mc70j2EbeTFRsrswaQeg==",

"license": "MIT",

"bin": {

"mime": "cli.js"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/mime-db": {

"version": "1.52.0",

"resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",

"integrity": "sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETVlOoK1UC2nU3gYvg==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/mime-types": {

"version": "2.1.35",

"resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",

"integrity": "sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1zxuWL50hzaeXiw==",

"license": "MIT",

"dependencies": {

"mime-db": "1.52.0"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/minimatch": {

"version": "3.1.2",

"resolved": "https://registry.npmjs.org/minimatch/-/minimatch-3.1.2.tgz",

"integrity": "sha512-J7p63hRiAjw1NDEww1W7i37+ByIrOWO5XQQAzZ3VOcL0PNybwpfmV/N05zFAzwQ9USyEcX6t3UO+K5aqBQOIHw==",

"license": "ISC",

"dependencies": {

"brace-expansion": "^1.1.7"

},

"engines": {

"node": "\*"

}

},

"node\_modules/ms": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",

"integrity": "sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==",

"license": "MIT"

},

"node\_modules/mysql2": {

"version": "3.14.1",

"resolved": "https://registry.npmjs.org/mysql2/-/mysql2-3.14.1.tgz",

"integrity": "sha512-7ytuPQJjQB8TNAYX/H2yhL+iQOnIBjAMam361R7UAL0lOVXWjtdrmoL9HYKqKoLp/8UUTRcvo1QPvK9KL7wA8w==",

"license": "MIT",

"dependencies": {

"aws-ssl-profiles": "^1.1.1",

"denque": "^2.1.0",

"generate-function": "^2.3.1",

"iconv-lite": "^0.6.3",

"long": "^5.2.1",

"lru.min": "^1.0.0",

"named-placeholders": "^1.1.3",

"seq-queue": "^0.0.5",

"sqlstring": "^2.3.2"

},

"engines": {

"node": ">= 8.0"

}

},

"node\_modules/mysql2/node\_modules/iconv-lite": {

"version": "0.6.3",

"resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.6.3.tgz",

"integrity": "sha512-4fCk79wshMdzMp2rH06qWrJE4iolqLhCUH+OiuIgU++RB0+94NlDL81atO7GX55uUKueo0txHNtvEyI6D7WdMw==",

"license": "MIT",

"dependencies": {

"safer-buffer": ">= 2.1.2 < 3.0.0"

},

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/named-placeholders": {

"version": "1.1.3",

"resolved": "https://registry.npmjs.org/named-placeholders/-/named-placeholders-1.1.3.tgz",

"integrity": "sha512-eLoBxg6wE/rZkJPhU/xRX1WTpkFEwDJEN96oxFrTsqBdbT5ec295Q+CoHrL9IT0DipqKhmGcaZmwOt8OON5x1w==",

"license": "MIT",

"dependencies": {

"lru-cache": "^7.14.1"

},

"engines": {

"node": ">=12.0.0"

}

},

"node\_modules/negotiator": {

"version": "0.6.3",

"resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.3.tgz",

"integrity": "sha512-+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGlQk/k5giakcKsuxCObBRu6DSm9opw/O6slWbJdghQM4bBg==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/node-addon-api": {

"version": "8.3.1",

"resolved": "https://registry.npmjs.org/node-addon-api/-/node-addon-api-8.3.1.tgz",

"integrity": "sha512-lytcDEdxKjGJPTLEfW4mYMigRezMlyJY8W4wxJK8zE533Jlb8L8dRuObJFWg2P+AuOIxoCgKF+2Oq4d4Zd0OUA==",

"license": "MIT",

"engines": {

"node": "^18 || ^20 || >= 21"

}

},

"node\_modules/node-gyp-build": {

"version": "4.8.4",

"resolved": "https://registry.npmjs.org/node-gyp-build/-/node-gyp-build-4.8.4.tgz",

"integrity": "sha512-LA4ZjwlnUblHVgq0oBF3Jl/6h/Nvs5fzBLwdEF4nuxnFdsfajde4WfxtJr3CaiH+F6ewcIB/q4jQ4UzPyid+CQ==",

"license": "MIT",

"bin": {

"node-gyp-build": "bin.js",

"node-gyp-build-optional": "optional.js",

"node-gyp-build-test": "build-test.js"

}

},

"node\_modules/nodemailer": {

"version": "7.0.5",

"resolved": "https://registry.npmjs.org/nodemailer/-/nodemailer-7.0.5.tgz",

"integrity": "sha512-nsrh2lO3j4GkLLXoeEksAMgAOqxOv6QumNRVQTJwKH4nuiww6iC2y7GyANs9kRAxCexg3+lTWM3PZ91iLlVjfg==",

"license": "MIT-0",

"engines": {

"node": ">=6.0.0"

}

},

"node\_modules/object-assign": {

"version": "4.1.1",

"resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",

"integrity": "sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU5hj+BcUg==",

"license": "MIT",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/object-inspect": {

"version": "1.13.4",

"resolved": "https://registry.npmjs.org/object-inspect/-/object-inspect-1.13.4.tgz",

"integrity": "sha512-W67iLl4J2EXEGTbfeHCffrjDfitvLANg0UlX3wFUUSTx92KXRFegMHUVgSqE+wvhAbi4WqjGg9czysTV2Epbew==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/on-finished": {

"version": "2.4.1",

"resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.4.1.tgz",

"integrity": "sha512-oVlzkg3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+brtuJjePWiYF/ClmuDr8Ch5+kg==",

"license": "MIT",

"dependencies": {

"ee-first": "1.1.1"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/parseurl": {

"version": "1.3.3",

"resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",

"integrity": "sha512-CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL+qQ==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/path-to-regexp": {

"version": "0.1.12",

"resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.12.tgz",

"integrity": "sha512-RA1GjUVMnvYFxuqovrEqZoxxW5NUZqbwKtYz/Tt7nXerk0LbLblQmrsgdeOxV5SFHf0UDggjS/bSeOZwt1pmEQ==",

"license": "MIT"

},

"node\_modules/proxy-addr": {

"version": "2.0.7",

"resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",

"integrity": "sha512-llQsMLSUDUPT44jdrU/O37qlnifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/gJEdZU7KMraoK1+XYAg==",

"license": "MIT",

"dependencies": {

"forwarded": "0.2.0",

"ipaddr.js": "1.9.1"

},

"engines": {

"node": ">= 0.10"

}

},

"node\_modules/proxy-from-env": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/proxy-from-env/-/proxy-from-env-1.1.0.tgz",

"integrity": "sha512-D+zkORCbA9f1tdWRK0RaCR3GPv50cMxcrz4X8k5LTSUD1Dkw47mKJEZQNunItRTkWwgtaUSo1RVFRIG9ZXiFYg==",

"license": "MIT"

},

"node\_modules/qs": {

"version": "6.13.0",

"resolved": "https://registry.npmjs.org/qs/-/qs-6.13.0.tgz",

"integrity": "sha512-+38qI9SOr8tfZ4QmJNplMUxqjbe7LKvvZgWdExBOmd+egZTtjLB67Gu0HRX3u/XOq7UU2Nx6nsjvS16Z9uwfpg==",

"license": "BSD-3-Clause",

"dependencies": {

"side-channel": "^1.0.6"

},

"engines": {

"node": ">=0.6"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/range-parser": {

"version": "1.2.1",

"resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",

"integrity": "sha512-Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRoyzfG3ccAveqCBrwr/2yxQ5BVd/GTl5agOwSg==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/raw-body": {

"version": "2.5.2",

"resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.2.tgz",

"integrity": "sha512-8zGqypfENjCIqGhgXToC8aB2r7YrBX+AQAfIPs/Mlk+BtPTztOvTS01NRW/3Eh60J+a48lt8qsCzirQ6loCVfA==",

"license": "MIT",

"dependencies": {

"bytes": "3.1.2",

"http-errors": "2.0.0",

"iconv-lite": "0.4.24",

"unpipe": "1.0.0"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/safe-buffer": {

"version": "5.2.1",

"resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",

"integrity": "sha512-rp3So07KcdmmKbGvgaNxQSJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCXmpuYvmCXQ==",

"funding": [

{

"type": "github",

"url": "https://github.com/sponsors/feross"

},

{

"type": "patreon",

"url": "https://www.patreon.com/feross"

},

{

"type": "consulting",

"url": "https://feross.org/support"

}

],

"license": "MIT"

},

"node\_modules/safer-buffer": {

"version": "2.1.2",

"resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",

"integrity": "sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQriUtg==",

"license": "MIT"

},

"node\_modules/semver": {

"version": "7.7.2",

"resolved": "https://registry.npmjs.org/semver/-/semver-7.7.2.tgz",

"integrity": "sha512-RF0Fw+rO5AMf9MAyaRXI4AV0Ulj5lMHqVxxdSgiVbixSCXoEmmX/jk0CuJw4+3SqroYO9VoUh+HcuJivvtJemA==",

"license": "ISC",

"bin": {

"semver": "bin/semver.js"

},

"engines": {

"node": ">=10"

}

},

"node\_modules/send": {

"version": "0.19.0",

"resolved": "https://registry.npmjs.org/send/-/send-0.19.0.tgz",

"integrity": "sha512-dW41u5VfLXu8SJh5bwRmyYUbAoSB3c9uQh6L8h/KtsFREPWpbX1lrljJo186Jc4nmci/sGUZ9a0a0J2zgfq2hw==",

"license": "MIT",

"dependencies": {

"debug": "2.6.9",

"depd": "2.0.0",

"destroy": "1.2.0",

"encodeurl": "~1.0.2",

"escape-html": "~1.0.3",

"etag": "~1.8.1",

"fresh": "0.5.2",

"http-errors": "2.0.0",

"mime": "1.6.0",

"ms": "2.1.3",

"on-finished": "2.4.1",

"range-parser": "~1.2.1",

"statuses": "2.0.1"

},

"engines": {

"node": ">= 0.8.0"

}

},

"node\_modules/send/node\_modules/encodeurl": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",

"integrity": "sha512-TPJXq8JqFaVYm2CWmPvnP2Iyo4ZSM7/QKcSmuMLDObfpH5fi7RUGmd/rTDf+rut/saiDiQEeVTNgAmJEdAOx0w==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/send/node\_modules/ms": {

"version": "2.1.3",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",

"integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA==",

"license": "MIT"

},

"node\_modules/seq-queue": {

"version": "0.0.5",

"resolved": "https://registry.npmjs.org/seq-queue/-/seq-queue-0.0.5.tgz",

"integrity": "sha512-hr3Wtp/GZIc/6DAGPDcV4/9WoZhjrkXsi5B/07QgX8tsdc6ilr7BFM6PM6rbdAX1kFSDYeZGLipIZZKyQP0O5Q=="

},

"node\_modules/serve-static": {

"version": "1.16.2",

"resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.16.2.tgz",

"integrity": "sha512-VqpjJZKadQB/PEbEwvFdO43Ax5dFBZ2UECszz8bQ7pi7wt//PWe1P6MN7eCnjsatYtBT6EuiClbjSWP2WrIoTw==",

"license": "MIT",

"dependencies": {

"encodeurl": "~2.0.0",

"escape-html": "~1.0.3",

"parseurl": "~1.3.3",

"send": "0.19.0"

},

"engines": {

"node": ">= 0.8.0"

}

},

"node\_modules/setprototypeof": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.2.0.tgz",

"integrity": "sha512-E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrlODCrkoxMEeNi/XZIwuRvY4XNwYMJpw==",

"license": "ISC"

},

"node\_modules/side-channel": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/side-channel/-/side-channel-1.1.0.tgz",

"integrity": "sha512-ZX99e6tRweoUXqR+VBrslhda51Nh5MTQwou5tnUDgbtyM0dBgmhEDtWGP/xbKn6hqfPRHujUNwz5fy/wbbhnpw==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"object-inspect": "^1.13.3",

"side-channel-list": "^1.0.0",

"side-channel-map": "^1.0.1",

"side-channel-weakmap": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/side-channel-list": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/side-channel-list/-/side-channel-list-1.0.0.tgz",

"integrity": "sha512-FCLHtRD/gnpCiCHEiJLOwdmFP+wzCmDEkc9y7NsYxeF4u7Btsn1ZuwgwJGxImImHicJArLP4R0yX4c2KCrMrTA==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"object-inspect": "^1.13.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/side-channel-map": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/side-channel-map/-/side-channel-map-1.0.1.tgz",

"integrity": "sha512-VCjCNfgMsby3tTdo02nbjtM/ewra6jPHmpThenkTYh8pG9ucZ/1P8So4u4FGBek/BjpOVsDCMoLA/iuBKIFXRA==",

"license": "MIT",

"dependencies": {

"call-bound": "^1.0.2",

"es-errors": "^1.3.0",

"get-intrinsic": "^1.2.5",

"object-inspect": "^1.13.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/side-channel-weakmap": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/side-channel-weakmap/-/side-channel-weakmap-1.0.2.tgz",

"integrity": "sha512-WPS/HvHQTYnHisLo9McqBHOJk2FkHO/tlpvldyrnem4aeQp4hai3gythswg6p01oSoTl58rcpiFAjF2br2Ak2A==",

"license": "MIT",

"dependencies": {

"call-bound": "^1.0.2",

"es-errors": "^1.3.0",

"get-intrinsic": "^1.2.5",

"object-inspect": "^1.13.3",

"side-channel-map": "^1.0.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/sqlstring": {

"version": "2.3.3",

"resolved": "https://registry.npmjs.org/sqlstring/-/sqlstring-2.3.3.tgz",

"integrity": "sha512-qC9iz2FlN7DQl3+wjwn3802RTyjCx7sDvfQEXchwa6CWOx07/WVfh91gBmQ9fahw8snwGEWU3xGzOt4tFyHLxg==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/statuses": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",

"integrity": "sha512-RwNA9Z/7PrK06rYLIzFMlaF+l73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB2fOC4AjuYSEndS7N/mTCbkdQ==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/supports-color": {

"version": "7.2.0",

"resolved": "https://registry.npmjs.org/supports-color/-/supports-color-7.2.0.tgz",

"integrity": "sha512-qpCAvRl9stuOHveKsn7HncJRvv501qIacKzQlO/+Lwxc9+0q2wLyv4Dfvt80/DPn2pqOBsJdDiogXGR9+OvwRw==",

"license": "MIT",

"dependencies": {

"has-flag": "^4.0.0"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/toidentifier": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",

"integrity": "sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTlZUtEe+JG7s5YhEz608PlAHRA==",

"license": "MIT",

"engines": {

"node": ">=0.6"

}

},

"node\_modules/type-is": {

"version": "1.6.18",

"resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",

"integrity": "sha512-TkRKr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFPW4ky9Q+iA+ma9BGm06XQBy8g==",

"license": "MIT",

"dependencies": {

"media-typer": "0.3.0",

"mime-types": "~2.1.24"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/unpipe": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",

"integrity": "sha512-pjy2bYhSsufwWlKwPc+l3cN7+wuJlK6uz0YdJEOlQDbl6jo/YlPi4mb8agUkVC8BF7V8NuzeyPNqRksA3hztKQ==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/utils-merge": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",

"integrity": "sha512-pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybbsHHoED/ztMtkv/VoYTYyShUn81hA==",

"license": "MIT",

"engines": {

"node": ">= 0.4.0"

}

},

"node\_modules/vary": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",

"integrity": "sha512-BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQOzmAKzYqYRAISoRhdQr3eIZ/PXqg==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

}

}

}

# Arquivo: package.json

{

"name": "oilsmart",

"version": "1.0.0",

"description": "Projeto reorganizado do OilSmart com Express",

"main": "server.js",

"scripts": {

"start": "node server.js"

},

"dependencies": {

"axios": "^1.11.0",

"bcrypt": "^6.0.0",

"bcryptjs": "^3.0.2",

"cors": "^2.8.5",

"dotenv": "^17.2.1",

"ejs": "^3.1.9",

"express": "^4.21.2",

"jsonwebtoken": "^9.0.2",

"mysql2": "^3.14.1",

"nodemailer": "^7.0.5"

},

"keywords": [],

"author": "",

"license": "ISC"

}

# Arquivo: public/css/adm.css

:root {

--primary-color: #213f57;

--testego: #384d5e;

--secondary-color: #b49434;

--accent-color: #d4b15f;

--text-color: #333;

--text-light: #777;

--border-color: #e0e0e0;

--success-color: #28a745;

--warning-color: #ffc107;

--danger-color: #dc3545;

--transition: all 0.3s ease;

--box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

--border-radius: 10px;

--card-shadow: 0 8px 25px rgba(0, 0, 0, 0.08);

}

/\* Dark mode variables \*/

@media (prefers-color-scheme: dark) {

:root {

--primary-color: #263f54;

--secondary-color: #b49434;

--accent-color: #d4b15f;

--text-color: #f0f0f0;

--text-light: #b0b0b0;

--border-color: #444;

--box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);

--card-shadow: 0 8px 25px rgba(0, 0, 0, 0.2);

}

}

/\* Automatic background colors based on mode \*/

@media (prefers-color-scheme: light) {

body {

--bg-color: #f9f9f9;

--light-color: #f8f9fa;

--dark-color: #343a40;

--f8f9fa: #dceeff;

--card-bg: #ffffff;

}

}

@media (prefers-color-scheme: dark) {

body {

--bg-color: #1a1a1a;

--light-color: #2a2a2a;

--dark-color: #f8f9fa;

--f8f9fa: #b6b6b6;

--card-bg: #2a2a2a;

}

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

html, body {

height: 100%;

width: 100%;

overflow: hidden;

}

body {

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

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

color: var(--text-color);

display: flex;

min-height: 100vh;

flex-direction: column;

}

.loading {

display: inline-block;

width: 20px;

height: 20px;

border: 3px solid rgba(255,255,255,.3);

border-radius: 50%;

border-top-color: #fff;

animation: spin 1s ease-in-out infinite;

}

@keyframes spin {

to { transform: rotate(360deg); }

}

/\* ===== SIDEBAR ===== \*/

.sidebar {

width: 280px;

background: var(--primary-color);

color: white;

display: flex;

flex-direction: column;

transition: all 0.3s ease;

box-shadow: var(--box-shadow);

}

.sidebar-header {

padding: 20px;

text-align: center;

border-bottom: 1px solid rgba(255, 255, 255, 0.1);

}

.sidebar-logo {

font-size: 2.5rem;

margin-bottom: 10px;

color: var(--secondary-color);

}

.sidebar-nav ul {

list-style: none;

padding: 20px 0;

}

.nav-item {

margin: 5px 0;

}

.nav-link {

display: flex;

align-items: center;

padding: 12px 20px;

color: rgba(255, 255, 255, 0.7);

text-decoration: none;

transition: var(--transition);

border-radius: var(--border-radius);

margin: 0 10px;

}

.nav-link:hover,

.nav-link.active {

background-color: rgba(255, 255, 255, 0.1);

color: var(--secondary-color);

border-left: 4px solid var(--secondary-color);

}

.nav-link i {

margin-right: 10px;

width: 20px;

text-align: center;

}

.sidebar-footer {

margin-top: auto;

padding: 20px;

display: flex;

align-items: center;

justify-content: space-between;

border-top: 1px solid rgba(255, 255, 255, 0.1);

}

.user-info {

display: flex;

align-items: center;

}

.user-avatar {

width: 40px;

height: 40px;

border-radius: 50%;

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

display: flex;

align-items: center;

justify-content: center;

margin-right: 10px;

}

.user-details {

display: flex;

flex-direction: column;

}

.user-name {

font-weight: 500;

}

.user-role {

font-size: 12px;

opacity: 0.7;

}

.logout-btn {

background: none;

border: none;

color: white;

cursor: pointer;

font-size: 18px;

transition: var(--transition);

border-radius: 50%;

width: 40px;

height: 40px;

display: flex;

align-items: center;

justify-content: center;

}

.logout-btn:hover {

color: var(--secondary-color);

background-color: rgba(255, 255, 255, 0.1);

}

/\* ===== MAIN CONTENT ===== \*/

.main-content {

flex: 1;

display: flex;

flex-direction: column;

height: 100vh;

}

.main-header {

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

padding: 15px 30px;

display: flex;

align-items: center;

justify-content: space-between;

box-shadow: var(--box-shadow);

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

}

.header-left {

display: flex;

align-items: center;

}

.sidebar-toggle {

background: none;

border: none;

font-size: 18px;

margin-right: 15px;

cursor: pointer;

color: var(--primary-color);

width: 40px;

height: 40px;

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

transition: var(--transition);

}

.sidebar-toggle:hover {

background-color: rgba(180, 148, 52, 0.1);

}

.page-title {

font-size: 24px;

font-weight: 700;

color: var(--primary-color);

}

.header-right {

display: flex;

align-items: center;

gap: 20px;

}

.search-box {

position: relative;

}

.search-box i {

position: absolute;

left: 10px;

top: 50%;

transform: translateY(-50%);

color: var(--text-light);

}

.search-box input {

padding: 10px 15px 10px 35px;

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

border-radius: 50px;

width: 250px;

transition: var(--transition);

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

color: var(--text-color);

}

.search-box input:focus {

outline: none;

border-color: var(--secondary-color);

box-shadow: 0 0 0 2px rgba(180, 148, 52, 0.2);

}

.notification-btn {

position: relative;

background: none;

border: none;

font-size: 18px;

color: var(--text-light);

cursor: pointer;

transition: var(--transition);

width: 40px;

height: 40px;

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

}

.notification-btn:hover {

color: var(--secondary-color);

background-color: rgba(180, 148, 52, 0.1);

}

.notification-badge {

position: absolute;

top: -5px;

right: -5px;

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

color: white;

font-size: 10px;

width: 16px;

height: 16px;

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

}

.date-time {

display: flex;

flex-direction: column;

font-size: 14px;

text-align: right;

}

.content-area {

flex: 1;

padding: 30px;

overflow-y: auto;

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

}

.content-section {

display: none;

}

.content-section.active {

display: block;

}

/\* ===== DASHBOARD METRICS ===== \*/

.metrics-grid {

display: grid;

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

gap: 20px;

margin-bottom: 30px;

}

.metric-card {

background: var(--card-bg);

border-radius: var(--border-radius);

padding: 25px;

box-shadow: var(--card-shadow);

display: flex;

align-items: center;

transition: var(--transition);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.metric-card:hover {

transform: translateY(-5px);

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);

}

.metric-icon {

width: 60px;

height: 60px;

border-radius: var(--border-radius);

background: linear-gradient(135deg, rgba(180, 148, 52, 0.1), rgba(212, 177, 95, 0.15));

display: flex;

align-items: center;

justify-content: center;

margin-right: 15px;

color: var(--secondary-color);

font-size: 24px;

}

.metric-content h3 {

font-size: 14px;

color: var(--text-light);

margin-bottom: 5px;

}

.metric-value {

font-size: 24px;

font-weight: 700;

margin-bottom: 5px;

color: var(--primary-color);

}

.metric-change {

font-size: 12px;

font-weight: 500;

}

.metric-change.positive {

color: var(--success-color);

}

.metric-change.negative {

color: var(--danger-color);

}

.metric-change.neutral {

color: var(--text-light);

}

/\* ===== CHARTS ===== \*/

.charts-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(500px, 1fr));

gap: 20px;

margin-bottom: 30px;

}

.chart-card {

background: var(--card-bg);

border-radius: var(--border-radius);

padding: 25px;

box-shadow: var(--card-shadow);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.chart-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 20px;

}

.chart-header h3 {

font-size: 18px;

font-weight: 600;

color: var(--primary-color);

}

.chart-header select {

padding: 8px 12px;

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

border-radius: var(--border-radius);

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

color: var(--text-color);

}

.chart-container {

height: 300px;

position: relative;

}

/\* ===== ACTIVITIES ===== \*/

.activities-section {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(400px, 1fr));

gap: 20px;

}

.activities-card {

background: var(--card-bg);

border-radius: var(--border-radius);

padding: 25px;

box-shadow: var(--card-shadow);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.card-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 20px;

padding-bottom: 15px;

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

}

.card-header h3 {

font-size: 18px;

font-weight: 600;

color: var(--primary-color);

}

.view-all {

color: var(--secondary-color);

text-decoration: none;

font-size: 14px;

transition: var(--transition);

font-weight: 500;

}

.view-all:hover {

text-decoration: underline;

}

.activities-list {

display: flex;

flex-direction: column;

gap: 15px;

}

.activity-item {

display: flex;

align-items: center;

padding: 10px 0;

}

.activity-icon {

width: 40px;

height: 40px;

border-radius: var(--border-radius);

background-color: rgba(180, 148, 52, 0.1);

display: flex;

align-items: center;

justify-content: center;

margin-right: 15px;

color: var(--secondary-color);

}

.activity-content {

flex: 1;

}

.activity-title {

font-weight: 500;

margin-bottom: 5px;

color: var(--primary-color);

}

.activity-time {

font-size: 12px;

color: var(--text-light);

}

/\* ===== SECTION HEADER ===== \*/

.section-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 30px;

}

.section-header h2 {

font-size: 24px;

font-weight: 700;

color: var(--primary-color);

}

/\* ===== BUTTONS ===== \*/

.btn {

display: inline-flex;

align-items: center;

justify-content: center;

padding: 12px 25px;

border: none;

border-radius: var(--border-radius);

font-weight: 600;

cursor: pointer;

transition: var(--transition);

gap: 8px;

text-decoration: none;

font-size: 1rem;

}

.btn-primary {

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: white;

box-shadow: 0 4px 10px rgba(180, 148, 52, 0.25);

}

.btn-primary:hover {

transform: translateY(-2px);

box-shadow: 0 6px 15px rgba(180, 148, 52, 0.35);

}

.btn-secondary {

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

color: var(--text-color);

}

.btn-secondary:hover {

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

}

.btn-outline {

background: transparent;

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

color: var(--secondary-color);

}

.btn-outline:hover {

background: var(--secondary-color);

color: white;

}

.btn-danger {

background: var(--danger-color);

color: white;

}

.btn-danger:hover {

background: #c82333;

}

.btn-sm {

padding: 8px 15px;

font-size: 14px;

}

.btn:disabled {

opacity: 0.6;

cursor: not-allowed;

}

/\* ===== FILTERS ===== \*/

.filters-bar {

display: flex;

gap: 15px;

margin-bottom: 20px;

padding: 20px;

background: var(--card-bg);

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.filter-group {

display: flex;

flex-direction: column;

gap: 5px;

}

.filter-group label {

font-size: 14px;

font-weight: 500;

color: var(--text-color);

}

.filter-group input,

.filter-group select {

padding: 10px 12px;

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

border-radius: var(--border-radius);

transition: var(--transition);

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

color: var(--text-color);

}

.filter-group input:focus,

.filter-group select:focus {

outline: none;

border-color: var(--secondary-color);

box-shadow: 0 0 0 2px rgba(180, 148, 52, 0.2);

}

/\* ===== TABLES ===== \*/

.table-container {

background: var(--card-bg);

border-radius: var(--border-radius);

overflow: hidden;

box-shadow: var(--card-shadow);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.data-table {

width: 100%;

border-collapse: collapse;

}

.data-table th {

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

padding: 15px;

text-align: left;

font-weight: 600;

color: var(--text-light);

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

}

.data-table td {

padding: 15px;

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

}

.data-table tr:last-child td {

border-bottom: none;

}

.data-table tr:hover {

background-color: rgba(180, 148, 52, 0.05);

}

/\* ===== PRODUCTS & EMPLOYEES GRID ===== \*/

.produtos-grid,

.funcionarios-grid {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));

gap: 20px;

}

.produto-card,

.funcionario-card {

background: var(--card-bg);

border-radius: var(--border-radius);

overflow: hidden;

box-shadow: var(--card-shadow);

transition: var(--transition);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.produto-card:hover,

.funcionario-card:hover {

transform: translateY(-5px);

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);

}

.produto-imagem {

height: 180px;

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

display: flex;

align-items: center;

justify-content: center;

color: var(--text-light);

font-size: 50px;

overflow: hidden;

}

.produto-imagem img {

width: 100%;

height: 100%;

object-fit: cover;

}

.produto-info,

.funcionario-info {

padding: 20px;

}

.produto-nome,

.funcionario-nome {

font-size: 18px;

font-weight: 600;

margin-bottom: 10px;

color: var(--primary-color);

}

.produto-detalhes,

.funcionario-detalhes {

color: var(--text-light);

margin-bottom: 15px;

font-size: 14px;

}

.produto-preco {

font-size: 20px;

font-weight: 700;

color: var(--primary-color);

margin-bottom: 15px;

}

.produto-acoes,

.funcionario-acoes {

display: flex;

gap: 10px;

}

.funcionario-avatar {

width: 80px;

height: 80px;

border-radius: 50%;

background: var(--primary-color);

display: flex;

align-items: center;

justify-content: center;

color: white;

font-size: 30px;

margin: 0 auto 15px;

}

/\* ===== SALES SUMMARY ===== \*/

.vendas-summary {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

gap: 20px;

margin-bottom: 30px;

}

.summary-card {

background: var(--card-bg);

border-radius: var(--border-radius);

padding: 20px;

text-align: center;

box-shadow: var(--card-shadow);

transition: var(--transition);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.summary-card:hover {

transform: translateY(-3px);

}

.summary-card h4 {

font-size: 14px;

color: var(--text-light);

margin-bottom: 10px;

}

.summary-value {

font-size: 24px;

font-weight: 700;

color: var(--primary-color);

}

/\* ===== REPORTS ===== \*/

.relatorios-grid {

display: grid;

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

gap: 20px;

}

.relatorio-card {

background: var(--card-bg);

border-radius: var(--border-radius);

padding: 30px;

text-align: center;

box-shadow: var(--card-shadow);

transition: var(--transition);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.relatorio-card:hover {

transform: translateY(-5px);

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);

}

.relatorio-card i {

font-size: 40px;

color: var(--secondary-color);

margin-bottom: 15px;

}

.relatorio-card h3 {

font-size: 18px;

margin-bottom: 10px;

color: var(--primary-color);

}

.relatorio-card p {

color: var(--text-light);

margin-bottom: 20px;

font-size: 14px;

}

/\* ===== MODALS ===== \*/

.modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

z-index: 1000;

align-items: center;

justify-content: center;

backdrop-filter: blur(3px);

}

.modal.active {

display: flex;

}

.modal-content {

background: var(--card-bg);

border-radius: var(--border-radius);

width: 100%;

max-width: 600px;

max-height: 90vh;

overflow-y: auto;

box-shadow: 0 10px 25px rgba(0, 0, 0, 0.2);

}

.modal-header {

padding: 20px;

display: flex;

justify-content: space-between;

align-items: center;

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

}

.modal-header h3 {

font-size: 20px;

font-weight: 600;

color: var(--primary-color);

}

.close-modal {

background: none;

border: none;

font-size: 24px;

cursor: pointer;

color: var(--text-light);

transition: var(--transition);

}

.close-modal:hover {

color: var(--text-color);

}

.modal-body {

padding: 20px;

background: var(--card-bg);

}

.form-row {

display: flex;

gap: 15px;

margin-bottom: 15px;

}

.form-group {

flex: 1;

display: flex;

flex-direction: column;

gap: 5px;

}

.form-group label {

font-weight: 500;

font-size: 14px;

color: var(--text-color);

}

.form-group input,

.form-group select,

.form-group textarea {

padding: 12px 15px;

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

border-radius: var(--border-radius);

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

transition: var(--transition);

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

color: var(--text-color);

}

.form-group input:focus,

.form-group select:focus,

.form-group textarea:focus {

outline: none;

border-color: var(--secondary-color);

box-shadow: 0 0 0 2px rgba(180, 148, 52, 0.2);

}

.modal-actions {

padding: 20px;

display: flex;

justify-content: flex-end;

gap: 10px;

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

background: var(--card-bg);

}

/\* ===== IMAGE UPLOAD ===== \*/

.image-upload {

display: flex;

flex-direction: column;

gap: 10px;

}

.image-preview {

width: 100%;

height: 200px;

border: 2px dashed var(--border-color);

border-radius: var(--border-radius);

display: flex;

align-items: center;

justify-content: center;

overflow: hidden;

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

}

.image-preview img {

max-width: 100%;

max-height: 100%;

object-fit: contain;

}

.image-preview .placeholder {

color: var(--text-light);

text-align: center;

padding: 20px;

}

.upload-actions {

display: flex;

gap: 10px;

}

.upload-btn {

flex: 1;

padding: 10px;

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

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

border-radius: var(--border-radius);

cursor: pointer;

text-align: center;

transition: var(--transition);

}

.upload-btn:hover {

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

}

.remove-image {

background-color: #fee2e2;

color: var(--danger-color);

border: 1px solid #fecaca;

display: flex;

align-items: center;

justify-content: center;

}

.remove-image:hover {

background-color: #fecaca;

}

/\* ===== STATUS BADGES ===== \*/

.status-badge {

padding: 5px 10px;

border-radius: 20px;

font-size: 12px;

font-weight: 500;

}

.status-pendente {

background-color: rgba(255, 193, 7, 0.1);

color: var(--warning-color);

}

.status-confirmado {

background-color: rgba(40, 167, 69, 0.1);

color: var(--success-color);

}

.status-realizado {

background-color: rgba(40, 167, 69, 0.1);

color: var(--success-color);

}

.status-cancelado {

background-color: rgba(220, 53, 69, 0.1);

color: var(--danger-color);

}

.status-ativo {

background-color: rgba(40, 167, 69, 0.1);

color: var(--success-color);

}

.status-inativo {

background-color: rgba(220, 53, 69, 0.1);

color: var(--danger-color);

}

/\* ===== AUTHENTICATION ===== \*/

.auth-container {

display: flex;

justify-content: center;

align-items: center;

width: 100%;

min-height: 100vh;

background: var(--primary-color);

}

.auth-card {

background: var(--card-bg);

border-radius: var(--border-radius);

padding: 40px;

width: 100%;

max-width: 400px;

box-shadow: var(--card-shadow);

position: relative;

}

.auth-logo {

text-align: center;

margin-bottom: 30px;

font-size: 40px;

color: var(--secondary-color);

}

.auth-title {

text-align: center;

margin-bottom: 30px;

font-size: 24px;

font-weight: 700;

color: var(--primary-color);

}

.auth-form {

display: flex;

flex-direction: column;

gap: 20px;

}

.form-group-auth {

display: flex;

flex-direction: column;

gap: 8px;

}

.form-group-auth label {

font-weight: 500;

font-size: 14px;

color: var(--text-color);

}

.form-group-auth input {

padding: 12px 15px;

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

border-radius: var(--border-radius);

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

transition: var(--transition);

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

color: var(--text-color);

}

.form-group-auth input:focus {

outline: none;

border-color: var(--secondary-color);

box-shadow: 0 0 0 2px rgba(180, 148, 52, 0.2);

}

.auth-btn {

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: white;

padding: 12px;

border: none;

border-radius: var(--border-radius);

font-weight: 600;

cursor: pointer;

transition: var(--transition);

display: flex;

align-items: center;

justify-content: center;

gap: 10px;

}

.auth-btn:hover:not(:disabled) {

transform: translateY(-2px);

box-shadow: 0 6px 15px rgba(180, 148, 52, 0.35);

}

.auth-error {

color: var(--danger-color);

text-align: center;

margin-top: 15px;

font-size: 14px;

display: none;

}

.auth-links {

text-align: center;

margin-top: 20px;

display: flex;

flex-direction: column;

gap: 10px;

}

.auth-link {

color: var(--secondary-color);

text-decoration: none;

cursor: pointer;

font-size: 14px;

transition: var(--transition);

font-weight: 500;

text-align: center;

}

.auth-link:hover {

text-decoration: underline;

}

/\* Estilos específicos para o formulário de cadastro \*/

.auth-links {

display: flex;

flex-direction: column;

gap: 10px;

margin-top: 20px;

}

.auth-link {

color: var(--secondary-color);

text-decoration: none;

cursor: pointer;

font-size: 14px;

transition: var(--transition);

font-weight: 500;

text-align: center;

}

.auth-link:hover {

text-decoration: underline;

}

/\* ===== RESPONSIVE DESIGN ===== \*/

/\* Sidebar mobile toggle \*/

.sidebar.collapsed {

transform: translateX(-100%);

}

.sidebar-overlay {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

z-index: 999;

}

.sidebar-overlay.active {

display: block;

}

/\* Tablet and small desktop \*/

@media (max-width: 1200px) {

.metrics-grid {

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

}

.charts-grid {

grid-template-columns: 1fr;

}

.search-box input {

width: 200px;

}

}

@media (max-width: 1024px) {

.sidebar {

width: 80px;

transition: all 0.3s ease;

}

.sidebar-header h3,

.nav-link span,

.user-details {

display: none;

}

.sidebar-nav {

display: flex;

justify-content: center;

}

.nav-link {

justify-content: center;

padding: 15px;

margin: 0 5px;

}

.nav-link i {

margin-right: 0;

}

.search-box input {

width: 150px;

}

.activities-section {

grid-template-columns: 1fr;

}

.produtos-grid,

.funcionarios-grid {

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

}

}

/\* Mobile landscape and tablet portrait \*/

@media (max-width: 768px) {

html, body {

overflow: auto;

}

body {

flex-direction: column;

}

.sidebar {

position: fixed;

top: 0;

left: 0;

width: 280px;

height: 100vh;

z-index: 1000;

transform: translateX(-100%);

transition: transform 0.3s ease;

}

.sidebar.active {

transform: translateX(0);

}

.sidebar-header h3,

.nav-link span,

.user-details {

display: block;

}

.nav-link {

justify-content: flex-start;

padding: 12px 20px;

margin: 0 10px;

}

.nav-link i {

margin-right: 10px;

}

.main-content {

width: 100%;

margin-left: 0;

}

.main-header {

padding: 10px 20px;

}

.header-left {

flex: 1;

}

.sidebar-toggle {

display: flex;

}

.header-right {

gap: 10px;

}

.search-box {

display: none;

}

.date-time {

display: none;

}

.metrics-grid {

grid-template-columns: 1fr;

gap: 15px;

}

.metric-card {

padding: 20px;

}

.activities-section {

grid-template-columns: 1fr;

}

.filters-bar {

flex-direction: column;

gap: 10px;

padding: 15px;

}

.filter-group {

width: 100%;

}

.form-row {

flex-direction: column;

gap: 0;

}

.upload-actions {

flex-direction: column;

}

.modal-content {

width: 95%;

margin: 20px auto;

max-height: calc(100vh - 40px);

}

.produtos-grid,

.funcionarios-grid {

grid-template-columns: 1fr;

}

.vendas-summary {

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

}

.relatorios-grid {

grid-template-columns: 1fr;

}

.table-container {

overflow-x: auto;

}

.data-table {

min-width: 600px;

}

.data-table th,

.data-table td {

padding: 10px 8px;

font-size: 14px;

}

}

/\* Mobile portrait \*/

@media (max-width: 480px) {

.content-area {

padding: 15px;

}

.main-header {

padding: 10px 15px;

}

.page-title {

font-size: 20px;

}

.section-header {

flex-direction: column;

align-items: flex-start;

gap: 15px;

}

.section-header h2 {

font-size: 20px;

}

.metric-card {

padding: 15px;

flex-direction: column;

text-align: center;

}

.metric-icon {

margin-right: 0;

margin-bottom: 10px;

}

.chart-card {

padding: 15px;

}

.chart-container {

height: 250px;

}

.activities-card {

padding: 15px;

}

.modal-content {

width: calc(100% - 20px);

margin: 10px;

border-radius: 10px;

}

.modal-header,

.modal-body,

.modal-actions {

padding: 15px;

}

.auth-card {

padding: 25px 20px;

margin: 20px;

}

.auth-title {

font-size: 20px;

}

.btn {

padding: 10px 20px;

font-size: 14px;

}

.btn-sm {

padding: 6px 12px;

font-size: 12px;

}

.vendas-summary {

grid-template-columns: 1fr;

}

}

/\* Responsividade para o formulário de cadastro \*/

@media (max-width: 480px) {

.auth-card {

padding: 20px 15px;

margin: 10px;

}

.auth-title {

font-size: 20px;

}

}

/\* Dark mode overrides \*/

@media (prefers-color-scheme: dark) {

.metric-card,

.chart-card,

.activities-card,

.produto-card,

.funcionario-card,

.summary-card,

.relatorio-card,

.filters-bar,

.table-container,

.modal-content {

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

}

.data-table th {

background-color: rgba(255, 255, 255, 0.05);

}

.metric-card:hover,

.chart-card:hover,

.activities-card:hover,

.produto-card:hover,

.funcionario-card:hover,

.summary-card:hover,

.relatorio-card:hover {

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);

}

.search-box input,

.filter-group input,

.filter-group select,

.form-group input,

.form-group select,

.form-group textarea,

.form-group-auth input {

background-color: rgba(255, 255, 255, 0.05);

border-color: rgba(255, 255, 255, 0.1);

}

}

/\* Print styles \*/

@media print {

.sidebar,

.main-header,

.btn,

.filters-bar,

.modal,

.sidebar-toggle,

.notification-btn {

display: none !important;

}

.main-content {

margin-left: 0;

width: 100%;

}

.content-area {

padding: 0;

overflow: visible;

}

.content-section {

display: block !important;

}

.metric-card,

.chart-card,

.activities-card,

.produto-card,

.funcionario-card,

.summary-card,

.relatorio-card,

.table-container {

box-shadow: none;

border: 1px solid #ddd;

break-inside: avoid;

}

.metric-card:hover,

.chart-card:hover,

.activities-card:hover,

.produto-card:hover,

.funcionario-card:hover,

.summary-card:hover,

.relatorio-card:hover {

transform: none;

box-shadow: none;

}

}

# Arquivo: public/css/auth.css

/\* auth.css - Estilos unificados para autenticação - Atualizado 2025 \*/

:root {

--primary-color: #213f57;

--secondary-color: #b49434;

--accent-color: #d4b15f;

--text-color: #333;

--text-light: #777;

--border-color: #e0e0e0;

--success-color: #28a745;

--warning-color: #ffc107;

--danger-color: #dc3545;

--transition: all 0.3s ease;

--box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

--border-radius: 10px;

--card-shadow: 0 8px 25px rgba(0, 0, 0, 0.08);

}

/\* Dark mode variables \*/

@media (prefers-color-scheme: dark) {

:root {

--primary-color: #263f54;

--secondary-color: #b49434;

--accent-color: #d4b15f;

--text-color: #f0f0f0;

--text-light: #b0b0b0;

--border-color: #444;

--box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);

--card-shadow: 0 8px 25px rgba(0, 0, 0, 0.2);

}

}

/\* Automatic background colors based on mode \*/

@media (prefers-color-scheme: light) {

:root {

--bg-color: #f9f9f9;

--light-color: #f8f9fa;

--dark-color: #343a40;

--card-bg: #ffffff;

}

}

@media (prefers-color-scheme: dark) {

:root {

--bg-color: #1a1a1a;

--light-color: #2a2a2a;

--dark-color: #f8f9fa;

--card-bg: #2a2a2a;

}

}

.user-dropdown {

position: relative;

cursor: pointer;

}

.user-icon {

display: flex;

align-items: center;

gap: 8px;

padding: 10px 15px;

border-radius: 8px;

transition: var(--transition);

color: var(--text-color);

background: rgba(180, 148, 52, 0.1);

border: 1px solid rgba(180, 148, 52, 0.2);

}

.user-icon:hover {

background: rgba(180, 148, 52, 0.2);

transform: translateY(-2px);

box-shadow: var(--box-shadow);

}

.user-icon i {

font-size: 1.2rem;

color: var(--secondary-color);

}

.login-status {

font-weight: 600;

font-size: 0.95rem;

}

.user-dropdown.logged-in .login-status {

color: var(--success-color);

}

.dropdown-menu {

position: absolute;

top: 100%;

right: 0;

width: 280px;

background: var(--card-bg);

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

padding: 15px 0;

margin-top: 10px;

opacity: 0;

visibility: hidden;

transform: translateY(-10px);

transition: var(--transition);

z-index: 1000;

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

}

.user-dropdown:hover .dropdown-menu {

opacity: 1;

visibility: visible;

transform: translateY(0);

}

.user-info {

padding: 0 20px 15px;

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

margin-bottom: 15px;

}

.user-info strong {

display: block;

color: var(--primary-color);

font-size: 1rem;

font-weight: 600;

margin-bottom: 5px;

}

.user-info span {

color: var(--text-light);

font-size: 0.85rem;

}

.dropdown-item {

display: flex;

align-items: center;

gap: 12px;

padding: 12px 20px;

color: var(--text-color);

text-decoration: none;

transition: var(--transition);

font-size: 0.95rem;

font-weight: 500;

}

.dropdown-item:hover {

color: var(--secondary-color);

background: rgba(180, 148, 52, 0.1);

padding-left: 25px;

}

.dropdown-item i {

width: 18px;

text-align: center;

color: var(--secondary-color);

}

.dropdown-divider {

height: 1px;

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

margin: 10px 0;

}

/\* Modal styles \*/

.modal {

display: none;

position: fixed;

z-index: 1000;

left: 0;

top: 0;

width: 100%;

height: 100%;

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

backdrop-filter: blur(5px);

animation: fadeIn 0.3s ease;

}

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

.modal-content {

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

margin: 5% auto;

padding: 0;

border-radius: var(--border-radius);

width: 90%;

max-width: 450px;

position: relative;

animation: modalSlideIn 0.4s ease;

box-shadow: var(--card-shadow);

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

overflow: hidden;

}

@keyframes modalSlideIn {

from {

opacity: 0;

transform: translateY(-50px) scale(0.95);

}

to {

opacity: 1;

transform: translateY(0) scale(1);

}

}

.close-modal {

position: absolute;

right: 20px;

top: 20px;

font-size: 1.5rem;

cursor: pointer;

color: var(--text-light);

transition: var(--transition);

z-index: 10;

background: none;

border: none;

}

.close-modal:hover {

color: var(--secondary-color);

transform: rotate(90deg);

}

.modal-header {

padding: 30px 30px 20px;

text-align: center;

background: linear-gradient(135deg, rgba(33, 63, 87, 0.05) 0%, rgba(180, 148, 52, 0.1) 100%);

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

}

.modal-header h2 {

margin: 0 0 8px 0;

color: var(--primary-color);

font-size: 1.8rem;

font-weight: 700;

}

.modal-header p {

margin: 0;

color: var(--text-light);

font-size: 0.95rem;

}

.login-form {

padding: 25px 30px 30px;

}

.form-group {

margin-bottom: 20px;

position: relative;

}

.form-group label {

display: block;

margin-bottom: 8px;

color: var(--text-color);

font-weight: 500;

font-size: 0.95rem;

}

.form-group input {

width: 100%;

padding: 14px 18px;

border: 2px solid var(--border-color);

border-radius: var(--border-radius);

font-size: 1rem;

transition: var(--transition);

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

color: var(--text-color);

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

}

.form-group input:focus {

outline: none;

border-color: var(--secondary-color);

box-shadow: 0 0 0 4px rgba(180, 148, 52, 0.15);

}

.password-input {

position: relative;

}

.toggle-password {

position: absolute;

right: 15px;

top: 50%;

transform: translateY(-50%);

background: none;

border: none;

color: var(--text-light);

cursor: pointer;

padding: 4px;

transition: var(--transition);

}

.toggle-password:hover {

color: var(--secondary-color);

}

.form-options {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 20px;

font-size: 0.9rem;

}

.remember-me {

display: flex;

align-items: center;

gap: 8px;

color: var(--text-light);

cursor: pointer;

}

.remember-me input[type="checkbox"] {

width: 18px;

height: 18px;

accent-color: var(--secondary-color);

}

.forgot-password {

color: var(--secondary-color);

text-decoration: none;

font-weight: 500;

transition: var(--transition);

}

.forgot-password:hover {

color: var(--accent-color);

text-decoration: underline;

}

.btn-block {

width: 100%;

padding: 14px;

font-size: 1rem;

font-weight: 600;

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: white;

border: none;

border-radius: var(--border-radius);

cursor: pointer;

transition: var(--transition);

box-shadow: 0 4px 10px rgba(180, 148, 52, 0.25);

}

.btn-block:hover {

transform: translateY(-2px);

box-shadow: 0 6px 15px rgba(180, 148, 52, 0.35);

}

.btn-block.loading {

opacity: 0.8;

pointer-events: none;

position: relative;

}

.btn-block.loading::after {

content: '';

display: inline-block;

width: 16px;

height: 16px;

border: 2px solid transparent;

border-top: 2px solid #ffffff;

border-radius: 50%;

animation: spin 1s linear infinite;

margin-left: 8px;

vertical-align: middle;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

.divider {

display: flex;

align-items: center;

margin: 25px 0;

color: var(--text-light);

font-size: 0.9rem;

}

.divider::before,

.divider::after {

content: '';

flex: 1;

height: 1px;

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

}

.divider span {

padding: 0 15px;

font-size: 0.85rem;

}

.social-login {

padding: 0 30px 20px;

}

.social-btn {

width: 100%;

padding: 12px;

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

border-radius: var(--border-radius);

background: var(--card-bg);

color: var(--text-color);

font-weight: 500;

cursor: pointer;

transition: var(--transition);

display: flex;

align-items: center;

justify-content: center;

gap: 8px;

margin-bottom: 10px;

}

.social-btn:hover {

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

transform: translateY(-2px);

box-shadow: var(--box-shadow);

}

.social-btn.google {

border-color: #DB4437;

color: #DB4437;

}

.social-btn.google:hover {

background-color: rgba(219, 68, 55, 0.1);

}

.social-btn.facebook {

border-color: #4267B2;

color: #4267B2;

}

.social-btn.facebook:hover {

background-color: rgba(66, 103, 178, 0.1);

}

.register-link {

text-align: center;

padding: 0 30px 30px;

color: var(--text-light);

font-size: 0.95rem;

}

.register-link a {

color: var(--secondary-color);

text-decoration: none;

font-weight: 600;

transition: var(--transition);

}

.register-link a:hover {

color: var(--accent-color);

text-decoration: underline;

}

/\* Error states \*/

.error-message {

color: var(--danger-color);

font-size: 0.85rem;

margin-top: 6px;

display: none;

padding-left: 5px;

}

.form-group.error input {

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

box-shadow: 0 0 0 4px rgba(220, 53, 69, 0.15);

}

.form-group.error .error-message {

display: block;

}

/\* Success states \*/

.form-group.success input {

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

box-shadow: 0 0 0 4px rgba(40, 167, 69, 0.15);

}

/\* Loading state \*/

.loading-spinner {

display: inline-block;

width: 16px;

height: 16px;

border: 2px solid transparent;

border-top: 2px solid var(--secondary-color);

border-radius: 50%;

animation: spin 1s linear infinite;

margin-left: 8px;

}

/\* Responsive \*/

@media (max-width: 768px) {

.modal-content {

margin: 10% auto;

width: 95%;

max-width: none;

}

.dropdown-menu {

width: 250px;

right: -10px;

}

.form-options {

flex-direction: column;

gap: 15px;

align-items: flex-start;

}

.modal-header {

padding: 25px 20px 15px;

}

.login-form {

padding: 20px 25px 25px;

}

.social-login {

padding: 0 25px 15px;

}

.register-link {

padding: 0 25px 25px;

}

}

@media (max-width: 480px) {

.user-icon .login-status {

display: none;

}

.dropdown-menu {

width: 220px;

}

.modal-header h2 {

font-size: 1.5rem;

}

.form-group input {

padding: 12px 15px;

}

}

.alert {

padding: 12px;

margin: 10px 0;

border-radius: 6px;

font-size: 14px;

font-weight: 500;

text-align: center;

}

.alert.error {

background: #f8d7da;

color: #721c24;

border: 1px solid #f5c6cb;

}

.alert.success {

background: #d4edda;

color: #155724;

border: 1px solid #c3e6cb;

}

.page-header {

display: flex;

align-items: center;

gap: 15px;

padding: 1rem;

background: #f8f9fa;

border-bottom: 1px solid #dee2e6;

}

.back-btn {

background: #6c757d;

color: white;

border: none;

padding: 8px 16px;

border-radius: 5px;

cursor: pointer;

font-size: 14px;

transition: background 0.3s;

}

.back-btn:hover {

background: #545b62;

}

# Arquivo: public/css/login.css

:root {

--primary-color: #213f57;

--secondary-color: #b49434;

--accent-color: #d4b15f;

--text-color: #333;

--text-light: #777;

--border-color: #e0e0e0;

--success-color: #28a745;

--warning-color: #ffc107;

--danger-color: #dc3545;

--transition: all 0.3s ease;

--box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

--border-radius: 12px;

--card-shadow: 0 8px 25px rgba(0, 0, 0, 0.08);

--gradient: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

}

/\* Dark mode variables \*/

@media (prefers-color-scheme: dark) {

:root {

--primary-color: #2a4d6e;

--secondary-color: #c6a64b;

--accent-color: #e0c67c;

--text-color: #f5f5f5;

--text-light: #a0a0a0;

--border-color: #444;

--box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);

--card-shadow: 0 8px 25px rgba(0, 0, 0, 0.2);

}

}

/\* Automatic background colors based on mode \*/

@media (prefers-color-scheme: light) {

:root {

--bg-color: #f8f9fa;

--light-color: #ffffff;

--dark-color: #343a40;

--card-bg: #ffffff;

}

}

@media (prefers-color-scheme: dark) {

:root {

--bg-color: #1a1a1a;

--light-color: #2a2a2a;

--dark-color: #f8f9fa;

--card-bg: #2a2a2a;

}

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

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

color: var(--text-color);

line-height: 1.6;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

padding: 20px;

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

font-weight: 400;

}

.login-container {

width: 100%;

max-width: 450px;

}

.login-card {

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

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

padding: 40px 35px;

border: 1px solid rgba(180, 148, 52, 0.1);

transition: var(--transition);

position: relative;

overflow: hidden;

}

.login-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: var(--gradient);

}

.login-card:hover {

box-shadow: 0 12px 30px rgba(0, 0, 0, 0.12);

transform: translateY(-5px);

}

.logo-container {

text-align: center;

margin-bottom: 20px;

}

.logo {

width: 60px;

height: 60px;

margin: 0 auto 15px;

background: var(--gradient);

color: white;

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

font-size: 24px;

font-weight: 700;

box-shadow: 0 4px 10px rgba(180, 148, 52, 0.3);

}

.login-title {

font-size: 1.8rem;

font-weight: 600;

color: var(--primary-color);

margin-bottom: 8px;

text-align: center;

}

.login-subtitle {

font-size: 1rem;

color: var(--text-light);

text-align: center;

margin-bottom: 25px;

line-height: 1.5;

}

.divider {

height: 1px;

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

margin: 25px 0;

position: relative;

}

.form-group {

margin-bottom: 20px;

position: relative;

}

.section-title {

display: block;

margin-bottom: 8px;

font-weight: 500;

color: var(--text-color);

font-size: 0.95rem;

}

.input-field {

width: 100%;

padding: 14px 16px;

border: 2px solid var(--border-color);

border-radius: var(--border-radius);

font-size: 1rem;

transition: var(--transition);

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

color: var(--text-color);

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

}

.input-field:focus {

outline: none;

border-color: var(--secondary-color);

box-shadow: 0 0 0 4px rgba(180, 148, 52, 0.15);

}

.field-hint {

display: block;

margin-top: 6px;

font-size: 0.8rem;

color: var(--text-light);

}

.password-options {

display: flex;

justify-content: space-between;

align-items: center;

font-size: 0.9rem;

color: var(--text-light);

margin-bottom: 25px;

}

.checkbox-container {

display: flex;

align-items: center;

gap: 10px;

cursor: pointer;

font-weight: 400;

position: relative;

padding-left: 30px;

}

.checkbox-container input[type="checkbox"] {

position: absolute;

opacity: 0;

cursor: pointer;

}

.checkmark {

position: absolute;

left: 0;

height: 20px;

width: 20px;

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

border: 2px solid var(--border-color);

border-radius: 5px;

transition: var(--transition);

}

.checkbox-container input[type="checkbox"]:checked ~ .checkmark {

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

border-color: var(--secondary-color);

}

.checkmark:after {

content: "";

position: absolute;

display: none;

}

.checkbox-container input[type="checkbox"]:checked ~ .checkmark:after {

display: block;

}

.checkbox-container .checkmark:after {

left: 6px;

top: 2px;

width: 5px;

height: 10px;

border: solid white;

border-width: 0 2px 2px 0;

transform: rotate(45deg);

}

.link {

color: var(--secondary-color);

text-decoration: none;

font-weight: 500;

transition: var(--transition);

}

.link:hover {

color: var(--accent-color);

text-decoration: underline;

}

.forgot-password {

color: var(--secondary-color);

text-decoration: none;

font-weight: 500;

transition: var(--transition);

font-size: 0.9rem;

}

.forgot-password:hover {

color: var(--accent-color);

text-decoration: underline;

}

.login-button {

width: 100%;

background: var(--gradient);

color: white;

border: none;

padding: 16px;

border-radius: var(--border-radius);

font-weight: 600;

font-size: 1rem;

cursor: pointer;

margin-bottom: 25px;

transition: var(--transition);

box-shadow: 0 4px 10px rgba(180, 148, 52, 0.25);

position: relative;

overflow: hidden;

}

.login-button::before {

content: '';

position: absolute;

top: 0;

left: -100%;

width: 100%;

height: 100%;

background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.2), transparent);

transition: 0.5s;

}

.login-button:hover::before {

left: 100%;

}

.login-button:hover {

transform: translateY(-3px);

box-shadow: 0 6px 15px rgba(180, 148, 52, 0.35);

}

.login-button:active {

transform: translateY(0);

}

.button-text {

position: relative;

z-index: 1;

}

.login-button.loading {

opacity: 0.7;

pointer-events: none;

}

.login-button.loading::after {

content: '';

display: inline-block;

width: 16px;

height: 16px;

border: 2px solid #ffffff;

border-radius: 50%;

border-top-color: transparent;

animation: spin 1s ease-in-out infinite;

margin-left: 8px;

}

.register-text {

text-align: center;

color: var(--text-light);

font-size: 0.95rem;

}

.register-link {

color: var(--secondary-color);

text-decoration: none;

font-weight: 500;

transition: var(--transition);

}

.register-link:hover {

color: var(--accent-color);

text-decoration: underline;

}

.password-match-error {

color: var(--danger-color);

font-size: 0.8rem;

margin-top: 8px;

display: none;

padding-left: 5px;

}

.form-group.error .error-message {

display: block;

}

/\* Loading animation \*/

@keyframes spin {

to {

transform: rotate(360deg);

}

}

/\* Responsividade \*/

@media (max-width: 768px) {

.login-card {

padding: 30px 25px;

}

.login-title {

font-size: 1.6rem;

}

.password-options {

flex-direction: column;

gap: 15px;

align-items: flex-start;

}

}

@media (max-width: 480px) {

body {

padding: 15px;

}

.login-card {

padding: 25px 20px;

border: none;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

}

.login-title {

font-size: 1.5rem;

}

.input-field {

padding: 12px 16px;

}

.login-button {

padding: 14px;

}

.logo {

width: 50px;

height: 50px;

font-size: 20px;

}

}

.alert {

padding: 12px;

margin: 10px 0;

border-radius: 6px;

font-size: 14px;

font-weight: 500;

text-align: center;

}

.alert.error {

background: #f8d7da;

color: #721c24;

border: 1px solid #f5c6cb;

}

.alert.success {

background: #d4edda;

color: #155724;

border: 1px solid #c3e6cb;

}

/\* === BOTÕES PADRÃO === \*/

.button-group {

display: flex;

justify-content: space-between;

gap: 12px;

margin-top: 20px;

}

.btn {

padding: 14px 22px;

border-radius: 50px;

font-weight: 600;

font-size: 1rem;

border: none;

cursor: pointer;

transition: all 0.3s ease;

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

}

/\* Principal \*/

.btn-primary {

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: #fff;

box-shadow: 0 4px 12px rgba(180, 148, 52, 0.3);

}

.btn-primary:hover {

transform: translateY(-2px);

box-shadow: 0 6px 18px rgba(180, 148, 52, 0.4);

}

/\* Secundário \*/

.btn-secondary {

background: #d4b15f;

color: var(--text-color);

border: 2px solid var(--border-color);

}

.btn-secondary:hover {

background: #213f57;

transform: translateY(-2px);

}

/\* Estilos para o indicador de carregamento \*/

.loading-overlay {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

display: flex;

justify-content: center;

align-items: center;

z-index: 1000;

display: none;

}

.loading-spinner {

width: 50px;

height: 50px;

border: 5px solid #f3f3f3;

border-top: 5px solid #3498db;

border-radius: 50%;

animation: spin 1s linear infinite;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

.btn:disabled {

opacity: 0.6;

cursor: not-allowed;

}

# Arquivo: public/css/main.css

:root {

--primary-color: #213f57;

--testego: #384d5e;

--secondary-color: #b49434;

--accent-color: #d4b15f;

--text-color: #333;

--text-light: #777;

--border-color: #e0e0e0;

--success-color: #28a745;

--warning-color: #ffc107;

--danger-color: #dc3545;

--transition: all 0.3s ease;

--box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

--border-radius: 10px;

--card-shadow: 0 8px 25px rgba(0, 0, 0, 0.08);

}

/\* Dark mode variables - will override light mode when preferred \*/

@media (prefers-color-scheme: dark) {

:root {

--primary-color: #263f54;

--secondary-color: #b49434;

--accent-color: #d4b15f;

--text-color: #f0f0f0;

--text-light: #b0b0b0;

--border-color: #444;

--box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);

--card-shadow: 0 8px 25px rgba(0, 0, 0, 0.2);

}

}

/\* Base styles that work for both modes \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

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

color: var(--text-color);

line-height: 1.6;

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

}

/\* Automatic background colors based on mode \*/

@media (prefers-color-scheme: light) {

body {

--bg-color: #f9f9f9;

--light-color: #f8f9fa;

--dark-color: #343a40;

--f8f9fa: #dceeff;

--card-bg: #ffffff;

}

}

@media (prefers-color-scheme: dark) {

body {

--bg-color: #1a1a1a;

--light-color: #2a2a2a;

--dark-color: #f8f9fa;

--f8f9fa: #b6b6b6;

--card-bg: #2a2a2a;

}

}

.container {

width: 90%;

max-width: 1200px;

margin: 0 auto;

padding: 0 15px;

}

/\* ===== COMPONENTES REUTILIZÁVEIS ===== \*/

.btn {

display: inline-block;

padding: 12px 25px;

border-radius: 6px;

font-weight: 600;

transition: var(--transition);

cursor: pointer;

border: none;

font-size: 1rem;

text-decoration: none;

text-align: center;

letter-spacing: 0.5px;

}

.btn-primary {

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: white;

box-shadow: 0 4px 10px rgba(180, 148, 52, 0.25);

}

.btn-primary:hover {

transform: translateY(-2px);

box-shadow: 0 6px 15px rgba(180, 148, 52, 0.35);

}

.btn-secondary {

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

color: var(--text-color);

}

.btn-secondary:hover {

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

}

.btn-outline {

background: transparent;

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

color: var(--secondary-color);

padding: 8px 15px;

border-radius: 6px;

font-size: 0.9rem;

transition: var(--transition);

cursor: pointer;

display: flex;

align-items: center;

gap: 8px;

}

.btn-outline:hover {

background: var(--secondary-color);

color: white;

}

.btn-block {

width: 100%;

padding: 14px;

font-size: 1rem;

}

.section-title {

font-size: 2rem;

color: var(--primary-color);

margin-bottom: 30px;

position: relative;

padding-bottom: 15px;

font-weight: 700;

}

.section-title::after {

content: '';

position: absolute;

bottom: 0;

left: 0;

width: 60px;

height: 4px;

background: linear-gradient(90deg, var(--secondary-color), var(--accent-color));

border-radius: 2px;

}

/\* ===== HEADER MODERNIZADO 2025 ===== \*/

.header {

background: rgba(56, 77, 94, 0.98);

position: fixed;

top: 0;

width: 100%;

z-index: 1000;

height: 85px;

display: flex;

align-items: center;

backdrop-filter: blur(10px);

box-shadow: 0 4px 20px rgba(0, 0, 0, 0.15);

border-bottom: 1px solid rgba(255, 255, 255, 0.1);

transition: all 0.3s ease;

}

.header.scrolled {

height: 75px;

background: rgba(56, 77, 94, 0.98);

box-shadow: 0 6px 25px rgba(0, 0, 0, 0.15);

}

.navbar {

display: flex;

align-items: center;

justify-content: space-between;

padding: 15px 0;

width: 100%;

}

.logo {

position: relative;

z-index: 1002;

}

.logo img {

height: 130px;

transition: all 0.3s ease;

filter: drop-shadow(0 2px 6px rgba(0, 0, 0, 0.25));

}

.logo img:hover {

transform: scale(1.05);

}

.nav\_\_list {

display: flex;

list-style: none;

gap: 8px;

position: relative;

}

.nav\_\_list a {

text-decoration: none;

color: var(--f8f9fa);

font-weight: 500;

transition: all 0.3s ease;

position: relative;

padding: 12px 18px;

letter-spacing: 0.5px;

border-radius: 10px;

overflow: hidden;

font-size: 0.95rem;

}

.nav\_\_list a::before {

content: '';

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: linear-gradient(135deg, rgba(180, 148, 52, 0.1), rgba(212, 177, 95, 0.15));

opacity: 0;

transition: opacity 0.3s ease;

z-index: -1;

border-radius: 10px;

}

.nav\_\_list a::after {

content: '';

position: absolute;

bottom: 0;

left: 50%;

transform: translateX(-50%);

width: 0;

height: 2px;

background: linear-gradient(90deg, var(--secondary-color), var(--accent-color));

transition: width 0.3s ease;

}

.nav\_\_list a:hover::before,

.nav\_\_list a.active::before {

opacity: 1;

}

.nav\_\_list a:hover::after,

.nav\_\_list a.active::after {

width: 70%;

}

.nav\_\_list a.active {

color: var(--accent-color);

font-weight: 600;

}

.nav\_\_list a:hover {

color: var(--accent-color);

transform: translateY(-1px);

}

.action-icons {

display: flex;

align-items: center;

gap: 15px;

}

.user-icon {

color: var(--f8f9fa);

font-size: 1.2rem;

transition: var(--transition);

}

.user-icon:hover {

color: var(--secondary-color);

}

/\* User Dropdown \*/

.user-dropdown {

position: relative;

cursor: pointer;

}

.user-dropdown .user-icon {

display: flex;

align-items: center;

gap: 8px;

padding: 8px 12px;

border-radius: 20px;

background-color: rgba(255, 255, 255, 0.1);

transition: background-color 0.3s ease;

}

.user-dropdown .user-icon:hover {

background-color: rgba(255, 255, 255, 0.2);

}

.user-dropdown .user-icon i {

font-size: 1rem;

}

.user-dropdown .login-status {

color: white;

font-size: 0.9rem;

font-weight: 500;

}

.dropdown-menu {

position: absolute;

top: 100%;

right: 0;

background-color: white;

border-radius: 8px;

box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);

min-width: 220px;

opacity: 0;

visibility: hidden;

transform: translateY(10px);

transition: opacity 0.3s ease, transform 0.3s ease, visibility 0.3s;

z-index: 1000;

padding: 10px 0;

}

.dropdown-menu.show {

opacity: 1;

visibility: visible;

transform: translateY(0);

}

.user-info {

padding: 10px 20px;

border-bottom: 1px solid #eee;

margin-bottom: 10px;

}

.user-info strong {

display: block;

font-size: 1rem;

color: var(--primary-color);

margin-bottom: 2px;

}

.user-info span {

font-size: 0.85rem;

color: var(--text-light);

}

.dropdown-item {

display: flex;

align-items: center;

gap: 10px;

padding: 10px 20px;

color: var(--text-color);

text-decoration: none;

transition: background-color 0.2s ease;

}

.dropdown-item i {

color: var(--secondary-color);

}

.dropdown-item:hover {

background-color: #f5f5f5;

}

.dropdown-divider {

height: 1px;

background-color: #eee;

margin: 5px 0;

}

@media (max-width: 1024px) {

.action-icons {

margin-top: 0;

justify-content: flex-start;

width: auto;

}

.user-dropdown {

position: static;

}

.dropdown-menu {

position: fixed !important;

top: 85px !important;

left: 0 !important;

width: 100vw !important;

height: auto !important;

max-width: none !important;

border-radius: 0 !important;

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1) !important;

border-top: 1px solid #eee !important;

padding: 0 !important;

margin: 0 !important;

background-color: white !important;

z-index: 9999 !important;

overflow: visible !important;

}

.dropdown-menu.show {

transform: translateY(0) !important;

display: block !important;

}

.user-info {

width: 100% !important;

height: auto !important;

min-height: 100px !important;

padding: 25px !important;

margin: 0 !important;

border-bottom: 2px solid #ddd !important;

background: linear-gradient(135deg, #f8f9fa 0%, #e9ecef 100%) !important;

box-sizing: border-box !important;

overflow: visible !important;

}

.user-info strong {

width: 100% !important;

height: auto !important;

font-size: 1.3rem !important;

color: #213f57 !important;

margin: 0 0 10px 0 !important;

padding: 0 !important;

display: block !important;

font-weight: 700 !important;

line-height: 1.5 !important;

text-align: left !important;

overflow: visible !important;

white-space: normal !important;

word-break: break-word !important;

}

.user-info span {

width: 100% !important;

height: auto !important;

font-size: 1.1rem !important;

color: #6c757d !important;

margin: 0 !important;

padding: 0 !important;

display: block !important;

line-height: 1.5 !important;

text-align: left !important;

overflow: visible !important;

white-space: normal !important;

word-break: break-word !important;

}

.dropdown-item {

width: 100% !important;

padding: 20px 25px !important;

font-size: 1.1rem !important;

border-bottom: 1px solid #f0f0f0 !important;

display: flex !important;

align-items: center !important;

gap: 15px !important;

box-sizing: border-box !important;

}

.dropdown-item:last-child {

border-bottom: none !important;

}

.dropdown-item i {

font-size: 1.2rem !important;

width: 25px !important;

text-align: center !important;

flex-shrink: 0 !important;

}

}

@media (max-width: 480px) {

.dropdown-menu {

top: 75px !important;

}

.user-info {

padding: 20px !important;

min-height: 90px !important;

}

.user-info strong {

font-size: 1.2rem !important;

}

.user-info span {

font-size: 1rem !important;

}

.dropdown-item {

padding: 18px 20px !important;

font-size: 1rem !important;

}

}

/\* Força absoluta para os elementos específicos \*/

#user-display-name {

display: block !important;

visibility: visible !important;

opacity: 1 !important;

position: static !important;

z-index: auto !important;

width: 100% !important;

height: auto !important;

overflow: visible !important;

text-overflow: clip !important;

white-space: normal !important;

}

#user-email {

display: block !important;

visibility: visible !important;

opacity: 1 !important;

position: static !important;

z-index: auto !important;

width: 100% !important;

height: auto !important;

overflow: visible !important;

text-overflow: clip !important;

white-space: normal !important;

}

/\* Menu Hamburger Moderno \*/

.hamburger {

display: none;

flex-direction: column;

justify-content: space-between;

width: 30px;

height: 22px;

background: transparent;

border: none;

cursor: pointer;

padding: 0;

z-index: 1001;

position: relative;

}

.hamburger-line {

width: 100%;

height: 3px;

background: var(--f8f9fa);

border-radius: 3px;

transition: all 0.3s ease;

transform-origin: center;

}

.hamburger.active .hamburger-line:nth-child(1) {

transform: translateY(9px) rotate(45deg);

}

.hamburger.active .hamburger-line:nth-child(2) {

opacity: 0;

}

.hamburger.active .hamburger-line:nth-child(3) {

transform: translateY(-9px) rotate(-45deg);

}

/\* Responsividade - Menu Mobile \*/

@media (max-width: 1024px) {

.hamburger {

display: flex;

}

.nav {

position: fixed;

top: 0;

left: -100%;

width: 80%;

max-width: 320px;

height: 100vh;

background: rgba(56, 77, 94, 0.98);

backdrop-filter: blur(15px);

transition: all 0.4s ease;

z-index: 999;

box-shadow: 10px 0 30px rgba(0, 0, 0, 0.2);

padding: 100px 25px 40px;

display: flex;

flex-direction: column;

justify-content: flex-start;

}

.nav.active {

left: 0;

}

.nav\_\_list {

flex-direction: column;

gap: 5px;

width: 100%;

}

.nav\_\_list a {

padding: 16px 20px;

font-size: 1.1rem;

border-radius: 10px;

margin-bottom: 5px;

width: 100%;

display: block;

background: rgba(255, 255, 255, 0.05);

border: 1px solid rgba(255, 255, 255, 0.08);

}

.nav\_\_list a:hover {

background: rgba(180, 148, 52, 0.15);

transform: translateX(5px);

}

.nav\_\_list a::after {

display: none;

}

.action-icons {

margin-top: 30px;

justify-content: center;

}

}

/\* Ajuste para telas menores \*/

@media (max-width: 480px) {

.header {

height: 75px;

}

.header.scrolled {

height: 65px;

}

.logo img {

height: 65px;

}

.nav {

width: 85%;

padding: 90px 20px 30px;

}

}

/\* Pequeno ajuste no conteúdo principal para o novo header \*/

.main-content {

padding-top: 95px;

}

@media (max-width: 768px) {

.main-content {

padding-top: 85px;

}

}

.page-title {

font-size: 2.2rem;

color: var(--primary-color);

margin: 30px 0 15px;

text-align: center;

font-weight: 700;

}

.page-subtitle {

font-size: 1.1rem;

color: var(--text-light);

text-align: center;

margin-bottom: 40px;

max-width: 700px;

margin-left: auto;

margin-right: auto;

}

/\* ===== HERO BANNER (INDEX) ===== \*/

.hero-banner {

background: linear-gradient(rgba(0, 0, 0, 0.7), rgba(0, 0, 0, 0.7)), url('../img/hero-bg.jpg');

background-size: cover;

background-position: center;

color: white;

padding: 180px 0 100px;

text-align: center;

position: relative;

overflow: hidden;

}

.hero-banner::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background: linear-gradient(135deg, rgba(33, 63, 87, 0.8) 0%, rgba(180, 148, 52, 0.6) 100%);

z-index: -1;

}

.hero-content {

max-width: 800px;

margin: 0 auto;

position: relative;

z-index: 1;

}

.hero-banner h1 {

font-size: 2.8rem;

margin-bottom: 20px;

font-weight: 700;

text-shadow: 0 2px 10px rgba(0, 0, 0, 0.3);

}

.hero-banner p {

font-size: 1.2rem;

margin-bottom: 30px;

line-height: 1.6;

}

.hero-btn {

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: white;

padding: 12px 30px;

border-radius: 6px;

font-weight: 600;

text-decoration: none;

transition: var(--transition);

display: inline-block;

box-shadow: 0 4px 15px rgba(180, 148, 52, 0.3);

}

.hero-btn:hover {

transform: translateY(-3px);

box-shadow: 0 6px 20px rgba(180, 148, 52, 0.4);

}

/\* ===== CARROSSEL DE SERVIÇOS (INDEX) ===== \*/

.services-carousel {

margin-bottom: 70px;

}

.carousel-container {

position: relative;

overflow: hidden;

padding: 20px 0;

}

.carousel-track {

display: flex;

transition: transform 0.5s ease;

gap: 25px;

}

.carousel-slide {

min-width: calc(25% - 19px);

flex: 0 0 calc(25% - 19px);

}

.service-card {

background: var(--card-bg);

border-radius: var(--border-radius);

padding: 30px 25px;

box-shadow: var(--card-shadow);

transition: var(--transition);

height: 100%;

display: flex;

flex-direction: column;

position: relative;

overflow: hidden;

border: 1px solid rgba(180, 148, 52, 0.1);

}

.service-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: linear-gradient(90deg, var(--secondary-color), var(--accent-color));

opacity: 0;

transition: var(--transition);

}

.service-card:hover {

transform: translateY(-10px);

box-shadow: 0 15px 30px rgba(0, 0, 0, 0.12);

}

.service-card:hover::before {

opacity: 1;

}

.service-icon {

width: 70px;

height: 70px;

background: linear-gradient(135deg, rgba(180, 148, 52, 0.1), rgba(212, 177, 95, 0.15));

border-radius: 18px;

display: flex;

align-items: center;

justify-content: center;

margin-bottom: 25px;

color: var(--secondary-color);

font-size: 1.8rem;

transition: var(--transition);

}

.service-card:hover .service-icon {

transform: scale(1.05);

background: linear-gradient(135deg, rgba(180, 148, 52, 0.15), rgba(212, 177, 95, 0.2));

}

.service-card h3 {

font-size: 1.4rem;

margin-bottom: 15px;

color: var(--primary-color);

font-weight: 600;

}

.service-card p {

color: var(--text-light);

margin-bottom: 20px;

flex-grow: 1;

line-height: 1.6;

}

.service-card .price {

display: block;

font-weight: 700;

color: var(--secondary-color);

font-size: 1.4rem;

margin-bottom: 20px;

}

/\* ===== SEÇÃO DE INFORMAÇÕES (INDEX) ===== \*/

.oil-info-section {

margin-bottom: 70px;

}

.info-grid {

display: grid;

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

gap: 25px;

}

.info-card {

background: var(--card-bg);

padding: 30px 25px;

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

text-align: center;

transition: var(--transition);

border: 1px solid rgba(180, 148, 52, 0.1);

position: relative;

overflow: hidden;

}

.info-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: linear-gradient(90deg, var(--secondary-color), var(--accent-color));

opacity: 0;

transition: var(--transition);

}

.info-card:hover {

transform: translateY(-5px);

box-shadow: 0 12px 25px rgba(0, 0, 0, 0.1);

}

.info-card:hover::before {

opacity: 1;

}

.info-card i {

font-size: 2.5rem;

color: var(--secondary-color);

margin-bottom: 20px;

transition: var(--transition);

}

.info-card:hover i {

transform: scale(1.1);

}

.info-card h3 {

font-size: 1.3rem;

margin-bottom: 15px;

color: var(--primary-color);

font-weight: 600;

}

.info-card p {

color: var(--text-light);

font-size: 0.95rem;

line-height: 1.6;

}

/\* ===== OFERTAS ESPECIAIS (INDEX) ===== \*/

.special-offers {

margin-bottom: 70px;

}

.offer-cards {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

gap: 25px;

}

.offer-card {

background: var(--card-bg);

padding: 30px 25px;

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

position: relative;

transition: var(--transition);

border: 1px solid rgba(180, 148, 52, 0.1);

overflow: hidden;

}

.offer-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: linear-gradient(90deg, var(--secondary-color), var(--accent-color));

opacity: 0;

transition: var(--transition);

}

.offer-card:hover {

transform: translateY(-5px);

box-shadow: 0 15px 30px rgba(0, 0, 0, 0.12);

}

.offer-card:hover::before {

opacity: 1;

}

.offer-card.highlight {

border: 2px solid var(--secondary-color);

transform: scale(1.02);

}

.offer-card.highlight:hover {

transform: translateY(-5px) scale(1.02);

}

.offer-badge {

position: absolute;

top: -10px;

right: 20px;

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: white;

padding: 6px 18px;

border-radius: 20px;

font-size: 0.8rem;

font-weight: 600;

box-shadow: 0 4px 10px rgba(180, 148, 52, 0.3);

z-index: 2;

}

.offer-card h3 {

font-size: 1.4rem;

margin-bottom: 15px;

color: var(--primary-color);

font-weight: 600;

}

.offer-card p {

color: var(--text-light);

margin-bottom: 20px;

font-size: 0.95rem;

line-height: 1.6;

}

.price-container {

margin-bottom: 20px;

position: relative;

display: flex;

align-items: center;

gap: 15px;

}

.old-price {

text-decoration: line-through;

color: var(--text-light);

font-size: 1.1rem;

}

.new-price {

font-weight: 700;

color: var(--secondary-color);

font-size: 1.5rem;

}

.discount {

background: var(--success-color);

color: white;

padding: 4px 10px;

border-radius: 20px;

font-size: 0.8rem;

font-weight: 600;

}

/\* ===== PROGRESS STEPS (SERVIÇOS) ===== \*/

.progress-steps {

display: flex;

justify-content: space-between;

margin-bottom: 40px;

position: relative;

padding: 0 20px;

}

.progress-steps::before {

content: '';

position: absolute;

top: 50%;

left: 10%;

right: 10%;

height: 2px;

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

z-index: -1;

}

.progress-step {

display: flex;

flex-direction: column;

align-items: center;

position: relative;

width: 25%;

}

.progress-step::before {

content: '';

width: 22px;

height: 22px;

border-radius: 50%;

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

margin-bottom: 12px;

transition: var(--transition);

z-index: 2;

box-shadow: 0 0 0 4px var(--bg-color);

}

.progress-step.active::before {

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

border: 3px solid var(--bg-color);

box-shadow: 0 0 0 2px var(--secondary-color);

}

.progress-step.completed::before {

background: linear-gradient(135deg, var(--success-color), #20c997);

content: '\f00c';

font-family: 'Font Awesome 6 Free';

font-weight: 900;

color: white;

display: flex;

align-items: center;

justify-content: center;

font-size: 0.7rem;

border: 3px solid var(--bg-color);

box-shadow: 0 0 0 2px var(--success-color);

}

.progress-step span {

font-size: 0.9rem;

color: var(--text-light);

text-align: center;

font-weight: 500;

}

.progress-step.active span {

color: var(--primary-color);

font-weight: 600;

}

/\* ===== SERVICE STEPS (SERVIÇOS) ===== \*/

.service-step {

display: none;

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

border-radius: var(--border-radius);

padding: 30px;

box-shadow: var(--card-shadow);

margin-bottom: 30px;

animation: fadeIn 0.5s ease-out;

border: 1px solid rgba(180, 148, 52, 0.1);

}

.service-step.active {

display: block;

}

@keyframes fadeIn {

from { opacity: 0; transform: translateY(10px); }

to { opacity: 1; transform: translateY(0); }

}

.step-number {

display: inline-flex;

align-items: center;

justify-content: center;

width: 32px;

height: 32px;

background: linear-gradient(135deg, var(--primary-color), #2a4f6d);

color: white;

border-radius: 50%;

margin-right: 12px;

font-size: 0.9rem;

font-weight: 600;

box-shadow: 0 4px 8px rgba(33, 63, 87, 0.2);

}

/\* ===== FORMULÁRIOS ===== \*/

/\* ===== FORMULÁRIOS ===== \*/

.form-row {

display: flex;

gap: 20px;

margin-bottom: 20px;

}

.form-group {

margin-bottom: 25px;

flex: 1;

position: relative;

}

.form-group label {

display: block;

margin-bottom: 10px;

font-weight: 500;

color: var(--text-color);

font-size: 0.95rem;

}

.form-group select,

.form-group input,

.form-group textarea {

width: 100%;

padding: 14px 20px;

border: 2px solid var(--border-color);

border-radius: 15px; /\* Aumentado para deixar mais redondinho \*/

font-size: 1rem;

transition: var(--transition);

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

color: var(--text-color);

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

}

.form-group select:focus,

.form-group input:focus,

.form-group textarea:focus {

outline: none;

border-color: var(--secondary-color);

box-shadow: 0 0 0 4px rgba(180, 148, 52, 0.15);

}

.form-group.error input,

.form-group.error select,

.form-group.error textarea {

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

box-shadow: 0 0 0 4px rgba(220, 53, 69, 0.15);

}

.error-message {

color: var(--danger-color);

font-size: 0.85rem;

margin-top: 8px;

display: none;

padding-left: 5px;

}

.form-group.error .error-message {

display: block;

}

.password-input {

position: relative;

}

.toggle-password {

position: absolute;

right: 20px;

top: 50%;

transform: translateY(-50%);

background: none;

border: none;

color: var(--text-light);

cursor: pointer;

font-size: 1rem;

transition: var(--transition);

z-index: 2;

}

.toggle-password:hover {

color: var(--secondary-color);

}

/\* Estilização para checkboxes e radio buttons \*/

.form-group.checkbox,

.form-group.radio {

display: flex;

align-items: center;

gap: 10px;

}

.form-group.checkbox input[type="checkbox"],

.form-group.radio input[type="radio"] {

width: 20px;

height: 20px;

border-radius: 5px; /\* Checkbox levemente arredondado \*/

accent-color: var(--secondary-color);

}

.form-group.radio input[type="radio"] {

border-radius: 50%; /\* Radio totalmente redondo \*/

}

/\* Estilos para selects com seta personalizada \*/

.form-group select {

appearance: none;

background-image: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg' width='16' height='16' viewBox='0 0 24 24' fill='none' stroke='%23777' stroke-width='2' stroke-linecap='round' stroke-linejoin='round'%3E%3Cpath d='M6 9l6 6 6-6'/%3E%3C/svg%3E");

background-repeat: no-repeat;

background-position: right 18px center;

background-size: 16px;

padding-right: 50px;

}

/\* Botões de formulário \*/

.form-buttons {

display: flex;

gap: 15px;

margin-top: 25px;

justify-content: flex-end;

}

.form-buttons .btn {

border-radius: 12px; /\* Botões mais arredondados \*/

padding: 12px 28px;

font-weight: 600;

}

/\* Campos de formulário com ícones \*/

.form-group.with-icon {

position: relative;

}

.form-group.with-icon i {

position: absolute;

left: 18px;

top: 50%;

transform: translateY(-50%);

color: var(--text-light);

z-index: 2;

}

.form-group.with-icon input,

.form-group.with-icon select {

padding-left: 50px;

}

/\* Agrupamento de campos relacionados \*/

.field-group {

background: rgba(180, 148, 52, 0.05);

border-radius: 15px;

padding: 20px;

margin-bottom: 25px;

border: 1px solid rgba(180, 148, 52, 0.1);

}

.field-group-legend {

font-weight: 600;

color: var(--secondary-color);

margin-bottom: 15px;

display: flex;

align-items: center;

gap: 10px;

}

/\* Estilo para textareas \*/

.form-group textarea {

min-height: 120px;

resize: vertical;

line-height: 1.5;

}

/\* Placeholders estilizados \*/

.form-group input::placeholder,

.form-group textarea::placeholder {

color: var(--text-light);

opacity: 0.7;

}

/\* Estados de desabilitado \*/

.form-group select:disabled,

.form-group input:disabled,

.form-group textarea:disabled {

background-color: rgba(0, 0, 0, 0.05);

color: var(--text-light);

cursor: not-allowed;

}

/\* ===== LOCATION SEARCH (SERVIÇOS) ===== \*/

.location-search {

display: flex;

gap: 15px;

}

.location-search input {

flex: 1;

border-radius: 50px;

padding: 12px 20px;

border: 2px solid var(--border-color);

transition: var(--transition);

}

.location-search input:focus {

border-color: var(--secondary-color);

box-shadow: 0 0 0 3px rgba(180, 148, 52, 0.2);

}

.location-or {

text-align: center;

margin: 30px 0;

color: var(--text-light);

font-size: 0.95rem;

position: relative;

}

.location-or::before,

.location-or::after {

content: '';

position: absolute;

top: 50%;

width: 40%;

height: 1px;

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

}

.location-or::before {

left: 0;

}

.location-or::after {

right: 0;

}

/\* ===== MAP SECTION (SERVIÇOS) ===== \*/

.map-container {

display: flex;

flex-direction: column;

gap: 25px;

margin-top: 30px;

}

#map {

height: 350px;

width: 100%;

border-radius: 20px;

box-shadow: var(--card-shadow);

border: 2px solid var(--border-color);

}

.map-sidebar {

margin-top: 25px;

}

.workshop-list {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));

gap: 25px;

}

.workshop-item {

background: var(--card-bg);

padding: 25px;

border-radius: 20px;

box-shadow: var(--card-shadow);

transition: var(--transition);

cursor: pointer;

border: 2px solid var(--border-color);

}

.workshop-item:hover {

transform: translateY(-8px);

box-shadow: 0 15px 30px rgba(0,0,0,0.15);

border-color: rgba(180, 148, 52, 0.4);

}

.workshop-item.active {

border-left: 6px solid var(--secondary-color);

background-color: rgba(180, 148, 52, 0.1);

transform: scale(1.02);

}

.workshop-item h4 {

color: var(--primary-color);

margin-bottom: 15px;

font-weight: 600;

font-size: 1.1rem;

}

.workshop-item p {

margin-bottom: 12px;

font-size: 0.95rem;

color: var(--text-light);

display: flex;

align-items: center;

gap: 10px;

}

.workshop-item p i {

color: var(--secondary-color);

width: 22px;

font-size: 1rem;

}

/\* Estilo para a lista de oficinas \*/

.workshop-list-container {

margin-top: 30px;

display: grid;

grid-template-columns: repeat(auto-fill, minmax(320px, 1fr));

gap: 30px;

}

.workshop-item {

border: 2px solid var(--border-color);

border-radius: 20px;

padding: 25px;

transition: all 0.3s ease;

background: var(--card-bg);

box-shadow: var(--card-shadow);

}

.workshop-item:hover {

transform: translateY(-8px) scale(1.02);

box-shadow: 0 20px 35px rgba(0,0,0,0.15);

}

.workshop-item h4 {

margin-top: 0;

color: var(--primary-color);

font-weight: 600;

font-size: 1.2rem;

}

.workshop-item .address,

.workshop-item .phone,

.workshop-item .hours {

margin: 12px 0;

color: var(--text-light);

display: flex;

align-items: center;

gap: 10px;

font-size: 0.95rem;

}

.select-btn {

background: linear-gradient(135deg, var(--primary-color), #2a4f6d);

color: white;

border: none;

padding: 12px 25px;

border-radius: 50px;

cursor: pointer;

width: 100%;

margin-top: 20px;

font-weight: 600;

transition: all 0.3s ease;

box-shadow: 0 4px 15px rgba(33, 63, 87, 0.3);

}

.select-btn:hover {

background: linear-gradient(135deg, #2a4f6d, var(--primary-color));

transform: translateY(-3px) scale(1.05);

box-shadow: 0 8px 20px rgba(33, 63, 87, 0.4);

}

/\* ===== PRODUCT RECOMMENDATIONS (SERVIÇOS) ===== \*/

.recommendation-container {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));

gap: 25px;

margin: 30px 0;

}

.product-card {

background: var(--card-bg);

padding: 25px;

border-radius: 20px;

box-shadow: var(--card-shadow);

transition: all 0.3s ease;

border: 2px solid var(--border-color);

overflow: hidden;

}

.product-card:hover {

transform: translateY(-8px) scale(1.02);

box-shadow: 0 20px 35px rgba(0, 0, 0, 0.15);

}

.product-card h3 {

color: var(--secondary-color);

margin-bottom: 20px;

padding-bottom: 15px;

border-bottom: 2px solid var(--border-color);

font-weight: 600;

font-size: 1.2rem;

}

.product-image {

width: 100%;

height: 180px;

background: var(--bg-color);

margin-bottom: 20px;

border-radius: 15px;

display: flex;

align-items: center;

justify-content: center;

overflow: hidden;

border: 2px solid var(--border-color);

}

.product-image img {

max-width: 100%;

max-height: 100%;

object-fit: contain;

transition: all 0.4s ease;

border-radius: 12px;

}

.product-card:hover .product-image img {

transform: scale(1.1);

}

/\* ===== SCHEDULING (SERVIÇOS) ===== \*/

.scheduling-container {

display: flex;

flex-direction: column;

gap: 30px;

}

.workshop-summary {

background: var(--card-bg);

padding: 25px;

border-radius: 20px;

border-left: 6px solid var(--secondary-color);

box-shadow: var(--card-shadow);

}

.schedule-form {

width: 100%;

}

.step-buttons {

display: flex;

justify-content: space-between;

margin-top: 40px;

gap: 25px;

}

/\* ===== CONFIRMATION (SERVIÇOS) ===== \*/

/\* ===== CONFIRMATION (SERVIÇOS) - ATUALIZADO ===== \*/

.confirmation-container {

text-align: center;

max-width: 500px;

margin: 0 auto;

padding: 20px;

background: var(--light-color);

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

}

.confirmation-icon {

font-size: 3rem;

color: var(--success-color);

margin-bottom: 15px;

animation: bounce 0.5s;

}

.confirmation-container h2 {

font-size: 1.5rem;

color: var(--primary-color);

margin-bottom: 20px;

}

.confirmation-details {

background: var(--bg-color);

padding: 20px;

border-radius: var(--border-radius);

margin: 20px 0;

text-align: left;

border-left: 4px solid var(--secondary-color);

}

.confirmation-details p {

margin-bottom: 12px;

font-size: 0.95rem;

line-height: 1.5;

}

.confirmation-details strong {

color: var(--primary-color);

display: inline-block;

min-width: 120px;

}

.confirmation-text {

text-align: center;

color: var(--text-light);

margin-bottom: 20px;

font-size: 0.9rem;

line-height: 1.4;

}

.confirmation-details .service-item {

margin-left: 20px;

padding: 8px 0;

border-bottom: 1px dashed var(--border-color);

}

.confirmation-details .service-details {

font-size: 0.85rem;

color: var(--text-light);

margin-top: 4px;

}

.confirmation-details .total-amount {

font-size: 1.1rem;

font-weight: 700;

color: var(--secondary-color);

margin-top: 15px;

padding-top: 15px;

border-top: 2px solid var(--border-color);

}

/\* Responsividade para a confirmação \*/

@media (max-width: 768px) {

.confirmation-container {

padding: 15px;

margin: 0 10px;

}

.confirmation-icon {

font-size: 2.5rem;

margin-bottom: 10px;

}

.confirmation-container h2 {

font-size: 1.3rem;

margin-bottom: 15px;

}

.confirmation-details {

padding: 15px;

margin: 15px 0;

}

.confirmation-details p {

font-size: 0.85rem;

margin-bottom: 15px;

}

.confirmation-details strong {

min-width: 100px;

display: block;

margin-bottom: 5px;

}

.confirmation-text {

font-size: 0.85rem;

margin-bottom: 15px;

}

}

@media (max-width: 480px) {

.confirmation-container {

padding: 12px;

}

.confirmation-icon {

font-size: 2rem;

}

.confirmation-container h2 {

font-size: 1.2rem;

}

.confirmation-details {

padding: 10px;

}

.confirmation-details p {

font-size: 0.8rem;

}

.confirmation-details strong {

min-width: 60px;

font-size: 0.8rem;

}

}

/\* ===== ABOUT BANNER (SOBRE) ===== \*/

.about-banner {

background: linear-gradient(rgba(0, 0, 0, 0.7), rgba(0, 0, 0, 0.7)), url('../img/about-bg.jpg');

background-size: cover;

background-position: center;

color: white;

padding: 160px 0 100px;

text-align: center;

position: relative;

}

.about-banner::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background: linear-gradient(135deg, rgba(33, 63, 87, 0.85) 0%, rgba(180, 148, 52, 0.7) 100%);

z-index: 0;

}

.about-banner h1 {

font-size: 2.8rem;

margin-bottom: 20px;

font-weight: 700;

position: relative;

z-index: 1;

text-shadow: 0 2px 10px rgba(0, 0, 0, 0.3);

}

.about-banner p {

font-size: 1.2rem;

position: relative;

z-index: 1;

max-width: 700px;

margin: 0 auto;

line-height: 1.6;

}

/\* ===== ABOUT CONTAINER (SOBRE) ===== \*/

.about-container {

padding: 60px 0;

}

.about-section {

display: flex;

gap: 60px;

margin-bottom: 80px;

align-items: center;

}

.about-content {

flex: 1;

}

.about-image {

flex: 1;

border-radius: var(--border-radius);

overflow: hidden;

box-shadow: var(--card-shadow);

position: relative;

}

.about-image::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background: linear-gradient(135deg, rgba(33, 63, 87, 0.1), rgba(180, 148, 52, 0.1));

z-index: 1;

opacity: 0;

transition: var(--transition);

}

.about-image:hover::before {

opacity: 1;

}

.about-image img {

width: 100%;

height: auto;

display: block;

transition: transform 0.5s ease;

}

.about-image:hover img {

transform: scale(1.05);

}

.milestones {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(160px, 1fr));

gap: 25px;

margin-top: 40px;

}

.milestone {

background: rgba(180, 148, 52, 0.1);

padding: 25px 20px;

border-radius: var(--border-radius);

text-align: center;

transition: var(--transition);

border: 1px solid rgba(180, 148, 52, 0.1);

}

.milestone:hover {

transform: translateY(-5px);

box-shadow: 0 10px 20px rgba(180, 148, 52, 0.1);

}

.milestone .number {

font-size: 2.2rem;

font-weight: 700;

color: var(--secondary-color);

margin-bottom: 8px;

}

.milestone .description {

font-size: 0.95rem;

color: var(--text-light);

}

/\* ===== MISSÃO, VISÃO E VALORES (SOBRE) ===== \*/

.mv-section {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(320px, 1fr));

gap: 35px;

margin-bottom: 80px;

}

.mv-card {

background: var(--card-bg);

padding: 35px 30px;

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

text-align: center;

transition: var(--transition);

border: 1px solid rgba(180, 148, 52, 0.1);

position: relative;

overflow: hidden;

}

.mv-card:hover {

transform: translateY(-10px);

}

.mv-card i {

font-size: 2.5rem;

color: var(--secondary-color);

margin-bottom: 20px;

}

.mv-card h3 {

font-size: 1.5rem;

margin-bottom: 15px;

color: var(--primary-color);

}

.mv-card p, .mv-card ul {

color: var(--text-light);

font-size: 0.95rem;

}

.mv-card ul {

text-align: left;

list-style-position: inside;

padding-left: 10px;

}

.mv-card li {

margin-bottom: 8px;

}

/\* ===== DIFERENCIAIS (SOBRE) ===== \*/

.diferenciais-section {

margin-bottom: 60px;

}

.diferenciais-grid {

display: grid;

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

gap: 25px;

}

.diferencial-card {

background: var(--light-color);

padding: 30px;

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

text-align: center;

transition: var(--transition);

}

.diferencial-card:hover {

transform: translateY(-5px);

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);

}

.diferencial-card i {

font-size: 2rem;

color: var(--secondary-color);

margin-bottom: 20px;

}

.diferencial-card h3 {

font-size: 1.3rem;

margin-bottom: 15px;

color: var(--primary-color);

}

.diferencial-card p {

color: var(--text-light);

font-size: 0.9rem;

}

/\* ===== TESTIMONIALS (SOBRE) ===== \*/

.testimonials-section {

margin-bottom: 60px;

}

.testimonials-container {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

gap: 30px;

margin-top: 40px;

}

.testimonial {

background: var(--light-color);

padding: 30px;

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

transition: var(--transition);

}

.testimonial:hover {

transform: translateY(-5px);

}

.quote {

position: relative;

margin-bottom: 25px;

}

.quote i {

position: absolute;

top: -10px;

left: -10px;

color: rgba(180, 148, 52, 0.2);

font-size: 3rem;

z-index: 1;

}

.quote p {

position: relative;

z-index: 2;

font-style: italic;

color: var(--text-color);

}

.client {

display: flex;

align-items: center;

gap: 15px;

}

.client img {

width: 60px;

height: 60px;

border-radius: 50%;

object-fit: cover;

}

.client-info h4 {

font-size: 1.1rem;

color: var(--primary-color);

margin-bottom: 5px;

}

.client-info p {

color: var(--text-light);

font-size: 0.8rem;

}

/\* ===== AGENDA ===== \*/

.agenda-container {

display: flex;

flex-direction: column;

gap: 40px;

margin-bottom: 50px;

}

.section-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 20px;

padding-bottom: 10px;

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

}

.section-header h2 {

font-size: 1.5rem;

color: var(--primary-color);

display: flex;

align-items: center;

gap: 10px;

}

.filter-options select {

padding: 8px 15px;

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

border-radius: var(--border-radius);

font-size: 0.9rem;

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

color: var(--text-color);

}

/\* ===== REMINDER CARD (AGENDA) ===== \*/

.reminder-section {

margin-bottom: 40px;

}

.reminder-card {

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

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

padding: 20px;

display: flex;

justify-content: space-between;

align-items: center;

transition: var(--transition);

}

.reminder-card:hover {

transform: translateY(-5px);

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);

}

.reminder-content {

display: flex;

align-items: center;

gap: 20px;

}

.reminder-icon {

width: 60px;

height: 60px;

background-color: rgba(180, 148, 52, 0.1);

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

color: var(--secondary-color);

font-size: 1.5rem;

}

.reminder-details h3 {

font-size: 1.2rem;

color: var(--primary-color);

margin-bottom: 5px;

}

.reminder-details p {

color: var(--text-light);

font-size: 0.95rem;

margin-bottom: 5px;

}

.protocol {

font-size: 0.85rem;

color: var(--text-light);

}

.status {

padding: 5px 10px;

border-radius: 20px;

font-size: 0.8rem;

font-weight: 600;

}

.status.completed {

background-color: rgba(40, 167, 69, 0.1);

color: var(--success-color);

}

.status.pending {

background-color: rgba(255, 193, 7, 0.1);

color: var(--warning-color);

}

.status.canceled {

background-color: rgba(220, 53, 69, 0.1);

color: var(--danger-color);

}

.service-info h3 {

font-size: 1.2rem;

color: var(--primary-color);

margin-bottom: 10px;

}

.service-info p, .date-info p, .location-info p {

color: var(--text-color);

font-size: 0.95rem;

margin-bottom: 8px;

display: flex;

align-items: center;

gap: 8px;

}

.service-info i, .date-info i, .location-info i {

color: var(--secondary-color);

width: 20px;

text-align: center;

}

.appointment-footer {

padding: 15px 20px;

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

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

display: flex;

gap: 10px;

flex-wrap: wrap;

}

/\* ===== MODAL ===== \*/

.modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

z-index: 1001;

justify-content: center;

align-items: center;

backdrop-filter: blur(3px);

}

.modal.show {

display: flex;

}

.modal-content {

background: var(--light-color);

padding: 30px;

border-radius: 15px;

width: 100%;

max-width: 450px;

position: relative;

animation: modalFadeIn 0.3s;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);

}

@keyframes modalFadeIn {

from { opacity: 0; transform: translateY(-20px); }

to { opacity: 1; transform: translateY(0); }

}

.modal-header {

text-align: center;

margin-bottom: 30px;

}

.modal-header h2 {

font-size: 1.8rem;

color: var(--primary-color);

margin-bottom: 5px;

}

.modal-header p {

color: var(--text-light);

font-size: 0.9rem;

}

.close-modal {

position: absolute;

top: 15px;

right: 15px;

font-size: 1.5rem;

cursor: pointer;

color: var(--text-light);

transition: var(--transition);

background: none;

border: none;

}

.close-modal:hover {

color: var(--text-color);

}

/\* ===== LOGIN FORM ===== \*/

.login-form {

margin-bottom: 20px;

}

/\* ===== DIVISOR ===== \*/

.divider {

display: flex;

align-items: center;

margin: 25px 0;

color: var(--text-light);

font-size: 0.9rem;

}

.divider::before,

.divider::after {

content: "";

flex: 1;

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

}

.divider::before {

margin-right: 10px;

}

.divider::after {

margin-left: 10px;

}

/\* ===== SOCIAL LOGIN ===== \*/

.social-login {

display: flex;

flex-direction: column;

gap: 10px;

margin-bottom: 20px;

}

.social-btn {

display: flex;

align-items: center;

justify-content: center;

gap: 10px;

padding: 12px;

border-radius: 8px;

font-weight: 500;

transition: var(--transition);

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

background: var(--light-color);

color: var(--text-color);

cursor: pointer;

font-size: 0.9rem;

}

.social-btn:hover {

background: var(--bg-color);

}

.social-btn.google i {

color: #DB4437;

}

.register-link {

text-align: center;

font-size: 0.9rem;

color: var(--text-light);

}

.register-link a {

color: var(--secondary-color);

text-decoration: none;

font-weight: 500;

transition: var(--transition);

}

.register-link a:hover {

text-decoration: underline;

}

/\* ===== LOADING & TOAST ===== \*/

#loading-overlay {

position: fixed;

top: 0;

left: 0;

right: 0;

bottom: 0;

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

display: none;

justify-content: center;

align-items: center;

z-index: 9999;

}

.loader {

border: 5px solid var(--border-color);

border-top: 5px solid var(--secondary-color);

border-radius: 50%;

width: 50px;

height: 50px;

animation: spin 1s linear infinite;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

.toast {

position: fixed;

bottom: 20px;

left: 50%;

transform: translateX(-50%);

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

color: white;

padding: 12px 24px;

border-radius: 4px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);

z-index: 1000;

opacity: 0;

transition: opacity 0.3s ease;

}

.toast.show {

opacity: 1;

}

.toast.success {

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

}

.toast.warning {

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

color: var(--dark-color);

}

.toast.error {

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

}

/\* ===== FOOTER ===== \*/

.footer {

background: var(--primary-color);

color: white;

padding: 40px 0 20px;

margin-top: 50px;

}

.footer-content {

display: grid;

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

gap: 30px;

margin-bottom: 30px;

}

.footer-brand img {

height: 110px;

margin-bottom: 15px;

}

.footer-brand p {

color: #ccc;

font-size: 0.9rem;

}

.footer-links h3,

.footer-contact h3 {

color: var(--secondary-color);

margin-bottom: 20px;

font-size: 1.2rem;

position: relative;

padding-bottom: 10px;

}

.footer-links h3::after,

.footer-contact h3::after {

content: '';

position: absolute;

left: 0;

bottom: 0;

width: 40px;

height: 2px;

background: var(--secondary-color);

}

.footer-links ul {

list-style: none;

}

.footer-links li {

margin-bottom: 10px;

}

.footer-links a {

color: #ccc;

text-decoration: none;

transition: color 0.3s;

font-size: 0.9rem;

}

.footer-links a:hover {

color: var(--secondary-color);

}

.footer-contact p {

color: #ccc;

margin-bottom: 15px;

font-size: 0.9rem;

display: flex;

align-items: center;

gap: 10px;

}

.footer-contact i {

color: var(--secondary-color);

}

.copyright {

text-align: center;

padding-top: 20px;

border-top: 1px solid rgba(255,255,255,0.1);

color: #ccc;

font-size: 0.8rem;

}

/\* ===== RESPONSIVIDADE MELHORADA ===== \*/

@media (max-width: 1200px) {

.container {

width: 95%;

}

.hero-banner h1 {

font-size: 2.2rem;

}

.about-banner h1 {

font-size: 2.2rem;

}

}

@media (max-width: 1024px) {

.main-grid {

grid-template-columns: 1fr;

}

.carousel-slide {

min-width: calc(33.333% - 15px);

flex: 0 0 calc(33.333% - 15px);

}

.about-section {

flex-direction: column;

gap: 30px;

}

.about-image {

order: -1;

margin-bottom: 20px;

}

.nav\_\_list {

gap: 20px;

}

.offer-cards {

grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));

}

}

/\* ===== MELHORIAS ADICIONAIS PARA MOBILE ===== \*/

@media (max-width: 768px) {

/\* Ajustes gerais de espaçamento \*/

.container {

padding: 0 15px;

}

.main-content {

padding-top: 90px;

padding-bottom: 30px;

}

/\* Melhorias no header \*/

.header {

height: 70px;

}

.logo img {

height: 100px;

}

/\* Menu hamburger \*/

.hamburger {

display: flex;

}

.nav {

position: fixed;

top: 70px;

left: -100%;

width: 100%;

height: calc(100vh - 70px);

background: var(--light-color);

transition: left 0.3s ease;

z-index: 999;

box-shadow: 2px 0 10px rgba(0, 0, 0, 0.1);

overflow-y: auto;

}

.nav.active {

left: 0;

}

.nav\_\_list {

flex-direction: column;

padding: 20px;

gap: 15px;

}

.nav\_\_list a {

color: var(--text-color);

font-size: 1.1rem;

padding: 12px 0;

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

width: 100%;

display: block;

}

.nav\_\_list a:last-child {

border-bottom: none;

}

.nav\_\_list a:hover::after,

.nav\_\_list a.active::after {

width: 0;

}

.nav\_\_list a:hover,

.nav\_\_list a.active {

color: var(--secondary-color);

}

/\* Ajustes no hero banner \*/

.hero-banner {

padding: 100px 0 60px;

}

.hero-banner h1 {

font-size: 2rem;

line-height: 1.3;

}

.hero-banner p {

font-size: 1rem;

margin-bottom: 20px;

}

/\* Melhorias nos cards \*/

.carousel-slide {

min-width: calc(50% - 10px);

flex: 0 0 calc(50% - 10px);

}

.service-card, .info-card, .offer-card {

padding: 20px;

}

.service-icon {

width: 50px;

height: 50px;

font-size: 1.2rem;

}

/\* Ajustes nos formulários \*/

.form-row {

flex-direction: column;

gap: 0;

}

.form-group {

margin-bottom: 15px;

}

.form-group input,

.form-group select {

padding: 10px 12px;

}

.location-search {

flex-direction: column;

gap: 10px;

}

.location-search input,

.location-search button {

width: 100%;

}

.btn {

padding: 10px 15px;

font-size: 0.9rem;

}

.step-buttons {

flex-direction: column;

gap: 10px;

}

.step-buttons .btn {

width: 100%;

}

/\* Melhorias específicas para a página de agendamento \*/

.progress-step {

width: calc(50% - 7.5px);

margin-bottom: 15px;

font-size: 0.8rem;

}

.progress-steps::before {

display: none;

}

.appointment-header,

.appointment-body,

.appointment-footer {

padding: 15px;

}

.appointment-footer .btn {

width: 100%;

margin-bottom: 5px;

}

/\* Ajustes para o mapa \*/

#map {

height: 250px;

}

/\* Melhorias na página sobre \*/

.mv-section {

grid-template-columns: 1fr;

gap: 20px;

}

.diferenciais-grid {

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

gap: 15px;

}

.testimonial {

padding: 20px;

}

/\* Ajustes nos modais \*/

.modal-content {

width: 90%;

max-width: none;

padding: 20px;

margin: 20px;

}

/\* Melhorias no footer \*/

.footer-content {

grid-template-columns: 1fr;

text-align: center;

gap: 25px;

}

.footer-links h3::after,

.footer-contact h3::after {

left: 50%;

transform: translateX(-50%);

}

}

@media (max-width: 480px) {

/\* Extra Small Mobile \*/

.container {

padding: 0 10px;

}

.carousel-slide {

min-width: 100%;

flex: 0 0 100%;

}

.offer-cards {

grid-template-columns: 1fr;

gap: 15px;

}

.hero-banner {

padding: 80px 0 50px;

}

.hero-banner h1 {

font-size: 1.8rem;

}

.hero-banner p {

font-size: 0.95rem;

}

.about-banner {

padding: 80px 0 50px;

}

.about-banner h1 {

font-size: 1.8rem;

}

.page-title {

font-size: 1.6rem;

}

.section-title {

font-size: 1.5rem;

}

.testimonial {

padding: 20px 15px;

}

.modal-content {

padding: 20px 15px;

margin: 15px;

width: calc(100% - 30px);

}

.modal-header h2 {

font-size: 1.5rem;

}

.main-content {

padding: 30px 0;

}

.service-card {

padding: 15px;

}

.service-card h3 {

font-size: 1.1rem;

}

.info-card {

padding: 20px 15px;

}

.offer-card {

padding: 20px 15px;

}

.mv-card {

padding: 25px 20px;

}

.diferencial-card {

padding: 25px 20px;

}

.progress-step {

width: 100%;

margin-bottom: 10px;

}

.appointment-body {

grid-template-columns: 1fr;

gap: 15px;

padding: 15px;

}

.appointment-header {

padding: 12px 15px;

flex-direction: column;

gap: 10px;

align-items: flex-start;

}

.appointment-footer {

padding: 12px 15px;

}

.btn {

padding: 10px 15px;

font-size: 0.9rem;

}

.btn-outline {

padding: 8px 12px;

font-size: 0.85rem;

}

}

/\* ===== MELHORIAS DE PERFORMANCE ===== \*/

@media (prefers-reduced-motion: reduce) {

\* {

animation-duration: 0.01ms !important;

animation-iteration-count: 1 !important;

transition-duration: 0.01ms !important;

}

}

/\* ===== PRINT STYLES ===== \*/

@media print {

.header,

.footer,

.hamburger,

.modal,

.btn,

.social-login {

display: none !important;

}

.main-content {

margin-top: 0;

}

body {

background: white;

color: black;

}

.container {

width: 100%;

max-width: none;

}

}

/\* Autocomplete styles \*/

.autocomplete-container {

position: relative;

flex: 1;

}

.autocomplete-results {

position: absolute;

top: 100%;

left: 0;

right: 0;

z-index: 1000;

background: var(--card-bg);

border: 2px solid rgba(180, 148, 52, 0.2);

border-radius: 0 0 20px 20px;

box-shadow: 0 15px 30px rgba(180, 148, 52, 0.15);

max-height: 300px;

overflow-y: auto;

display: none;

backdrop-filter: blur(10px);

margin-top: 5px;

}

.autocomplete-results.active {

display: block;

animation: slideDown 0.3s ease-out;

}

@keyframes slideDown {

from {

opacity: 0;

transform: translateY(-10px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

.autocomplete-item {

padding: 15px 20px;

cursor: pointer;

transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);

border-bottom: 1px solid rgba(180, 148, 52, 0.1);

display: flex;

flex-direction: column;

}

.autocomplete-item:last-child {

border-bottom: none;

border-radius: 0 0 20px 20px;

}

.autocomplete-item:hover {

background: linear-gradient(135deg, rgba(180, 148, 52, 0.1), rgba(212, 177, 95, 0.15));

transform: translateX(5px);

}

.autocomplete-item .main-text {

font-weight: 600;

color: var(--primary-color);

font-size: 1rem;

margin-bottom: 4px;

}

.autocomplete-item .secondary-text {

font-size: 0.85rem;

color: var(--text-light);

opacity: 0.9;

}

/\* Scrollbar personalizada \*/

.autocomplete-results::-webkit-scrollbar {

width: 6px;

}

.autocomplete-results::-webkit-scrollbar-track {

background: rgba(180, 148, 52, 0.1);

border-radius: 0 0 20px 0;

}

.autocomplete-results::-webkit-scrollbar-thumb {

background: linear-gradient(to bottom, var(--secondary-color), var(--accent-color));

border-radius: 3px;

}

.autocomplete-results::-webkit-scrollbar-thumb:hover {

background: var(--accent-color);

}

/\* Responsividade Mobile - Apenas Banner e Cards \*/

@media (max-width: 768px) {

/\* Hero Banner ajustes \*/

.hero-content {

padding: 1.5rem;

text-align: center;

}

.hero-content h1 {

font-size: 1.8rem;

line-height: 1.3;

margin-bottom: 0.8rem;

}

.hero-content p {

font-size: 1rem;

line-height: 1.5;

margin-bottom: 1.5rem;

}

.hero-btn {

padding: 0.8rem 1.5rem;

font-size: 0.9rem;

}

/\* Ajustes dos Cards de Serviços \*/

.services-carousel {

padding: 0 1rem;

}

.services-carousel .carousel-track {

flex-direction: column;

gap: 1rem;

}

.service-card {

width: 100%;

margin: 0 auto 1.5rem;

padding: 1.2rem;

}

.service-card h3 {

font-size: 1.2rem;

}

.service-card p {

font-size: 0.9rem;

}

.service-icon {

width: 50px;

height: 50px;

margin-bottom: 1rem;

}

}

/\* Estilos para seleção de produtos \*/

.product-selection {

margin-top: 15px;

padding-top: 15px;

border-top: 2px dashed #eee;

}

.checkbox-container {

display: flex;

align-items: center;

cursor: pointer;

font-weight: 500;

color: #1d3557;

}

.checkbox-container input[type="checkbox"] {

margin-right: 10px;

width: 18px;

height: 18px;

}

.checkmark {

width: 18px;

height: 18px;

background: #fff;

border: 2px solid #1d3557;

border-radius: 3px;

margin-right: 10px;

position: relative;

}

.checkbox-container input[type="checkbox"]:checked + .checkmark {

background: #1d3557;

border-color: #1d3557;

}

.checkbox-container input[type="checkbox"]:checked + .checkmark::after {

content: "✓";

position: absolute;

color: white;

font-size: 12px;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

.checkbox-container input[type="checkbox"]:disabled + .checkmark {

background: #ccc;

border-color: #999;

cursor: not-allowed;

}

/\* ===== SELECTION SUMMARY - ESTILO ATUALIZADO ===== \*/

.selection-summary {

background: linear-gradient(135deg, rgba(42, 157, 143, 0.1) 0%, rgba(29, 53, 87, 0.1) 100%);

padding: 20px 25px;

border-radius: 16px;

margin: 25px 0;

border-left: 4px solid #2a9d8f;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08);

position: relative;

overflow: hidden;

transition: all 0.3s ease;

border: 1px solid rgba(42, 157, 143, 0.2);

}

.selection-summary:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(0, 0, 0, 0.12);

border-color: rgba(42, 157, 143, 0.4);

}

.selection-summary::before {

content: '';

position: absolute;

top: 0;

right: 0;

width: 80px;

height: 80px;

background: linear-gradient(135deg, rgba(42, 157, 143, 0.1) 0%, rgba(29, 53, 87, 0.1) 100%);

border-radius: 0 0 0 80px;

z-index: 0;

}

.selection-summary h4 {

margin: 0 0 12px 0;

color: #1d3557;

font-size: 1.1rem;

font-weight: 600;

display: flex;

align-items: center;

gap: 10px;

position: relative;

z-index: 1;

}

.selection-summary h4::before {

font-size: 1.2rem;

}

#selected-items {

margin: 0;

font-weight: 500;

font-size: 1rem;

color: #2a9d8f;

position: relative;

z-index: 1;

padding: 12px 15px;

background: rgba(255, 255, 255, 0.7);

border-radius: 10px;

border: 1px solid rgba(42, 157, 143, 0.2);

display: inline-flex;

align-items: center;

gap: 8px;

}

#selected-items::before {

content: '✓';

color: #2a9d8f;

font-weight: bold;

}

/\* Estado quando nenhum produto está selecionado \*/

.selection-summary.empty {

border-left-color: #e63946;

background: linear-gradient(135deg, rgba(230, 57, 70, 0.1) 0%, rgba(29, 53, 87, 0.1) 100%);

}

.selection-summary.empty h4 {

color: #e63946;

}

.selection-summary.empty #selected-items {

color: #e63946;

background: rgba(255, 255, 255, 0.7);

border-color: rgba(230, 57, 70, 0.2);

}

.selection-summary.empty #selected-items::before {

content: '⚠';

color: #e63946;

}

/\* Adicionando animação de destaque \*/

@keyframes highlightSummary {

0% { transform: scale(1); }

50% { transform: scale(1.02); }

100% { transform: scale(1); }

}

.selection-summary.highlight {

animation: highlightSummary 0.5s ease;

}

/\* Responsividade \*/

@media (max-width: 768px) {

.selection-summary {

padding: 18px 20px;

margin: 20px 0;

}

.selection-summary h4 {

font-size: 1rem;

}

#selected-items {

font-size: 0.95rem;

padding: 10px 12px;

}

}

/\* Estado desabilitado \*/

input[type="checkbox"]:disabled {

cursor: not-allowed;

}

@keyframes pulse {

0% { transform: scale(1); }

50% { transform: scale(1.02); }

100% { transform: scale(1); }

}

@keyframes shake {

0%, 100% { transform: translateX(0); }

25% { transform: translateX(-5px); }

75% { transform: translateX(5px); }

}

.workshop-item.selected {

border: 2px solid #2a9d8f;

background-color: #f8f9fa;

}

/\* ===== PÁGINA DE CONTATOS ===== \*/

/\* Banner da página de contatos \*/

.contact-banner {

background: linear-gradient(rgba(0, 0, 0, 0.7), rgba(0, 0, 0, 0.7)), url('../img/contact-banner.jpg');

background-size: cover;

background-position: center;

color: white;

padding: 100px 0;

text-align: center;

margin-bottom: 50px;

}

.contact-banner h1 {

font-size: 2.5rem;

margin-bottom: 20px;

font-weight: 700;

}

.contact-banner p {

font-size: 1.2rem;

max-width: 600px;

margin: 0 auto;

}

/\* Grid de informações de contato \*/

.contact-info-section {

margin-bottom: 60px;

}

.contact-grid {

display: grid;

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

gap: 25px;

margin-top: 30px;

}

.contact-card {

background: var(--light-color);

padding: 30px;

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

text-align: center;

transition: var(--transition);

display: flex;

flex-direction: column;

align-items: center;

}

.contact-card:hover {

transform: translateY(-5px);

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);

}

.contact-icon {

width: 70px;

height: 70px;

background-color: rgba(180, 148, 52, 0.1);

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

margin-bottom: 20px;

color: var(--secondary-color);

font-size: 1.8rem;

}

.contact-card h3 {

font-size: 1.3rem;

margin-bottom: 10px;

color: var(--primary-color);

}

.contact-card p {

color: var(--text-color);

font-weight: 600;

margin-bottom: 10px;

font-size: 1.1rem;

}

.contact-detail {

color: var(--text-light);

font-size: 0.9rem;

}

/\* Seção do formulário de contato \*/

.contact-form-section {

display: grid;

grid-template-columns: 2fr 1fr;

gap: 40px;

margin-bottom: 60px;

}

.form-container {

background: var(--light-color);

padding: 30px;

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

}

.form-header {

text-align: center;

margin-bottom: 30px;

}

.form-header h2 {

font-size: 1.8rem;

color: var(--primary-color);

margin-bottom: 10px;

}

.form-header p {

color: var(--text-light);

}

.contact-form textarea {

width: 100%;

padding: 12px 15px;

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

border-radius: var(--border-radius);

font-size: 1rem;

transition: var(--transition);

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

color: var(--text-color);

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

resize: vertical;

}

.contact-form textarea:focus {

outline: none;

border-color: var(--secondary-color);

box-shadow: 0 0 0 2px rgba(180, 148, 52, 0.2);

}

.contact-extra-info {

display: flex;

flex-direction: column;

gap: 20px;

}

.contact-extra-info .info-card {

background: var(--light-color);

padding: 25px;

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

}

.contact-extra-info .info-card h3 {

font-size: 1.2rem;

color: var(--primary-color);

margin-bottom: 15px;

display: flex;

align-items: center;

gap: 10px;

}

.contact-extra-info .info-card p {

color: var(--text-light);

margin-bottom: 8px;

font-size: 0.95rem;

}

/\* FAQ Section \*/

.faq-section {

margin-bottom: 60px;

}

.faq-container {

margin-top: 30px;

}

.faq-item {

background: var(--light-color);

border-radius: var(--border-radius);

margin-bottom: 15px;

overflow: hidden;

box-shadow: var(--box-shadow);

}

.faq-question {

padding: 20px;

display: flex;

justify-content: space-between;

align-items: center;

cursor: pointer;

transition: var(--transition);

}

.faq-question:hover {

background-color: rgba(180, 148, 52, 0.05);

}

.faq-question h3 {

font-size: 1.1rem;

color: var(--primary-color);

margin: 0;

}

.faq-question i {

color: var(--secondary-color);

transition: transform 0.3s ease;

}

.faq-answer {

padding: 0 20px;

max-height: 0;

overflow: hidden;

transition: max-height 0.3s ease, padding 0.3s ease;

}

.faq-answer p {

color: var(--text-light);

line-height: 1.6;

padding-bottom: 20px;

}

.faq-item.active .faq-answer {

max-height: 200px;

padding: 0 20px;

}

.faq-item.active .faq-question i {

transform: rotate(180deg);

}

/\* Responsividade \*/

@media (max-width: 992px) {

.contact-form-section {

grid-template-columns: 1fr;

gap: 30px;

}

.contact-extra-info {

flex-direction: row;

}

}

@media (max-width: 768px) {

.contact-banner {

padding: 80px 0;

}

.contact-banner h1 {

font-size: 2rem;

}

.contact-banner p {

font-size: 1rem;

}

.contact-grid {

grid-template-columns: 1fr;

gap: 20px;

}

.contact-card {

padding: 25px 20px;

}

.contact-extra-info {

flex-direction: column;

}

.form-container {

padding: 25px 20px;

}

}

@media (max-width: 480px) {

.contact-banner {

padding: 60px 0;

}

.contact-banner h1 {

font-size: 1.8rem;

}

.faq-question {

padding: 15px;

}

.faq-question h3 {

font-size: 1rem;

}

.faq-answer p {

font-size: 0.9rem;

}

}

/\* Animação para a seção About Banner \*/

.about-banner {

background: url('../img/banner.jpg') center/cover no-repeat;

padding: 80px 20px;

text-align: center;

color: #fff;

overflow: hidden;

position: relative;

animation: zoomIn 6s ease-in-out infinite alternate;

}

.about-banner .container {

position: relative;

z-index: 2;

}

.about-banner h1 {

font-size: 3rem;

opacity: 0;

transform: translateY(30px);

animation: fadeUp 1.2s ease forwards;

}

.about-banner p {

font-size: 1.3rem;

margin-top: 10px;

opacity: 0;

transform: translateY(20px);

animation: fadeUp 1.2s ease forwards 0.5s; /\* Delay de 0.5s \*/

}

/\* Animação para subir e aparecer \*/

@keyframes fadeUp {

to {

opacity: 1;

transform: translateY(0);

}

}

/\* Animação leve no fundo \*/

@keyframes zoomIn {

from {

transform: scale(1);

}

to {

transform: scale(1.05);

}

}

.hero-divider {

width: 100%;

height: 40px;

background: linear-gradient(to right, #f7b900, #E05D00);

box-shadow: 0 5px 15px rgba(0,0,0,0.2);

border-radius: 0 0 20px 20px;

margin-bottom: 2rem;

}

/\* Adicione ao main.css \*/

.service-item {

margin-left: 20px;

padding: 8px 0;

border-bottom: 1px dashed #eee;

}

.service-details {

font-size: 0.85em;

color: #666;

margin-left: 25px;

margin-top: 4px;

font-style: italic;

}

.total-amount {

font-size: 1.1em;

font-weight: bold;

color: #2a9d8f;

margin-top: 15px;

padding-top: 10px;

border-top: 2px solid #eee;

}

.confirmation-details p {

margin-bottom: 12px;

line-height: 1.5;

}

.confirmation-details strong {

color: #1d3557;

min-width: 120px;

display: inline-block;

}

/\* ===== INPUT DE DATA E ESTADOS DE HORÁRIOS - ESTILO ATUALIZADO E OTIMIZADO ===== \*/

/\* Estilos Base para Input de Data \*/

input[type="date"] {

width: 100%;

padding: 16px 20px;

border: 2px solid var(--border-color);

border-radius: 16px;

font-size: 1rem;

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

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

color: var(--text-color);

transition: all 0.3s ease;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.05);

cursor: pointer;

outline: none;

}

/\* Ícone Personalizado para Input de Data \*/

input[type="date"] {

background-image: url("data:image/svg+xml,%3Csvg xmlns=\'http://www.w3.org/2000/svg\' width=\'24\' height=\'24\' viewBox=\'0 0 24 24\' fill=\'none\' stroke=\'%23b49434\' stroke-width=\'2\' stroke-linecap=\'round\' stroke-linejoin=\'round\'%3E%3Crect x=\'3\' y=\'4\' width=\'18\' height=\'18\' rx=\'2\' ry=\'2\'%3E%3C/rect%3E%3Cline x1=\'16\' y1=\'2\' x2=\'16\' y2=\'6\'%3E%3C/line%3E%3Cline x1=\'8\' y1=\'2\' x2=\'8\' y2=\'6\'%3E%3C/line%3E%3Cline x1=\'3\' y1=\'10\' x2=\'21\' y2=\'10\'%3E%3C/line%3E%3C/svg%3E");

background-repeat: no-repeat;

background-position: right 18px center;

background-size: 20px;

padding-right: 50px;

}

/\* Efeitos de Interação (Hover e Focus) \*/

input[type="date"]:hover {

border-color: rgba(180, 148, 52, 0.4);

box-shadow: 0 4px 12px rgba(180, 148, 52, 0.1);

}

input[type="date"]:focus {

border-color: var(--secondary-color);

box-shadow: 0 0 0 4px rgba(180, 148, 52, 0.15);

transform: translateY(-2px);

}

/\* Ocultar Ícone Padrão do Calendário \*/

input[type="date"]::-webkit-calendar-picker-indicator {

opacity: 0;

cursor: pointer;

position: absolute;

right: 0;

width: 100%;

height: 100%;

}

/\* Estilização do Calendário Nativo (WebKit) \*/

input[type="date"]::-webkit-datetime-edit {

color: var(--text-color);

font-weight: 500;

}

input[type="date"]::-webkit-datetime-edit-fields-wrapper {

padding: 2px;

}

input[type="date"]::-webkit-datetime-edit-text {

color: var(--secondary-color);

padding: 0 4px;

}

/\* ===== ESTILOS PARA SELEÇÃO DE HORÁRIOS ===== \*/

/\* Opções Desabilitadas (Horários Indisponíveis) \*/

select option:disabled {

color: #a0aec0;

background-color: #f7fafc;

font-style: italic;

}

/\* Indicador de Carregamento de Horários \*/

.loading-horarios {

text-align: center;

padding: 20px;

color: var(--text-light);

font-size: 0.95rem;

display: flex;

flex-direction: column;

align-items: center;

gap: 12px;

animation: pulse 1.5s ease-in-out infinite; /\* Animação aplicada diretamente \*/

}

.loading-horarios::before {

content: '';

width: 24px;

height: 24px;

border: 2px solid var(--border-color);

border-top: 2px solid var(--secondary-color);

border-radius: 50%;

animation: spin 1s linear infinite;

}

/\* Mensagem de Nenhum Horário Disponível \*/

.no-horarios {

text-align: center;

padding: 20px;

color: #e53e3e;

font-style: italic;

background-color: rgba(229, 62, 62, 0.05);

border-radius: 12px;

border: 1px dashed rgba(229, 62, 62, 0.3);

display: flex;

flex-direction: column;

align-items: center;

gap: 8px;

}

.no-horarios::before {

content: '⏰';

font-size: 1.5rem;

font-style: normal;

}

/\* Container do Select de Horários \*/

.time-select-container {

position: relative;

margin-bottom: 20px;

}

.time-select-container select {

width: 100%;

padding: 16px 20px;

border: 2px solid var(--border-color);

border-radius: 16px;

font-size: 1rem;

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

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

color: var(--text-color);

transition: all 0.3s ease;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.05);

appearance: none;

background-image: url("data:image/svg+xml,%3Csvg xmlns=\'http://www.w3.org/2000/svg\' width=\'20\' height=\'20\' viewBox=\'0 0 24 24\' fill=\'none\' stroke=\'%23b49434\' stroke-width=\'2\' stroke-linecap=\'round\' stroke-linejoin=\'round\'%3E%3Cpath d=\'m6 9 6 6 6-6\'%3E%3C/path\'%3E%3C/svg%3E");

background-repeat: no-repeat;

background-position: right 18px center;

background-size: 20px;

padding-right: 50px;

cursor: pointer;

}

.time-select-container select:focus {

border-color: var(--secondary-color);

box-shadow: 0 0 0 4px rgba(180, 148, 52, 0.15);

outline: none;

}

/\* Badges de Status de Horário \*/

.horario-disponivel,

.horario-indisponivel {

display: inline-block;

padding: 4px 8px;

border-radius: 6px;

font-size: 0.85rem;

font-weight: 500;

margin-left: 8px;

}

.horario-disponivel {

background-color: rgba(72, 187, 120, 0.1);

color: #48bb78;

}

.horario-indisponivel {

background-color: rgba(229, 62, 62, 0.1);

color: #e53e3e;

}

/\* ===== ANIMAÇÕES ===== \*/

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

@keyframes pulse {

0% { opacity: 1; }

50% { opacity: 0.5; }

100% { opacity: 1; }

}

/\* ===== RESPONSIVIDADE ===== \*/

@media (max-width: 768px) {

input[type="date"],

.time-select-container select {

padding: 14px 16px;

font-size: 16px;

background-position: right 16px center;

padding-right: 45px;

}

.loading-horarios,

.no-horarios {

padding: 16px;

font-size: 0.9rem;

}

}

/\* ===== SUPORTE A DARK MODE ===== \*/

@media (prefers-color-scheme: dark) {

input[type="date"] {

background-image: url("data:image/svg+xml,%3Csvg xmlns=\'http://www.w3.org/2000/svg\' width=\'24\' height=\'24\' viewBox=\'0 0 24 24\' fill=\'none\' stroke=\'%23d4b15f\' stroke-width=\'2\' stroke-linecap=\'round\' stroke-linejoin=\'round\'%3E%3Crect x=\'3\' y=\'4\' width=\'18\' height=\'18\' rx=\'2\' ry=\'2\'%3E%3C/rect%3E%3Cline x1=\'16\' y1=\'2\' x2=\'16\' y2=\'6\'%3E%3C/line\'%3E%3Cline x1=\'8\' y1=\'2\' x2=\'8\' y2=\'6\'%3E%3C/line\'%3E%3Cline x1=\'3\' y1=\'10\' x2=\'21\' y2=\'10\'%3E%3C/line\'%3E%3C/svg\'%3E");

}

select option:disabled {

color: #718096;

background-color: #2d3748;

}

.no-horarios {

background-color: rgba(229, 62, 62, 0.1);

border-color: rgba(229, 62, 62, 0.2);

}

}

/\* Barra de progresso simplificada \*/

.progress-container {

margin: 30px 0;

padding: 0 20px;

}

.progress-bar {

display: flex;

justify-content: space-between;

align-items: center;

position: relative;

max-width: 800px;

margin: 0 auto;

}

.progress-step {

display: flex;

flex-direction: column;

align-items: center;

position: relative;

z-index: 2;

}

.step-bubble {

width: 36px;

height: 36px;

border-radius: 50%;

background-color: #e9ecef;

color: #6c757d;

display: flex;

align-items: center;

justify-content: center;

font-weight: 600;

transition: all 0.3s ease;

border: 2px solid #e9ecef;

}

.step-label {

margin-top: 8px;

font-size: 14px;

color: #6c757d;

font-weight: 500;

transition: all 0.3s ease;

}

.progress-step.active .step-bubble {

background-color: #1d3557;

border-color: #1d3557;

color: white;

box-shadow: 0 0 0 4px rgba(29, 53, 87, 0.2);

}

.progress-step.active .step-label {

color: #1d3557;

font-weight: 600;

}

.progress-step.completed .step-bubble {

background-color: #2a9d8f;

border-color: #2a9d8f;

color: white;

}

.progress-line {

position: absolute;

top: 18px;

left: 0;

right: 0;

height: 2px;

background-color: #e9ecef;

z-index: 1;

}

/\* Responsividade \*/

@media (max-width: 768px) {

.progress-bar {

padding: 0 10px;

}

.step-bubble {

width: 32px;

height: 32px;

font-size: 14px;

}

.step-label {

font-size: 12px;

text-align: center;

}

}

/\* Adicione estilos para a distância \*/

.workshop-item .distance {

color: #e63946;

font-weight: bold;

margin-bottom: 8px;

}

.workshop-popup .fa-route {

color: #e63946;

margin-right: 5px;

}

.workshop-item .distance .fa-route {

color: #e63946;

margin-right: 5px;

}

/\* Estilo para destacar a distância \*/

.distance-badge {

background-color: #e63946;

color: white;

padding: 4px 8px;

border-radius: 12px;

font-size: 12px;

font-weight: bold;

}

/\* Barra de rolagem personalizada (Chrome, Edge, Opera, Safari) \*/

::-webkit-scrollbar {

width: 10px;

height: 10px;

}

::-webkit-scrollbar-track {

background: #f1f1f1; /\* fundo do trilho \*/

}

::-webkit-scrollbar-thumb {

background: var(--secondary-color); /\* usa sua cor \*/

border-radius: 6px;

}

::-webkit-scrollbar-thumb:hover {

background: #8a6f26; /\* tom mais escuro ao passar o mouse \*/

}

/\* Barra de rolagem no Firefox \*/

\* {

scrollbar-width: thin;

scrollbar-color: var(--secondary-color) #384d5e;

}

/\* ===== RECOMMENDATION SECTION - ESTILOS MELHORADOS ===== \*/

.recommendation-container {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

gap: 25px;

margin: 30px 0;

}

.product-card {

background: var(--card-bg);

border-radius: 20px;

box-shadow: var(--card-shadow);

transition: all 0.3s ease;

border: 2px solid var(--border-color);

overflow: hidden;

position: relative;

}

.product-card:hover {

transform: translateY(-8px) scale(1.02);

box-shadow: 0 20px 35px rgba(0, 0, 0, 0.15);

border-color: rgba(180, 148, 52, 0.4);

}

.product-card-header {

background: linear-gradient(135deg, rgba(180, 148, 52, 0.1), rgba(212, 177, 95, 0.15));

padding: 15px 20px;

border-bottom: 1px solid rgba(180, 148, 52, 0.1);

}

.product-card-header h3 {

color: var(--secondary-color);

margin: 0;

font-size: 1.2rem;

font-weight: 600;

display: flex;

align-items: center;

gap: 10px;

}

.product-card-content {

padding: 20px;

display: flex;

flex-direction: column;

gap: 15px;

}

.product-image-container {

width: 100%;

height: 160px;

background: var(--bg-color);

border-radius: 12px;

display: flex;

align-items: center;

justify-content: center;

overflow: hidden;

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

position: relative;

}

.product-image {

max-width: 100%;

max-height: 100%;

object-fit: contain;

transition: all 0.4s ease;

}

.product-card:hover .product-image {

transform: scale(1.1);

}

.product-details {

display: flex;

flex-direction: column;

gap: 10px;

}

.product-specs {

display: flex;

flex-direction: column;

gap: 8px;

}

.product-specs p {

margin: 0;

font-size: 0.9rem;

color: var(--text-light);

align-items: flex-start;

gap: 8px;

}

.product-specs strong {

color: var(--primary-color);

min-width: 100px;

font-weight: 600;

}

.product-price {

margin-top: 10px;

padding-top: 10px;

border-top: 1px dashed var(--border-color);

display: flex;

justify-content: space-between;

align-items: center;

}

.price {

font-size: 1.2rem;

font-weight: 700;

color: var(--secondary-color);

}

.product-selection {

padding: 15px 20px;

background: rgba(180, 148, 52, 0.05);

border-top: 1px solid rgba(180, 148, 52, 0.1);

}

.checkbox-container {

display: flex;

align-items: center;

cursor: pointer;

font-weight: 500;

color: var(--primary-color);

transition: all 0.3s ease;

padding: 5px;

border-radius: 6px;

}

.checkbox-container:hover {

background: rgba(180, 148, 52, 0.1);

}

.checkbox-container input[type="checkbox"] {

display: none;

}

.checkmark {

width: 20px;

height: 20px;

background: #fff;

border: 2px solid var(--secondary-color);

border-radius: 4px;

margin-right: 12px;

position: relative;

transition: all 0.3s ease;

display: flex;

align-items: center;

justify-content: center;

}

.checkbox-container input[type="checkbox"]:checked + .checkmark {

background: var(--secondary-color);

border-color: var(--secondary-color);

}

.checkbox-container input[type="checkbox"]:checked + .checkmark::after {

content: "✓";

color: white;

font-size: 12px;

font-weight: bold;

}

.checkbox-container input[type="checkbox"]:disabled + .checkmark {

background: #ccc;

border-color: #999;

cursor: not-allowed;

}

.checkbox-container input[type="checkbox"]:disabled ~ span {

color: #999;

cursor: not-allowed;

}

/\* Estado quando não há recomendação \*/

.no-recommendation {

text-align: center;

padding: 30px 20px;

color: var(--text-light);

}

.no-recommendation i {

font-size: 2rem;

color: #ccc;

margin-bottom: 15px;

display: block;

}

.no-recommendation p {

margin: 0;

font-style: italic;

}

/\* Badge de recomendação automática \*/

.recommendation-badge {

position: absolute;

top: 15px;

right: 15px;

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: white;

padding: 4px 10px;

border-radius: 20px;

font-size: 0.7rem;

font-weight: 600;

box-shadow: 0 2px 8px rgba(180, 148, 52, 0.3);

z-index: 2;

}

/\* Responsividade \*/

@media (max-width: 768px) {

.recommendation-container {

grid-template-columns: 1fr;

gap: 20px;

}

.product-card-content {

padding: 15px;

}

.product-image-container {

height: 140px;

}

.product-specs p {

flex-direction: column;

gap: 4px;

}

.product-specs strong {

min-width: auto;

}

}

/\* Animações \*/

@keyframes productHighlight {

0% { box-shadow: 0 0 0 0 rgba(180, 148, 52, 0.4); }

70% { box-shadow: 0 0 0 10px rgba(180, 148, 52, 0); }

100% { box-shadow: 0 0 0 0 rgba(180, 148, 52, 0); }

}

.product-card.highlight {

animation: productHighlight 1s ease;

}

/\* ===== POPUPS DO LEAFLET - ESTILO ATUALIZADO ===== \*/

.leaflet-popup-content-wrapper {

background: linear-gradient(135deg, #ffffff 0%, #f8f9fa 100%);

color: #1d3557;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.15);

border-radius: 16px;

padding: 2px;

border: 2px solid rgba(180, 148, 52, 0.3);

overflow: hidden;

}

.leaflet-popup-content {

margin: 15px 18px;

line-height: 1.5;

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

font-size: 0.95rem;

}

.leaflet-popup-tip {

background: linear-gradient(135deg, #ffffff 0%, #f8f9fa 100%);

box-shadow: 0 3px 14px rgba(0, 0, 0, 0.2);

border: 1px solid rgba(180, 148, 52, 0.2);

}

/\* Conteúdo personalizado do popup \*/

.workshop-popup {

min-width: 250px;

padding: 5px;

}

.workshop-popup h4 {

color: #1d3557;

margin: 0 0 12px 0;

font-size: 1.1rem;

font-weight: 600;

padding-bottom: 8px;

border-bottom: 2px solid rgba(180, 148, 52, 0.3);

display: flex;

align-items: center;

gap: 8px;

}

.workshop-popup h4::before {

content: '🔧';

font-size: 1rem;

}

.workshop-popup p {

margin: 8px 0;

color: #384d5e;

display: flex;

align-items: center;

gap: 8px;

font-size: 0.9rem;

}

.workshop-popup i {

color: #b49434;

width: 16px;

text-align: center;

}

.workshop-popup .distance {

color: #e63946;

font-weight: 600;

}

.workshop-popup .distance i {

color: #e63946;

}

.select-workshop-btn {

background: linear-gradient(135deg, #1d3557, #2a4f6d);

color: white;

border: none;

padding: 10px 20px;

border-radius: 8px;

cursor: pointer;

font-weight: 600;

font-size: 0.9rem;

width: 100%;

margin-top: 12px;

transition: all 0.3s ease;

display: flex;

align-items: center;

justify-content: center;

gap: 8px;

}

.select-workshop-btn:hover {

background: linear-gradient(135deg, #2a4f6d, #1d3557);

transform: translateY(-2px);

box-shadow: 0 4px 12px rgba(29, 53, 87, 0.3);

}

.select-workshop-btn::before {

content: '✓';

font-weight: bold;

}

/\* Popup para localização do usuário \*/

.user-popup h4 {

color: #2a9d8f;

border-bottom-color: rgba(42, 157, 143, 0.3);

}

.user-popup h4::before {

content: '📍';

}

/\* Animações para os popups \*/

@keyframes popupAppear {

from {

opacity: 0;

transform: translateY(10px) scale(0.95);

}

to {

opacity: 1;

transform: translateY(0) scale(1);

}

}

.leaflet-popup {

animation: popupAppear 0.3s ease;

}

/\* Responsividade para mobile \*/

@media (max-width: 768px) {

.leaflet-popup-content-wrapper {

border-radius: 14px;

max-width: 280px;

}

.leaflet-popup-content {

margin: 12px 15px;

font-size: 0.9rem;

}

.workshop-popup {

min-width: 220px;

}

.workshop-popup h4 {

font-size: 1rem;

}

.select-workshop-btn {

padding: 8px 16px;

font-size: 0.85rem;

}

}

/\* Efeito de destaque quando o popup é aberto \*/

.leaflet-popup-content-wrapper:hover {

box-shadow: 0 15px 35px rgba(0, 0, 0, 0.2);

transform: translateY(-2px);

transition: all 0.3s ease;

}

/\* Estilo para o botão de fechar do popup \*/

.leaflet-container a.leaflet-popup-close-button {

color: #1d3557;

padding: 6px 8px 0 0;

font-size: 1.2rem;

transition: color 0.3s ease;

}

.leaflet-container a.leaflet-popup-close-button:hover {

color: #e63946;

}

/\* Gradiente sutil no fundo do popup \*/

.leaflet-popup-content-wrapper::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: linear-gradient(90deg, #b49434, #d4b15f);

opacity: 0.8;

z-index: 1;

}

/\* ===== ESTILOS DO DROPDOWN DO USUÁRIO ===== \*/

.user-dropdown {

position: relative;

cursor: pointer;

}

.user-ico {

display: flex;

align-items: center;

gap: 8px;

padding: 10px 15px;

border-radius: 8px;

transition: all 0.3s ease;

text-decoration: none;

color: var(--f8f9fa);

}

.user-ico:hover {

background-color: rgba(255, 255, 255, 0.1);

}

.login-status {

font-size: 0.9rem;

font-weight: 500;

}

.dropdown-menu {

position: absolute;

top: 100%;

right: 0;

width: 250px;

background: white;

border-radius: 8px;

box-shadow: 0 4px 20px rgba(0, 0, 0, 0.15);

padding: 15px 0;

z-index: 1000;

display: none;

animation: fadeIn 0.2s ease;

margin-top: 10px;

}

.dropdown-menu.show {

display: block;

}

.user-info {

padding: 0 15px 10px;

border-bottom: 1px solid #eee;

margin-bottom: 10px;

}

.user-info strong {

display: block;

color: #333;

font-size: 14px;

margin-bottom: 2px;

}

.user-info span {

color: #666;

font-size: 12px;

}

.dropdown-divider {

height: 1px;

background: #eee;

margin: 8px 0;

}

.dropdown-item {

display: flex;

align-items: center;

gap: 10px;

padding: 10px 15px;

color: #333;

text-decoration: none;

transition: background-color 0.3s ease;

font-size: 14px;

}

.dropdown-item:hover {

background-color: #f5f5f5;

}

.dropdown-item i {

width: 16px;

color: #666;

}

#logout-btn {

color: #e74c3c;

}

#logout-btn i {

color: #e74c3c;

}

#logout-btn:hover {

background-color: #fee;

}

/\* Quando logado \*/

.user-logged-in .user-ico {

color: #FF6B00;

}

.user-logged-in .user-ico i {

color: #FF6B00;

}

/\* Animações \*/

@keyframes fadeIn {

from { opacity: 0; transform: translateY(-10px); }

to { opacity: 1; transform: translateY(0); }

}

/\* Responsividade \*/

@media (max-width: 768px) {

.dropdown-menu {

position: fixed;

top: auto;

bottom: 0;

left: 0;

width: 100%;

border-radius: 16px 16px 0 0;

transform: translateY(100%);

transition: transform 0.3s ease;

}

.dropdown-menu.show {

transform: translateY(0);

}

.user-icon .login-status {

display: inline-block !important;

font-size: 0.8rem;

color: white;

}

}

.alert {

padding: 12px;

margin: 10px 0;

border-radius: 6px;

font-size: 14px;

font-weight: 500;

text-align: center;

}

.alert.error {

background: #f8d7da;

color: #721c24;

border: 1px solid #f5c6cb;

}

.alert.success {

background: #d4edda;

color: #155724;

border: 1px solid #c3e6cb;

}

/\* ===== MELHORIAS PARA A PÁGINA AGENDA ===== \*/

.agenda-container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

.filter-container {

display: flex;

gap: 20px;

margin-bottom: 30px;

flex-wrap: wrap;

align-items: end;

}

.filter-group {

display: flex;

flex-direction: column;

min-width: 200px;

}

.filter-group label {

margin-bottom: 8px;

font-weight: 600;

color: var(--primary-color);

}

.filter-group select {

padding: 12px 15px;

border: 2px solid var(--border-color);

border-radius: 10px;

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

color: var(--text-color);

font-size: 1rem;

transition: var(--transition);

}

.filter-group select:focus {

border-color: var(--secondary-color);

outline: none;

}

.no-appointments {

text-align: center;

padding: 60px 20px;

color: var(--text-light);

}

.no-appointments i {

font-size: 4rem;

color: var(--border-color);

margin-bottom: 20px;

display: block;

}

.no-appointments h3 {

font-size: 1.5rem;

margin-bottom: 10px;

color: var(--text-color);

}

.price-info {

background: rgba(180, 148, 52, 0.05);

padding: 15px;

border-radius: 10px;

margin-top: 10px;

}

.price-info p {

margin: 0;

font-weight: 600;

color: var(--secondary-color);

display: flex;

align-items: center;

gap: 8px;

}

.appointment-header {

padding: 20px;

background: var(--light-color);

display: flex;

justify-content: space-between;

align-items: center;

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

}

.protocol {

font-weight: 600;

color: var(--primary-color);

font-size: 0.9rem;

}

.status {

padding: 6px 12px;

border-radius: 20px;

font-size: 0.8rem;

font-weight: 600;

}

.status.pending {

background: rgba(255, 193, 7, 0.2);

color: #856404;

border: 1px solid rgba(255, 193, 7, 0.3);

}

.status.completed {

background: rgba(40, 167, 69, 0.2);

color: #155724;

border: 1px solid rgba(40, 167, 69, 0.3);

}

.status.cancelled {

background: rgba(220, 53, 69, 0.2);

color: #721c24;

border: 1px solid rgba(220, 53, 69, 0.3);

}

.appointment-body {

padding: 25px;

display: grid;

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

gap: 20px;

}

.service-info h3 {

color: var(--secondary-color);

margin-bottom: 15px;

font-size: 1.2rem;

}

.service-info p, .date-info p, .location-info p {

margin: 8px 0;

display: flex;

align-items: center;

gap: 10px;

color: var(--text-color);

}

.service-info i, .date-info i, .location-info i {

color: var(--secondary-color);

width: 16px;

}

.appointment-footer {

padding: 20px;

background: var(--light-color);

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

display: flex;

gap: 15px;

flex-wrap: wrap;

}

/\* Responsividade \*/

@media (max-width: 768px) {

.filter-container {

flex-direction: column;

gap: 15px;

}

.filter-group {

min-width: 100%;

}

.appointment-header {

flex-direction: column;

gap: 15px;

align-items: flex-start;

}

.appointment-body {

grid-template-columns: 1fr;

gap: 15px;

padding: 20px;

}

.appointment-footer {

flex-direction: column;

}

.appointment-footer .btn {

width: 100%;

justify-content: center;

}

}

/\* ===============================

FILTROS E CONTROLES - VERSÃO RESPONSIVA

=================================\*/

.filter-controls {

display: flex;

gap: 20px;

align-items: center;

flex-wrap: wrap;

margin-bottom: 25px;

padding: 15px 20px;

background: #ffffff;

border-radius: 12px;

border: 1px solid #e1e5eb;

box-shadow: 0 2px 6px rgba(0,0,0,0.05);

transition: all 0.3s ease;

}

.filter-controls:hover {

box-shadow: 0 4px 12px rgba(0,0,0,0.08);

}

.filter-group {

display: flex;

flex-direction: column;

gap: 6px;

min-width: 160px;

flex: 1;

}

.filter-group label {

font-weight: 600;

font-size: 13px;

color: #495057;

margin-bottom: 2px;

letter-spacing: 0.3px;

}

.filter-select {

padding: 10px 12px;

border: 1px solid #d0d7de;

border-radius: 8px;

background: #fdfdfd;

font-size: 14px;

transition: border 0.2s ease, box-shadow 0.2s ease;

width: 100%;

box-sizing: border-box;

}

.filter-select:focus {

outline: none;

border-color: #0d6efd;

box-shadow: 0 0 0 3px rgba(13,110,253,0.15);

}

/\* Botão de ações dos filtros \*/

.filter-actions {

display: flex;

gap: 10px;

align-items: flex-end;

flex-wrap: wrap;

}

/\* ===============================

BOTÕES

=================================\*/

.btn-small {

padding: 7px 14px;

font-size: 13px;

border-radius: 6px;

font-weight: 600;

transition: all 0.2s ease;

border: none;

cursor: pointer;

white-space: nowrap;

}

.btn-primary {

color: white;

}

.btn-primary:hover {

transform: translateY(-1px);

}

.btn-secondary {

background: #6c757d;

color: white;

}

.btn-secondary:hover {

background: #5c636a;

transform: translateY(-1px);

}

.btn-danger {

background: #dc3545;

border: 1px solid #dc3545;

color: white;

}

.btn-danger:hover {

background: #bb2d3b;

border-color: #b02a37;

transform: translateY(-1px);

}

/\* ===============================

STATUS

=================================\*/

.status-pending {

background: #fff8e1;

color: #8a6d3b;

border: 1px solid #ffe8a1;

padding: 6px 12px;

border-radius: 6px;

font-size: 13px;

font-weight: 600;

display: inline-block;

}

.status-completed {

background: #e6f7ff;

color: #055160;

border: 1px solid #b6effb;

padding: 6px 12px;

border-radius: 6px;

font-size: 13px;

font-weight: 600;

display: inline-block;

}

.status-cancelled {

background: #fde2e4;

color: #842029;

border: 1px solid #f5c2c7;

padding: 6px 12px;

border-radius: 6px;

font-size: 13px;

font-weight: 600;

display: inline-block;

}

.status-expired {

background: #f1f3f5;

color: #495057;

border: 1px solid #dee2e6;

padding: 6px 12px;

border-radius: 6px;

font-size: 13px;

font-weight: 600;

display: inline-block;

}

/\* ===============================

INFOS

=================================\*/

.action-disabled {

color: #6c757d;

font-style: italic;

font-size: 14px;

}

.cancel-info {

background: #fff5f5;

padding: 12px 14px;

border-radius: 6px;

border-left: 4px solid #dc3545;

margin-top: 12px;

}

.cancel-info p {

margin: 0;

color: #842029;

font-size: 14px;

}

.price-info {

background: #ebf5ff;

padding: 10px 14px;

border-radius: 6px;

margin-top: 12px;

}

.price-info p {

margin: 0;

font-weight: 600;

color: #004085;

}

/\* ===============================

VAZIO

=================================\*/

.no-appointments {

text-align: center;

padding: 50px 20px;

color: #6c757d;

}

.no-appointments i {

font-size: 50px;

margin-bottom: 12px;

color: #adb5bd;

}

/\* ===============================

RESPONSIVIDADE MELHORADA

=================================\*/

/\* Tablets e telas médias (768px - 1024px) \*/

@media (max-width: 1024px) {

.filter-controls {

gap: 15px;

padding: 12px 16px;

}

.filter-group {

min-width: 140px;

}

}

/\* Tablets pequenos (576px - 768px) \*/

@media (max-width: 768px) {

.filter-controls {

flex-direction: column;

align-items: stretch;

gap: 12px;

padding: 16px;

}

.filter-group {

width: 100%;

min-width: unset;

}

.filter-select {

width: 100%;

padding: 12px 14px;

font-size: 16px; /\* Evita zoom no iOS \*/

}

.filter-actions {

width: 100%;

justify-content: center;

margin-top: 8px;

}

.btn-small {

flex: 1;

min-width: 120px;

max-width: 200px;

text-align: center;

padding: 10px 16px;

}

}

/\* Smartphones (até 576px) \*/

@media (max-width: 576px) {

.filter-controls {

margin-bottom: 20px;

padding: 12px;

border-radius: 8px;

}

.filter-group {

margin-bottom: 4px;

}

.filter-group label {

font-size: 14px;

margin-bottom: 4px;

}

.filter-select {

padding: 14px 12px;

font-size: 16px;

border-radius: 6px;

}

.filter-actions {

flex-direction: column;

gap: 8px;

margin-top: 12px;

}

.btn-small {

width: 100%;

max-width: none;

text-align: center;

padding: 12px 16px;

font-size: 14px;

}

.status-pending,

.status-completed,

.status-cancelled,

.status-expired {

display: block;

text-align: center;

margin: 4px 0;

width: 100%;

box-sizing: border-box;

}

.cancel-info,

.price-info {

margin-top: 16px;

padding: 16px 12px;

}

.no-appointments {

padding: 40px 16px;

}

.no-appointments i {

font-size: 40px;

}

}

/\* Smartphones muito pequenos (até 360px) \*/

@media (max-width: 360px) {

.filter-controls {

padding: 10px;

gap: 10px;

}

.filter-select {

padding: 12px 10px;

font-size: 16px;

}

.btn-small {

padding: 10px 12px;

font-size: 13px;

}

.filter-group label {

font-size: 13px;

}

}

/\* ===============================

MELHORIAS DE ACESSIBILIDADE

=================================\*/

/\* Foco visível para navegação por teclado \*/

.filter-select:focus,

.btn-small:focus {

outline: 2px solid #0d6efd;

outline-offset: 2px;

}

/\* Estados de hover para dispositivos que suportam \*/

@media (hover: hover) {

.filter-select:hover {

border-color: #0d6efd;

}

.btn-small:hover {

transform: translateY(-1px);

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

}

}

/\* Redução de movimento para usuários que preferem \*/

@media (prefers-reduced-motion: reduce) {

.filter-controls,

.filter-select,

.btn-small {

transition: none;

}

.btn-small:hover {

transform: none;

}

}

/\* ===============================

LAYOUT FLEXÍVEL PARA FILTROS

=================================\*/

/\* Container flexível para múltiplos filtros \*/

.filters-row {

display: flex;

flex-wrap: wrap;

gap: 15px;

align-items: flex-end;

}

.filters-row .filter-group {

flex: 1;

min-width: 160px;

}

@media (max-width: 768px) {

.filters-row {

flex-direction: column;

gap: 12px;

}

.filters-row .filter-group {

width: 100%;

min-width: unset;

}

}

/\* Layout em grid para filtros complexos \*/

.filters-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(160px, 1fr));

gap: 15px;

align-items: end;

}

@media (max-width: 768px) {

.filters-grid {

grid-template-columns: 1fr;

gap: 12px;

}

}

@media (max-width: 576px) {

.filters-grid {

gap: 10px;

}

}

/\* ===== MELHORIAS PARA O HEADER DA SEÇÃO DE HISTÓRICO ===== \*/

.section-header {

display: block; /\* Muda de flex para block \*/

margin-bottom: 25px;

}

.section-header h2 {

margin-bottom: 20px;

font-size: 1.5rem;

color: var(--primary-color);

display: flex;

align-items: center;

gap: 10px;

padding-bottom: 10px;

border-bottom: 2px solid rgba(180, 148, 52, 0.2);

}

.filter-controls {

display: flex;

gap: 20px;

align-items: flex-end;

flex-wrap: wrap;

padding: 15px 20px;

background: #ffffff;

border-radius: 12px;

border: 1px solid #e1e5eb;

box-shadow: 0 2px 6px rgba(0,0,0,0.05);

transition: all 0.3s ease;

}

/\* Responsividade para o novo layout \*/

@media (max-width: 768px) {

.section-header h2 {

font-size: 1.3rem;

text-align: center;

}

.filter-controls {

flex-direction: column;

align-items: stretch;

gap: 12px;

padding: 16px;

}

.filter-group {

width: 100%;

min-width: unset;

}

.filter-select {

width: 100%;

padding: 12px 14px;

font-size: 16px;

}

.filter-actions {

width: 100%;

justify-content: center;

margin-top: 8px;

}

.btn-small {

flex: 1;

min-width: 120px;

max-width: 200px;

text-align: center;

padding: 10px 16px;

}

}

@media (max-width: 576px) {

.section-header h2 {

font-size: 1.2rem;

margin-bottom: 15px;

}

.filter-controls {

margin-bottom: 20px;

padding: 12px;

border-radius: 8px;

}

}

/\* Estilos para o modal de cancelamento \*/

.modal-backdrop {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

display: flex;

justify-content: center;

align-items: center;

z-index: 10000;

}

.modal-content {

background: white;

padding: 30px;

border-radius: 10px;

width: 90%;

max-width: 500px;

box-shadow: 0 5px 15px rgba(0,0,0,0.3);

}

.modal-title {

margin-bottom: 15px;

color: #333;

}

.modal-text {

margin-bottom: 15px;

color: #666;

}

.modal-textarea {

width: 100%;

height: 100px;

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

resize: vertical;

font-family: inherit;

}

.modal-buttons {

margin-top: 20px;

display: flex;

gap: 10px;

justify-content: flex-end;

}

.modal-cancel-btn {

padding: 10px 20px;

border: 1px solid #ddd;

background: white;

border-radius: 5px;

cursor: pointer;

}

.modal-confirm-btn {

padding: 10px 20px;

background: #dc3545;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

/\* Loading overlay \*/

.loading-overlay {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

display: flex;

justify-content: center;

align-items: center;

z-index: 10001;

color: white;

font-size: 18px;

}

.loading-content {

text-align: center;

}

.spinner {

border: 4px solid #f3f3f3;

border-top: 4px solid #3498db;

border-radius: 50%;

width: 50px;

height: 50px;

animation: spin 1s linear infinite;

margin: 0 auto 15px;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

/\* ===== MELHORIAS PARA A PÁGINA DE AGENDAMENTOS ===== \*/

/\* Estilos aprimorados para os cards de agendamento \*/

.appointment-card {

background: var(--card-bg);

border-radius: 16px;

box-shadow: 0 8px 25px rgba(0, 0, 0, 0.08);

margin-bottom: 25px;

overflow: hidden;

transition: all 0.3s ease;

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

position: relative;

}

.appointment-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

width: 6px;

height: 100%;

background: var(--secondary-color);

opacity: 0;

transition: opacity 0.3s ease;

}

.appointment-card.completed::before {

background: var(--success-color);

opacity: 1;

}

.appointment-card.pending::before {

background: var(--warning-color);

opacity: 1;

}

.protocol {

font-weight: 600;

color: var(--primary-color);

font-size: 0.95rem;

letter-spacing: 0.5px;

}

.status {

padding: 8px 16px;

border-radius: 20px;

font-size: 0.85rem;

font-weight: 600;

text-transform: uppercase;

letter-spacing: 0.5px;

}

.status.completed {

background: rgba(40, 167, 69, 0.15);

color: var(--success-color);

border: 1px solid rgba(40, 167, 69, 0.3);

}

.status.pending {

background: rgba(255, 193, 7, 0.15);

color: var(--warning-color);

border: 1px solid rgba(255, 193, 7, 0.3);

}

.status.cancelled {

background: rgba(220, 53, 69, 0.15);

color: var(--danger-color);

border: 1px solid rgba(220, 53, 69, 0.3);

}

.appointment-body {

padding: 25px;

display: grid;

grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));

gap: 25px;

}

.service-info h3 {

color: var(--secondary-color);

margin-bottom: 15px;

font-size: 1.3rem;

font-weight: 600;

display: flex;

align-items: center;

gap: 10px;

}

.service-info h3::before {

content: '🔧';

font-size: 1.2rem;

}

.service-info p, .date-info p, .location-info p {

margin: 10px 0;

display: flex;

align-items: center;

gap: 12px;

color: var(--text-color);

font-size: 0.95rem;

}

.service-info i, .date-info i, .location-info i {

color: var(--secondary-color);

width: 18px;

text-align: center;

font-size: 1rem;

}

.price-info {

background: linear-gradient(135deg, rgba(180, 148, 52, 0.08), rgba(212, 177, 95, 0.12));

padding: 15px 20px;

border-radius: 12px;

margin-top: 15px;

border: 1px solid rgba(180, 148, 52, 0.15);

}

.price-info p {

margin: 0;

font-weight: 700;

color: var(--secondary-color);

font-size: 1.1rem;

display: flex;

align-items: center;

gap: 10px;

}

.cancel-info {

background: rgba(220, 53, 69, 0.08);

padding: 15px 20px;

border-radius: 12px;

margin-top: 15px;

border-left: 4px solid var(--danger-color);

}

.cancel-info p {

margin: 0;

color: var(--danger-color);

font-weight: 500;

display: flex;

align-items: center;

gap: 10px;

}

.appointment-footer {

padding: 20px 25px;

background: var(--light-color);

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

display: flex;

gap: 15px;

flex-wrap: wrap;

justify-content: flex-end;

}

.appointment-footer .btn {

display: flex;

align-items: center;

gap: 8px;

padding: 10px 20px;

border-radius: 8px;

font-weight: 500;

transition: all 0.3s ease;

}

.appointment-footer .btn:hover {

transform: translateY(-2px);

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);

}

.text-muted {

color: var(--text-light) !important;

font-style: italic;

padding: 10px 0;

display: flex;

align-items: center;

gap: 8px;

}

/\* Botão de download de comprovante destacado \*/

.btn-download {

background: linear-gradient(135deg, var(--success-color), #20c997);

color: white;

border: none;

}

.btn-download:hover {

background: linear-gradient(135deg, #20c997, var(--success-color));

}

/\* Estilos para o modal de confirmação de download \*/

.download-modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

z-index: 10000;

justify-content: center;

align-items: center;

}

.download-modal.show {

display: flex;

}

.download-modal-content {

background: white;

padding: 30px;

border-radius: 16px;

max-width: 500px;

width: 90%;

text-align: center;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.3);

animation: modalAppear 0.3s ease;

}

@keyframes modalAppear {

from {

opacity: 0;

transform: scale(0.9) translateY(-20px);

}

to {

opacity: 1;

transform: scale(1) translateY(0);

}

}

.download-icon {

font-size: 3rem;

color: var(--success-color);

margin-bottom: 20px;

}

.download-modal h3 {

color: var(--primary-color);

margin-bottom: 15px;

font-size: 1.5rem;

}

.download-modal p {

color: var(--text-light);

margin-bottom: 25px;

line-height: 1.6;

}

.download-modal-buttons {

display: flex;

gap: 15px;

justify-content: center;

}

/\* Responsividade \*/

@media (max-width: 768px) {

.appointment-header {

flex-direction: column;

gap: 15px;

align-items: flex-start;

padding: 20px;

}

.appointment-body {

grid-template-columns: 1fr;

gap: 20px;

padding: 20px;

}

.appointment-footer {

flex-direction: column;

padding: 20px;

}

.appointment-footer .btn {

width: 100%;

justify-content: center;

}

.download-modal-buttons {

flex-direction: column;

}

}

/\* Estados de loading para os botões \*/

.btn.loading {

position: relative;

color: transparent;

}

.btn.loading::after {

content: '';

position: absolute;

width: 20px;

height: 20px;

top: 50%;

left: 50%;

margin-left: -10px;

margin-top: -10px;

border: 2px solid #ffffff;

border-radius: 50%;

border-right-color: transparent;

animation: button-loading 0.8s linear infinite;

}

@keyframes button-loading {

from {

transform: rotate(0);

}

to {

transform: rotate(360deg);

}

}

/\* Estilos para os diferentes estados do lembrete \*/

.reminder-card.next-appointment {

background: linear-gradient(135deg, #e7f3ff 0%, #d1e7ff 100%);

border-left: 4px solid #0066cc;

}

.reminder-card.reminder-urgent {

background: linear-gradient(135deg, #ffe7e7 0%, #ffd1d1 100%);

border-left: 4px solid #dc3545;

}

.reminder-icon.urgent {

background: #dc3545 !important;

color: white !important;

}

.btn-warning {

background: #ffc107 !important;

color: #212529 !important;

border-color: #ffc107 !important;

}

.btn-warning:hover {

background: #e0a800 !important;

border-color: #d39e00 !important;

}

/\* Melhorar a visualização dos cards \*/

.appointment-card {

transition: all 0.3s ease;

}

.appointment-card:hover {

transform: translateY(-2px);

box-shadow: 0 4px 12px rgba(0,0,0,0.15);

}

/\* Animações \*/

@keyframes pulse {

0% { box-shadow: 0 0 0 0 rgba(0, 102, 204, 0.4); }

70% { box-shadow: 0 0 0 10px rgba(0, 102, 204, 0); }

100% { box-shadow: 0 0 0 0 rgba(0, 102, 204, 0); }

}

@keyframes highlight {

0% { background-color: rgba(255, 193, 7, 0.3); }

100% { background-color: transparent; }

}

/\* Estilos para o modal de download \*/

.download-modal {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

display: flex;

justify-content: center;

align-items: center;

z-index: 10000;

opacity: 0;

visibility: hidden;

transition: all 0.3s ease;

}

.download-modal.show {

opacity: 1;

visibility: visible;

}

.download-modal-content {

background: white;

padding: 30px;

border-radius: 10px;

text-align: center;

max-width: 400px;

width: 90%;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.3);

transform: translateY(-20px);

transition: transform 0.3s ease;

}

.download-modal.show .download-modal-content {

transform: translateY(0);

}

.download-icon {

font-size: 48px;

color: #28a745;

margin-bottom: 20px;

}

.download-modal h3 {

color: #213f57;

margin-bottom: 15px;

font-size: 1.5rem;

}

.download-modal p {

color: #666;

margin-bottom: 20px;

line-height: 1.5;

}

.download-modal-buttons {

display: flex;

justify-content: center;

gap: 10px;

}

/\* Estilo para botão de loading \*/

.btn.loading {

position: relative;

color: transparent !important;

}

.btn.loading::after {

content: '';

position: absolute;

width: 16px;

height: 16px;

top: 50%;

left: 50%;

margin-left: -8px;

margin-top: -8px;

border: 2px solid #ffffff;

border-radius: 50%;

border-right-color: transparent;

animation: spin 1s linear infinite;

}

@keyframes spin {

from { transform: rotate(0deg); }

to { transform: rotate(360deg); }

}

/\* Melhorar aparência dos botões na agenda \*/

.appointment-footer .btn {

position: relative;

transition: all 0.3s ease;

}

.appointment-footer .btn:disabled {

opacity: 0.6;

cursor: not-allowed;

}

/\* Feedback visual para download \*/

@keyframes downloadPulse {

0% { transform: scale(1); }

50% { transform: scale(0.95); }

100% { transform: scale(1); }

}

.btn-download-complete {

animation: downloadPulse 0.5s ease;

background-color: #28a745 !important;

}

/\* ===== CORREÇÃO DO STATUS DO USUÁRIO NO MODO RESPONSIVO ===== \*/

/\* Garantir que o status do usuário seja visível em mobile \*/

@media (max-width: 768px) {

.user-icon .login-status {

display: inline-block !important;

font-size: 0.85rem;

margin-left: 8px;

color: white;

white-space: nowrap;

}

.user-icon {

padding: 8px 12px;

border-radius: 20px;

background: rgba(255, 255, 255, 0.1);

display: flex !important;

align-items: center;

gap: 8px;

}

}

/\* Para telas muito pequenas \*/

@media (max-width: 480px) {

.user-icon .login-status {

font-size: 0.8rem;

margin-left: 6px;

}

.user-icon {

padding: 6px 10px;

gap: 6px;

}

}

/\* Garantir que o dropdown funcione corretamente no mobile \*/

@media (max-width: 1024px) {

.user-dropdown {

position: static;

}

.dropdown-menu {

position: fixed;

top: 85px;

left: 0;

width: 100%;

max-width: none;

border-radius: 0;

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);

border-top: 1px solid #eee;

padding: 15px 0;

}

}

/\* ===== GARANTIR VISIBILIDADE DO STATUS EM TODAS AS TELAS ===== \*/

@media (max-width: 1024px) {

.user-dropdown .user-icon .login-status {

display: inline-block !important;

visibility: visible !important;

opacity: 1 !important;

color: white !important;

font-size: 0.85rem;

}

.user-dropdown .user-icon {

display: flex !important;

align-items: center !important;

gap: 8px !important;

}

}

/\* Correção específica para o menu mobile \*/

@media (max-width: 768px) {

.nav.active ~ .action-icons .user-dropdown .user-icon .login-status {

display: inline-block !important;

color: white !important;

}

/\* Garantir que o elemento seja sempre visível no header mobile \*/

.header .user-dropdown .user-icon .login-status {

display: inline-block !important;

color: white !important;

font-weight: 500;

}

}

# Arquivo: public/html/admindex.html

<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

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

<title>OilSmart - Sistema Administrativo</title>

<link rel="stylesheet" href="/css/adm.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

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

</head>

<body>

<!-- Container de Autenticação -->

<div id="authContainer" class="auth-container">

<!-- Card de Login -->

<div id="loginCard" class="auth-card">

<div class="auth-logo">

<i class="fas fa-oil-can"></i>

</div>

<h2 class="auth-title">Login OilSmart</h2>

<form id="loginForm" class="auth-form">

<div class="form-group-auth">

<label for="loginEmail">E-mail</label>

<input type="email" id="loginEmail" placeholder="seu@email.com" required>

</div>

<div class="form-group-auth">

<label for="loginSenha">Senha</label>

<input type="password" id="loginSenha" placeholder="Sua senha" required>

</div>

<button type="submit" class="auth-btn">Entrar no Sistema</button>

<div id="loginError" class="auth-error">

E-mail ou senha incorretos. Tente novamente.

</div>

</form>

<div class="auth-links">

<a href="#" class="auth-link" onclick="mostrarCadastro()">Criar nova conta</a>

<a href="#" class="auth-link">Esqueci minha senha</a>

</div>

</div>

<!-- Card de Cadastro -->

<div id="cadastroCard" class="auth-card" style="display: none;">

<div class="auth-logo">

<i class="fas fa-oil-can"></i>

</div>

<h2 class="auth-title">Criar Conta</h2>

<form id="cadastroForm" class="auth-form">

<div class="form-group-auth">

<label for="cadastroNome">Nome Completo</label>

<input type="text" id="cadastroNome" placeholder="Seu nome completo" required>

</div>

<div class="form-group-auth">

<label for="cadastroEmail">E-mail</label>

<input type="email" id="cadastroEmail" placeholder="seu@email.com" required>

</div>

<div class="form-group-auth">

<label for="cadastroSenha">Senha</label>

<input type="password" id="cadastroSenha" placeholder="Mínimo 6 caracteres" required>

</div>

<div class="form-group-auth">

<label for="cadastroConfirmarSenha">Confirmar Senha</label>

<input type="password" id="cadastroConfirmarSenha" placeholder="Digite novamente a senha" required>

</div>

<button type="submit" class="auth-btn">Criar Conta</button>

<div id="cadastroError" class="auth-error" style="display: none;">

Erro no cadastro. Verifique os dados.

</div>

</form>

<div class="auth-links">

<a href="#" class="auth-link" onclick="voltarParaLogin()">Voltar para o login</a>

</div>

</div>

</div>

<!-- Container Administrativo -->

<div id="adminContainer" style="display: none;">

<!-- Sidebar -->

<aside id="sidebar" class="sidebar">

<div class="sidebar-header">

<div class="sidebar-logo">

<i class="fas fa-oil-can"></i>

</div>

<h3>OilSmart</h3>

</div>

<nav class="sidebar-nav">

<ul>

<li class="nav-item">

<a href="#" class="nav-link active" onclick="navegarPara('dashboard')">

<i class="fas fa-home"></i>

<span>Dashboard</span>

</a>

</li>

<li class="nav-item">

<a href="#" class="nav-link" onclick="navegarPara('produtos')">

<i class="fas fa-box"></i>

<span>Produtos</span>

</a>

</li>

<li class="nav-item">

<a href="#" class="nav-link" onclick="navegarPara('vendas')">

<i class="fas fa-shopping-cart"></i>

<span>Vendas</span>

</a>

</li>

<li class="nav-item">

<a href="#" class="nav-link" onclick="navegarPara('funcionarios')">

<i class="fas fa-users"></i>

<span>Funcionários</span>

</a>

</li>

<li class="nav-item">

<a href="#" class="nav-link" onclick="navegarPara('relatorios')">

<i class="fas fa-chart-bar"></i>

<span>Relatórios</span>

</a>

</li>

</ul>

</nav>

<div class="sidebar-footer">

<div class="user-info">

<div class="user-avatar">

<i class="fas fa-user"></i>

</div>

<div class="user-details">

<span class="user-name">Administrador</span>

<span class="user-role">Super Admin</span>

</div>

</div>

<button class="logout-btn" onclick="fazerLogout()">

<i class="fas fa-sign-out-alt"></i>

</button>

</div>

</aside>

<!-- Overlay para sidebar em mobile -->

<div id="sidebarOverlay" class="sidebar-overlay"></div>

<!-- Conteúdo Principal -->

<main class="main-content">

<header class="main-header">

<div class="header-left">

<button class="sidebar-toggle" onclick="toggleSidebar()">

<i class="fas fa-bars"></i>

</button>

<h1 id="pageTitle" class="page-title">Dashboard</h1>

</div>

<div class="header-right">

<div class="search-box">

<i class="fas fa-search"></i>

<input type="text" placeholder="Pesquisar...">

</div>

<button class="notification-btn">

<i class="fas fa-bell"></i>

<span class="notification-badge">3</span>

</button>

<div class="date-time">

<span id="currentDate">Segunda-feira, 1 de Janeiro de 2023</span>

<span id="currentTime">00:00:00</span>

</div>

</div>

</header>

<div class="content-area">

<!-- Dashboard -->

<section id="dashboard" class="content-section active">

<div class="metrics-grid">

<div class="metric-card">

<div class="metric-icon">

<i class="fas fa-shopping-cart"></i>

</div>

<div class="metric-content">

<h3>TOTAL DE VENDAS</h3>

<div id="totalVendas" class="metric-value">R$ 0,00</div>

<div class="metric-change positive">

<i class="fas fa-arrow-up"></i> 12% desde o mês passado

</div>

</div>

</div>

<div class="metric-card">

<div class="metric-icon">

<i class="fas fa-box"></i>

</div>

<div class="metric-content">

<h3>PRODUTOS</h3>

<div id="totalProdutos" class="metric-value">0</div>

<div class="metric-change neutral">

<i class="fas fa-minus"></i> Sem alteração

</div>

</div>

</div>

<div class="metric-card">

<div class="metric-icon">

<i class="fas fa-users"></i>

</div>

<div class="metric-content">

<h3>FUNCIONÁRIOS</h3>

<div id="totalFuncionarios" class="metric-value">0</div>

<div class="metric-change positive">

<i class="fas fa-arrow-up"></i> 5% desde o mês passado

</div>

</div>

</div>

<div class="metric-card">

<div class="metric-icon">

<i class="fas fa-clock"></i>

</div>

<div class="metric-content">

<h3>VENDAS PENDENTES</h3>

<div id="vendasPendentes" class="metric-value">0</div>

<div class="metric-change negative">

<i class="fas fa-arrow-down"></i> 3% desde o mês passado

</div>

</div>

</div>

</div>

<div class="charts-grid">

<div class="chart-card">

<div class="chart-header">

<h3>Vendas Mensais</h3>

<select>

<option>2023</option>

<option>2022</option>

</select>

</div>

<div class="chart-container">

<canvas id="vendasChart"></canvas>

</div>

</div>

<div class="chart-card">

<div class="chart-header">

<h3>Vendas por Categoria</h3>

<select>

<option>2023</option>

<option>2022</option>

</select>

</div>

<div class="chart-container">

<canvas id="categoriasChart"></canvas>

</div>

</div>

</div>

<div class="activities-section">

<div class="activities-card">

<div class="card-header">

<h3>Atividades Recentes</h3>

<a href="#" class="view-all">Ver tudo</a>

</div>

<div id="atividadesList" class="activities-list">

<!-- Atividades serão carregadas via JavaScript -->

</div>

</div>

<div class="activities-card">

<div class="card-header">

<h3>Status de Estoque</h3>

<a href="#" class="view-all">Ver detalhes</a>

</div>

<div class="activities-list">

<div class="activity-item">

<div class="activity-icon" style="background-color: rgba(220, 53, 69, 0.1); color: #dc3545;">

<i class="fas fa-exclamation-triangle"></i>

</div>

<div class="activity-content">

<div class="activity-title">Produtos com estoque crítico</div>

<div class="activity-time">0 produtos com menos de 5 unidades</div>

</div>

</div>

<div class="activity-item">

<div class="activity-icon" style="background-color: rgba(255, 193, 7, 0.1); color: #ffc107;">

<i class="fas fa-info-circle"></i>

</div>

<div class="activity-content">

<div class="activity-title">Produtos com estoque baixo</div>

<div class="activity-time">0 produtos com menos de 15 unidades</div>

</div>

</div>

<div class="activity-item">

<div class="activity-icon" style="background-color: rgba(40, 167, 69, 0.1); color: #28a745;">

<i class="fas fa-check-circle"></i>

</div>

<div class="activity-content">

<div class="activity-title">Estoque suficiente</div>

<div class="activity-time">0 produtos com estoque adequado</div>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- Produtos -->

<section id="produtos" class="content-section">

<div class="section-header">

<h2>Gerenciamento de Produtos</h2>

<button class="btn btn-primary" onclick="abrirModalProduto()">

<i class="fas fa-plus"></i> Novo Produto

</button>

</div>

<div id="produtosGrid" class="produtos-grid">

<!-- Produtos serão carregados via JavaScript -->

</div>

</section>

<!-- Vendas -->

<section id="vendas" class="content-section">

<div class="section-header">

<h2>Histórico de Vendas</h2>

<button class="btn btn-primary" onclick="abrirModalVenda()">

<i class="fas fa-plus"></i> Nova Venda

</button>

</div>

<div class="vendas-summary">

<div class="summary-card">

<h4>Total de Vendas</h4>

<div id="resumoTotalVendas" class="summary-value">R$ 0,00</div>

</div>

<div class="summary-card">

<h4>Vendas Confirmadas</h4>

<div id="resumoVendasConfirmadas" class="summary-value">0</div>

</div>

<div class="summary-card">

<h4>Vendas Pendentes</h4>

<div id="resumoVendasPendentes" class="summary-value">0</div>

</div>

<div class="summary-card">

<h4>Média por Venda</h4>

<div id="resumoMediaVendas" class="summary-value">R$ 0,00</div>

</div>

</div>

<div class="filters-bar">

<div class="filter-group">

<label for="filtroData">Data</label>

<input type="date" id="filtroData">

</div>

<div class="filter-group">

<label for="filtroStatus">Status</label>

<select id="filtroStatus">

<option value="">Todos</option>

<option value="pendente">Pendente</option>

<option value="confirmado">Confirmado</option>

<option value="cancelado">Cancelado</option>

</select>

</div>

<div class="filter-group">

<label for="filtroCliente">Cliente</label>

<input type="text" id="filtroCliente" placeholder="Nome do cliente">

</div>

<div class="filter-group">

<label>&nbsp;</label>

<button class="btn btn-primary">

<i class="fas fa-filter"></i> Filtrar

</button>

</div>

</div>

<div class="table-container">

<table class="data-table">

<thead>

<tr>

<th>ID</th>

<th>Data</th>

<th>Cliente</th>

<th>Itens</th>

<th>Total</th>

<th>Status</th>

<th>Ações</th>

</tr>

</thead>

<tbody id="vendasTable">

<!-- Vendas serão carregadas via JavaScript -->

</tbody>

</table>

</div>

</section>

<!-- Funcionários -->

<section id="funcionarios" class="content-section">

<div class="section-header">

<h2>Gerenciamento de Funcionários</h2>

<button class="btn btn-primary" onclick="abrirModalFuncionario()">

<i class="fas fa-plus"></i> Novo Funcionário

</button>

</div>

<div id="funcionariosGrid" class="funcionarios-grid">

<!-- Funcionários serão carregados via JavaScript -->

</div>

</section>

<!-- Relatórios -->

<section id="relatorios" class="content-section">

<div class="section-header">

<h2>Relatórios</h2>

</div>

<div class="relatorios-grid">

<div class="relatorio-card">

<i class="fas fa-shopping-cart"></i>

<h3>Relatório de Vendas</h3>

<p>Gere relatórios detalhados de vendas por período, incluindo totais, produtos mais vendidos e análise de desempenho.</p>

<button class="btn btn-primary" onclick="gerarRelatorioVendas()">

<i class="fas fa-download"></i> Gerar Relatório

</button>

</div>

<div class="relatorio-card">

<i class="fas fa-box"></i>

<h3>Relatório de Estoque</h3>

<p>Relatório completo do estoque atual, incluindo produtos com baixa quantidade e necessidade de reposição.</p>

<button class="btn btn-primary" onclick="gerarRelatorioEstoque()">

<i class="fas fa-download"></i> Gerar Relatório

</button>

</div>

<div class="relatorio-card">

<i class="fas fa-users"></i>

<h3>Relatório de Funcionários</h3>

<p>Relatório com informações dos funcionários, cargos, departamentos e status no sistema.</p>

<button class="btn btn-primary" onclick="gerarRelatorioFuncionarios()">

<i class="fas fa-download"></i> Gerar Relatório

</button>

</div>

</div>

</section>

</div>

</main>

</div>

<!-- Modal Produto -->

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

<div class="modal-content">

<div class="modal-header">

<h3 id="modalProdutoTitulo">Adicionar Produto</h3>

<button class="close-modal" onclick="fecharModal('modalProduto')">&times;</button>

</div>

<form id="produtoForm" class="modal-body">

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

<div class="form-row">

<div class="form-group">

<label for="produtoNome">Nome do Produto</label>

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

</div>

<div class="form-group">

<label for="produtoCategoria">Categoria</label>

<select id="produtoCategoria" required>

<option value="">Selecione uma categoria</option>

<option value="Óleos">Óleos</option>

<option value="Filtros">Filtros</option>

<option value="Aditivos">Aditivos</option>

<option value="Acessórios">Acessórios</option>

</select>

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="produtoMarca">Marca</label>

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

</div>

<div class="form-group">

<label for="produtoPreco">Preço (R$)</label>

<input type="number" id="produtoPreco" step="0.01" min="0" required>

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="produtoEstoque">Quantidade em Estoque</label>

<input type="number" id="produtoEstoque" min="0" required>

</div>

<div class="form-group">

<label for="produtoImagem">Imagem do Produto</label>

<input type="file" id="produtoImagem" accept="image/\*">

</div>

</div>

<div class="form-group">

<label for="produtoDescricao">Descrição</label>

<textarea id="produtoDescricao" rows="3"></textarea>

</div>

<div class="form-group">

<label>Prévia da Imagem</label>

<div class="image-upload">

<div id="produtoImagemPreview" class="image-preview">

<div class="placeholder">Imagem do produto</div>

</div>

</div>

</div>

</form>

<div class="modal-actions">

<button type="button" class="btn btn-secondary" onclick="fecharModal('modalProduto')">Cancelar</button>

<button type="submit" form="produtoForm" class="btn btn-primary">Salvar Produto</button>

</div>

</div>

</div>

<!-- Modal Funcionário -->

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

<div class="modal-content">

<div class="modal-header">

<h3 id="modalFuncionarioTitulo">Adicionar Funcionário</h3>

<button class="close-modal" onclick="fecharModal('modalFuncionario')">&times;</button>

</div>

<form id="funcionarioForm" class="modal-body">

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

<div class="form-row">

<div class="form-group">

<label for="funcionarioNome">Nome Completo</label>

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

</div>

<div class="form-group">

<label for="funcionarioEmail">E-mail</label>

<input type="email" id="funcionarioEmail" required>

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="funcionarioTelefone">Telefone</label>

<input type="tel" id="funcionarioTelefone" required>

</div>

<div class="form-group">

<label for="funcionarioCargo">Cargo</label>

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

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="funcionarioDepartamento">Departamento</label>

<select id="funcionarioDepartamento" required>

<option value="">Selecione um departamento</option>

<option value="Vendas">Vendas</option>

<option value="Administração">Administração</option>

<option value="Logística">Logística</option>

<option value="TI">TI</option>

</select>

</div>

<div class="form-group">

<label for="funcionarioStatus">Status</label>

<select id="funcionarioStatus" required>

<option value="ativo">Ativo</option>

<option value="inativo">Inativo</option>

</select>

</div>

</div>

</form>

<div class="modal-actions">

<button type="button" class="btn btn-secondary" onclick="fecharModal('modalFuncionario')">Cancelar</button>

<button type="submit" form="funcionarioForm" class="btn btn-primary">Salvar Funcionário</button>

</div>

</div>

</div>

<!-- Modal Detalhes da Venda -->

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

<div class="modal-content">

<div class="modal-header">

<h3>Detalhes da Venda</h3>

<button class="close-modal" onclick="fecharModal('modalVendaDetalhes')">&times;</button>

</div>

<div class="modal-body">

<!-- Conteúdo será preenchido via JavaScript -->

</div>

<div class="modal-actions">

<button type="button" class="btn btn-secondary" onclick="fecharModal('modalVendaDetalhes')">Fechar</button>

<button type="button" class="btn btn-primary">Imprimir</button>

</div>

</div>

</div>

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

</body>

</html>

# Arquivo: public/html/agenda.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>Agendamentos - OilSmart</title>

<meta name="description" content="Agende seus serviços automotivos com a OilSmart. Histórico de agendamentos e lembretes para próxima troca.">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<link rel="stylesheet" href="/css/main.css">

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon">

<link rel="stylesheet" href="/css/auth.css">

<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>

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

</head>

<body>

<header class="header">

<div class="container">

<div class="navbar">

<button class="hamburger" id="hamburger">

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

</button>

<div class="logo">

<a href="index.html"><img src="/img/oilamarelo.png" alt="OilSmart"></a>

</div>

<nav class="nav" id="nav">

<ul class="nav\_\_list">

<li><a href="/html/index.html">Início</a></li>

<li><a href="/html/servicos.html">Serviços</a></li>

<li><a href="/html/agenda.html" class="active">Agenda</a></li>

<li><a href="/html/sobre.html">Sobre</a></li>

<li><a href="/html/contato.html">Contato</a></li>

</ul>

</nav>

<div class="action-icons">

<div class="user-dropdown" id="user-dropdown">

<div class="user-icon" id="login-btn">

<i class="fas fa-user"></i>

<span class="login-status" id="login-status">Login</span>

</div>

<div class="dropdown-menu" id="dropdown-menu">

<div class="user-info">

<strong id="user-display-name">Usuário</strong>

<span id="user-email">email@exemplo.com</span>

</div>

<div class="dropdown-divider"></div>

<a href="/html/agenda.html" class="dropdown-item">

<i class="fas fa-calendar-alt"></i> Meus Agendamentos

</a>

<div class="dropdown-divider"></div>

<a href="/html/index.html" id="logout-btn" class="dropdown-item">

<i class="fas fa-sign-out-alt"></i> Sair

</a>

</div>

</div>

</div>

</div>

</div>

</header>

<main class="main-content">

<div class="container">

<h1 class="page-title">Meus Agendamentos</h1>

<div class="agenda-container">

<!-- Lembrete de Próxima Troca -->

<section class="reminder-section">

<div class="section-header">

<h2><i class="fas fa-bell"></i> Lembrete de Próxima Troca</h2>

</div>

<div class="reminder-card">

<div class="reminder-content">

<div class="reminder-icon">

<i class="fas fa-oil-can"></i>

</div>

<div class="reminder-details">

<h3>Troca de Óleo e Filtro</h3>

<p>Recomendamos realizar a próxima troca até: <strong>15/11/2023</strong></p>

<p>Quilometragem estimada: <strong>85.000 km</strong></p>

</div>

</div>

<a href="/html/servicos.html" class="btn btn-primary">Agendar Troca</a>

</div>

</section>

<section class="history-section">

<div class="section-header">

<h2><i class="fas fa-history"></i> Histórico de Agendamentos</h2>

<div class="filter-controls">

<div class="filter-group">

<label for="filter-status"><i class="fas fa-filter"></i> Status:</label>

<select id="filter-status" class="filter-select">

<option value="all">Todos os Status</option>

<option value="pendente">Pendentes</option>

<option value="agendado">Agendados</option>

<option value="concluido">Concluídos</option>

<option value="cancelado">Cancelados</option>

<option value="fora\_prazo">Fora do Prazo</option>

</select>

</div>

<div class="filter-group">

<label for="filter-period"><i class="fas fa-calendar"></i> Período:</label>

<select id="filter-period" class="filter-select">

<option value="all">Todos os Períodos</option>

<option value="today">Hoje</option>

<option value="week">Esta Semana</option>

<option value="month">Este Mês</option>

<option value="last\_month">Mês Passado</option>

<option value="year">Este Ano</option>

</select>

</div>

<div class="filter-group">

<label for="filter-sort"><i class="fas fa-sort"></i> Ordenar:</label>

<select id="filter-sort" class="filter-select">

<option value="newest">Mais Recentes</option>

<option value="oldest">Mais Antigos</option>

<option value="date\_asc">Data (Crescente)</option>

<option value="date\_desc">Data (Decrescente)</option>

</select>

</div>

<button id="clear-filters" class="btn btn-outline btn-small">

<i class="fas fa-times"></i> Limpar Filtros

</button>

</div>

</div>

<div class="appointments-list">

<!-- Agendamento 1 -->

<div class="appointment-card">

<div class="appointment-header">

<span class="protocol">Protocolo: #OIL20231015-001</span>

<span class="status completed">Concluído</span>

</div>

<div class="appointment-body">

<div class="service-info">

<h3>Troca de Óleo Sintético</h3>

<p><i class="fas fa-car"></i> Volkswagen Gol 1.6 2020</p>

<p><i class="fas fa-oil-can"></i> Óleo Mobil 1 5W30</p>

</div>

<div class="date-info">

<p><i class="far fa-calendar-alt"></i> 15/10/2023</p>

<p><i class="far fa-clock"></i> 14:00 - 15:30</p>

</div>

<div class="location-info">

<p><i class="fas fa-map-marker-alt"></i> OilSmart - Unidade Paulista</p>

<p>Av. Paulista, 1000 - São Paulo/SP</p>

</div>

<div class="price-info">

<p><i class="fas fa-tag"></i> Total: R$ 189,90</p>

</div>

</div>

<div class="appointment-footer">

<button class="btn btn-outline" onclick="repeatAppointment(1)"><i class="fas fa-redo"></i> Repetir Agendamento</button>

<button class="btn btn-outline" onclick="downloadInvoice(1)"><i class="fas fa-file-invoice"></i> Comprovante</button>

<button class="btn btn-outline" onclick="rateService(1)"><i class="fas fa-star"></i> Avaliar Serviço</button>

</div>

</div>

<!-- Agendamento 2 -->

<div class="appointment-card">

<div class="appointment-header">

<span class="protocol">Protocolo: #OIL20230822-002</span>

<span class="status completed">Concluído</span>

</div>

<div class="appointment-body">

<div class="service-info">

<h3>Revisão Completa</h3>

<p><i class="fas fa-car"></i> Volkswagen Gol 1.6 2020</p>

<p><i class="fas fa-tools"></i> Troca de óleo, filtros e verificação geral</p>

</div>

<div class="date-info">

<p><i class="far fa-calendar-alt"></i> 22/08/2023</p>

<p><i class="far fa-clock"></i> 10:00 - 12:30</p>

</div>

<div class="location-info">

<p><i class="fas fa-map-marker-alt"></i> OilSmart - Unidade Morumbi</p>

<p>Av. Morumbi, 2500 - São Paulo/SP</p>

</div>

<div class="price-info">

<p><i class="fas fa-tag"></i> Total: R$ 320,50</p>

</div>

</div>

<div class="appointment-footer">

<button class="btn btn-outline" onclick="repeatAppointment(2)"><i class="fas fa-redo"></i> Repetir Agendamento</button>

<button class="btn btn-outline" onclick="downloadInvoice(2)"><i class="fas fa-file-invoice"></i> Comprovante</button>

<button class="btn btn-outline" onclick="rateService(2)"><i class="fas fa-star"></i> Avaliar Serviço</button>

</div>

</div>

<!-- Agendamento 3 - Cancelado -->

<div class="appointment-card cancelled">

<div class="appointment-header">

<span class="protocol">Protocolo: #OIL20230910-003</span>

<span class="status cancelled">Cancelado</span>

</div>

<div class="appointment-body">

<div class="service-info">

<h3>Troca de Óleo Semissintético</h3>

<p><i class="fas fa-car"></i> Fiat Uno 1.0 2018</p>

<p><i class="fas fa-oil-can"></i> Óleo Semissintético 10W-40</p>

</div>

<div class="date-info">

<p><i class="far fa-calendar-alt"></i> 10/09/2023</p>

<p><i class="far fa-clock"></i> 09:00 - 10:30</p>

</div>

<div class="location-info">

<p><i class="fas fa-map-marker-alt"></i> OilSmart - Unidade Centro</p>

<p>Rua da Consolação, 500 - São Paulo/SP</p>

</div>

<div class="cancel-info">

<p><i class="fas fa-info-circle"></i> Motivo: Mudança de planos</p>

</div>

</div>

<div class="appointment-footer">

<button class="btn btn-outline" onclick="repeatAppointment(3)"><i class="fas fa-redo"></i> Novo Agendamento</button>

<button class="btn btn-outline" onclick="downloadInvoice(3)"><i class="fas fa-file-invoice"></i> Comprovante</button>

<span class="text-muted">Ações indisponíveis</span>

</div>

</div>

</div>

</section>

</div>

</div>

</main>

<footer class="footer">

<div class="container">

<div class="footer-content">

<div class="footer-brand">

<img src="/img/fjkaemfkemif.png" alt="OilSmart" class="footer-logo">

<p>Soluções completas para seu veículo</p>

</div>

<div class="footer-links">

<h3>Links Úteis</h3>

<ul>

<li><a href="/html/politicas.html">Política de Privacidade</a></li>

<li><a href="/html/termos.html">Termos de Serviço</a></li>

<li><a href="/html/faq.html">Perguntas Frequentes</a></li>

</ul>

</div>

<div class="footer-contact">

<h3>Contato</h3>

<p><i class="fas fa-phone"></i> (11) 4002-8922</p>

<p><i class="fas fa-envelope"></i> contato@OilSmart.com.br</p>

</div>

</div>

<div class="copyright">

<p>© 2025 OilSmart. Todos os direitos reservados.</p>

</div>

</div>

</footer>

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

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

</body>

</html>

# Arquivo: public/html/cadastro.html

<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

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

<title>Cadastro - OilSmart</title>

<link rel="stylesheet" href="/css/login.css">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon">

</head>

<body>

<div class="login-card">

<div class="logo-container">

<div class="logo">OS</div>

<h1 class="login-title">Junte-se à OilSmart</h1>

</div>

<p class="login-subtitle">Crie sua conta e comece a gerenciar seus veículos com inteligência</p>

<div class="divider"></div>

<form id="registerForm">

<div class="form-group">

<label for="register-fullname" class="section-title">Nome completo</label>

<input type="text" id="register-fullname" class="input-field" placeholder="Digite seu nome completo" required>

</div>

<div class="form-group">

<label for="register-email" class="section-title">E-mail</label>

<input type="email" id="register-email" class="input-field" placeholder="seu.email@exemplo.com" required>

</div>

<div class="form-group">

<label for="register-password" class="section-title">Senha</label>

<input type="password" id="register-password" class="input-field" placeholder="Crie uma senha segura" required>

</div>

<div class="form-group">

<label for="register-confirm-password" class="section-title">Confirmar Senha</label>

<input type="password" id="register-confirm-password" class="input-field" placeholder="Digite a mesma senha" required>

</div>

<div class="password-options">

<label class="checkbox-container">

<input type="checkbox" id="register-terms" required>

<span class="checkmark"></span>

Concordo com os <a href="/html/termos.html">Termos</a> e <a href="/html/politicas.html">Política</a>

</label>

</div>

<div class="divider"></div>

<div class="button-group">

<button type="button" class="btn btn-secondary" onclick="window.location.href='login.html'">Voltar</button>

<button type="submit" class="btn btn-primary">Criar minha conta</button>

</div>

<p class="register-text">Já possui uma conta? <a href="login.html" class="register-link">Faça login aqui</a></p>

</form>

</div>

<script>

document.getElementById('registerForm').addEventListener('submit', async function (e) {

e.preventDefault();

// Obter valores dos campos

const nome = document.getElementById('register-fullname').value.trim();

const email = document.getElementById('register-email').value.trim();

const senha = document.getElementById('register-password').value;

const confirmarSenha = document.getElementById('register-confirm-password').value;

const termosAceitos = document.getElementById('register-terms').checked;

// Validações básicas

if (!nome) {

alert('Por favor, informe seu nome completo');

return;

}

if (nome.split(' ').length < 2) {

alert('Por favor, informe seu nome e sobrenome');

return;

}

if (!email) {

alert('Por favor, informe seu e-mail');

return;

}

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailRegex.test(email)) {

alert('Por favor, informe um e-mail válido');

return;

}

if (!senha) {

alert('Por favor, crie uma senha');

return;

}

if (senha.length < 6) {

alert('A senha deve ter pelo menos 6 caracteres');

return;

}

if (senha !== confirmarSenha) {

alert('As senhas não coincidem');

return;

}

if (!termosAceitos) {

alert('Você precisa aceitar os termos e condições');

return;

}

try {

console.log('Enviando dados:', { nome, email, senha: '\*\*\*' });

// Enviar dados para o servidor

const response = await fetch('http://localhost:3000/api/auth/register', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

nome: nome,

email: email,

senha: senha,

telefone: '',

cpf: '',

endereco: '',

cep: '',

cidade: '',

estado: ''

})

});

console.log('Resposta do servidor:', response.status);

// Verificar se a resposta é JSON

const contentType = response.headers.get('content-type');

let result;

if (contentType && contentType.includes('application/json')) {

result = await response.json();

} else {

const text = await response.text();

throw new Error(`Resposta não é JSON: ${text}`);

}

console.log('Resultado:', result);

if (result.success) {

alert(`Conta criada com sucesso! Bem-vindo(a), ${nome}`);

window.location.href = 'login.html';

} else {

alert(result.message);

}

} catch (error) {

console.error('Erro completo:', error);

alert(`Erro: ${error.message}\n\nVerifique o console para mais detalhes.`);

}

});

// Testar conexão ao carregar a página

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

try {

const response = await fetch('http://localhost:3000/api/test');

if (response.ok) {

const data = await response.json();

console.log('✅ Servidor conectado:', data);

}

} catch (error) {

console.warn('❌ Não foi possível conectar ao servidor:', error.message);

alert('Servidor não está respondendo. Execute: node server.js');

}

});

</script>

</body>

</html>

# Arquivo: public/html/contato.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>OilSmart - Contatos</title>

<meta name="description" content="Entre em contato com a OilSmart. Estamos à disposição para tirar suas dúvidas sobre serviços automotivos.">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<link rel="stylesheet" href="/css/main.css">

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon"></head>

<link rel="stylesheet" href="/css/auth.css">

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

</head>

<body>

<header class="header">

<div class="container">

<div class="navbar">

<button class="hamburger" id="hamburger">

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

</button>

<div class="logo">

<a href="index.html"><img src="/img/oilamarelo.png" alt="OilSmart"></a>

</div>

<nav class="nav" id="nav">

<ul class="nav\_\_list">

<li><a href="/html/index.html">Início</a></li>

<li><a href="/html/servicos.html">Serviços</a></li>

<li><a href="/html/agenda.html">Agenda</a></li>

<li><a href="/html/sobre.html">Sobre</a></li>

<li><a href="/html/Contato.html" class="active">Contato</a></li>

</ul>

</nav>

<div class="action-icons">

<div class="user-dropdown" id="user-dropdown">

<div class="user-icon" id="login-btn">

<i class="fas fa-user"></i>

<span class="login-status" id="login-status">Login</span>

</div>

<div class="dropdown-menu" id="dropdown-menu">

<div class="user-info">

<strong id="user-display-name">Usuário</strong>

<span id="user-email">email@exemplo.com</span>

</div>

<div class="dropdown-divider"></div>

<a href="/html/agenda.html" class="dropdown-item">

<i class="fas fa-calendar-alt"></i> Meus Agendamentos

</a>

<div class="dropdown-divider"></div>

<a href="/html/index.html" id="logout-btn" class="dropdown-item">

<i class="fas fa-sign-out-alt"></i> Sair

</a>

</div>

</div>

</div>

</div>

</div>

</header>

<main class="main-content">

<section class="about-banner">

<div class="container">

<h1>Entre em Contato</h1>

<p>Estamos aqui para ajudar com todas as suas necessidades automotivas</p>

</div>

</section>

<div class="hero-divider"></div>

<div class="container">

<!-- Informações de Contato -->

<section class="contact-info-section">

<h2 class="section-title">Nossos Canais de Atendimento</h2>

<div class="contact-grid">

<div class="contact-card">

<div class="contact-icon">

<i class="fas fa-phone"></i>

</div>

<h3>Telefone</h3>

<p>(11) 4002-8922</p>

<span class="contact-detail">Segunda a Sexta, 8h às 18h</span>

</div>

<div class="contact-card">

<div class="contact-icon">

<i class="fas fa-envelope"></i>

</div>

<h3>E-mail</h3>

<p>oilsmartoficiall@gmail.com</p>

<span class="contact-detail">Respondemos em até 24h</span>

</div>

<div class="contact-card">

<div class="contact-icon">

<i class="fab fa-whatsapp"></i>

</div>

<h3>WhatsApp</h3>

<p>(11) 99999-9999</p>

<span class="contact-detail">Atendimento rápido pelo app</span>

</div>

</div>

</section>

<!-- Formulário de Contato -->

<section class="contact-form-section">

<div class="form-container">

<div class="form-header">

<h2>Envie sua Mensagem</h2>

<p>Preencha o formulário abaixo e entraremos em contato o mais breve possível</p>

</div>

<form id="contact-form" class="contact-form">

<div class="form-row">

<div class="form-group">

<label for="contact-name">Nome Completo \*</label>

<input type="text" id="contact-name" placeholder="Digite seu nome completo" required>

</div>

<div class="form-group">

<label for="contact-email">E-mail \*</label>

<input type="email" id="contact-email" placeholder="Digite seu e-mail" required>

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="contact-phone">Telefone</label>

<input type="tel" id="contact-phone" placeholder="(11) 99999-9999">

</div>

<div class="form-group">

<label for="contact-subject">Assunto \*</label>

<select id="contact-subject" required>

<option value="">Selecione um assunto</option>

<option value="duvida">Dúvida sobre serviços</option>

<option value="orcamento">Solicitação de orçamento</option>

<option value="reclamacao">Reclamação</option>

<option value="sugestao">Sugestão</option>

<option value="parceria">Proposta de parceria</option>

<option value="outro">Outro</option>

</select>

</div>

</div>

<div class="form-group">

<label for="contact-message">Mensagem \*</label>

<textarea id="contact-message" rows="5" placeholder="Descreva sua dúvida, solicitação ou mensagem" required></textarea>

</div>

<button type="submit" class="btn btn-primary btn-block">Enviar Mensagem</button>

</form>

</div>

<div class="contact-extra-info">

<div class="info-card">

<h3><i class="fas fa-clock"></i> Horário de Atendimento</h3>

<p>Segunda a Sexta: 8h às 18h</p>

<p>Sábado: 9h às 13h</p>

<p>Domingo: Fechado</p>

</div>

<div class="info-card">

<h3><i class="fas fa-info-circle"></i> Resposta Rápida</h3>

<p>Garantimos retorno em até 24 horas úteis para todas as solicitações recebidas através do nosso formulário.</p>

</div>

</div>

</section>

</div>

</main>

<footer class="footer">

<div class="container">

<div class="footer-content">

<div class="footer-brand">

<img src="/img/fjkaemfkemif.png" alt="OilSmart" class="footer-logo">

<p>Soluções completas para seu veículo</p>

</div>

<div class="footer-links">

<h3>Links Úteis</h3>

<ul>

<li><a href="/html/politicas.html">Política de Privacidade</a></li>

<li><a href="/html/termos.html">Termos de Serviço</a></li>

<li><a href="/html/faq.html">Perguntas Frequentes</a></li>

</ul>

</div>

<div class="footer-contact">

<h3>Contato</h3>

<p><i class="fas fa-phone"></i> (11) 4002-8922</p>

<p><i class="fas fa-envelope"></i> contato@OilSmart.com.br</p>

</div>

</div>

<div class="copyright">

<p>© 2025 OilSmart. Todos os direitos reservados.</p>

</div>

</div>

</footer>

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

</body>

</html>

# Arquivo: public/html/esqueci-senha.html

<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

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

<title>Recuperar Senha - OilSmart</title>

<link rel="stylesheet" href="/css/login.css">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon">

<style>

/\* Estilos para o indicador de carregamento \*/

.loading-overlay {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

display: flex;

justify-content: center;

align-items: center;

z-index: 1000;

display: none;

}

.loading-spinner {

width: 50px;

height: 50px;

border: 5px solid #f3f3f3;

border-top: 5px solid #3498db;

border-radius: 50%;

animation: spin 1s linear infinite;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

.btn:disabled {

opacity: 0.6;

cursor: not-allowed;

}

</style>

</head>

<body>

<!-- Overlay de carregamento -->

<div class="loading-overlay" id="loadingOverlay">

<div class="loading-spinner"></div>

</div>

<div class="login-card">

<div class="logo-container">

<div class="logo">OS</div>

<h1 class="login-title">Recuperar Senha</h1>

</div>

<p class="login-subtitle">Informe seu e-mail para receber instruções de redefinição</p>

<div class="divider"></div>

<form id="recoveryForm">

<div class="form-group">

<label for="recovery-email" class="section-title">E-mail</label>

<input type="email" id="recovery-email" class="input-field" placeholder="seu.email@exemplo.com" required>

</div>

<div class="divider"></div>

<div class="button-group">

<button type="button" class="btn btn-secondary" id="backButton" onclick="window.location.href='login.html'">Voltar</button>

<button type="submit" class="btn btn-primary" id="submitButton">Enviar instruções</button>

</div>

<p class="register-text">Lembrou sua senha? <a href="login.html" class="register-link">Faça login aqui</a></p>

</form>

</div>

<script>

document.getElementById('recoveryForm').addEventListener('submit', async function (e) {

e.preventDefault();

const email = document.getElementById('recovery-email').value.trim();

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

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

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

if (!email) {

alert('Por favor, informe seu e-mail');

return;

}

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailRegex.test(email)) {

alert('Por favor, informe um e-mail válido');

return;

}

// Mostrar carregamento e desabilitar botões

loadingOverlay.style.display = 'flex';

submitButton.disabled = true;

backButton.disabled = true;

try {

const response = await fetch('http://localhost:3000/api/auth/forgot-password', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ email })

});

const result = await response.json();

if (result.success) {

alert('Instruções de recuperação enviadas para seu e-mail!');

window.location.href = 'login.html';

} else {

alert(result.message);

}

} catch (error) {

console.error('Erro ao solicitar recuperação:', error);

alert('Erro ao conectar com o servidor. Tente novamente.');

} finally {

// Sempre esconder o carregamento e reabilitar os botões

loadingOverlay.style.display = 'none';

submitButton.disabled = false;

backButton.disabled = false;

}

});

</script>

</body>

</html>

# Arquivo: public/html/faq.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>Perguntas Frequentes - OilSmart</title>

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<link rel="stylesheet" href="../css/main.css">

</head>

<body>

<!-- Header -->

<header class="header">

<div class="container">

<nav class="navbar">

<a href="index.html" class="logo">

<img src="../img/LogoIOil.png" alt="OilSmart">

</a>

<div class="nav" id="nav">

<ul class="nav\_\_list">

<li><a href="index.html">Início</a></li>

<li><a href="servicos.html">Serviços</a></li>

<li><a href="sobre.html">Sobre</a></li>

<li><a href="agenda.html">Agenda</a></li>

<li><a href="contato.html">Contatos</a></li>

<li><a href="faq.html" class="active">FAQ</a></li>

</ul>

</div>

<div class="action-icons">

<button class="hamburger" id="hamburger">

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

</button>

</div>

</nav>

</div>

</header>

<!-- Main Content -->

<main class="main-content">

<div class="container">

<h1 class="page-title">Perguntas Frequentes</h1>

<p class="page-subtitle">Encontre respostas para as dúvidas mais comuns sobre nossos serviços e plataforma.</p>

<div class="faq-section">

<div class="faq-container">

<div class="faq-item">

<div class="faq-question">

<h3>Como faço para agendar uma troca de óleo?</h3>

<i class="fas fa-chevron-down"></i>

</div>

<div class="faq-answer">

<p>Basta acessar a página "Serviços", selecionar o tipo de serviço desejado, informar os dados do seu veículo e escolher a oficina parceira mais próxima. Em seguida, selecione a data e horário disponíveis para finalizar o agendamento.</p>

</div>

</div>

<div class="faq-item">

<div class="faq-question">

<h3>Preciso me cadastrar para usar o site?</h3>

<i class="fas fa-chevron-down"></i>

</div>

<div class="faq-answer">

<p>Sim, o cadastro é necessário para que possamos armazenar os dados do seu veículo, histórico de serviços e permitir o agendamento com nossas oficinas parceiras. Seus dados são protegidos conforme nossa Política de Privacidade.</p>

</div>

</div>

<div class="faq-item">

<div class="faq-question">

<h3>Quais dados são coletados pela OilSmart?</h3>

<i class="fas fa-chevron-down"></i>

</div>

<div class="faq-answer">

<p>Coletamos dados como nome, e-mail, telefone, localização (com sua permissão) e informações do seu veículo (marca, modelo, ano e tipo de motor) para oferecer recomendações personalizadas e agendamentos eficientes, conforme detalhado em nossa Política de Privacidade.</p>

</div>

</div>

<div class="faq-item">

<div class="faq-question">

<h3>Posso cancelar um agendamento?</h3>

<i class="fas fa-chevron-down"></i>

</div>

<div class="faq-answer">

<p>Sim, você pode cancelar com até 1 hora de antecedência diretamente pela área de "Agenda" em seu perfil. Após esse período, entre em contato diretamente com a oficina para verificar as possibilidades de cancelamento.</p>

</div>

</div>

<div class="faq-item">

<div class="faq-question">

<h3>A OilSmart realiza os serviços diretamente?</h3>

<i class="fas fa-chevron-down"></i>

</div>

<div class="faq-answer">

<p>Não. A OilSmart atua como intermediária, conectando você com oficinas parceiras devidamente verificadas. A execução dos serviços é de responsabilidade exclusiva das oficinas selecionadas.</p>

</div>

</div>

<div class="faq-item">

<div class="faq-question">

<h3>Como posso entrar em contato com o suporte?</h3>

<i class="fas fa-chevron-down"></i>

</div>

<div class="faq-answer">

<p>Você pode nos contatar pelo e-mail <strong>contato@oilsmart.com.br</strong> ou através do formulário de contato disponível em nossa página "Contatos". Nosso time responde em até 24 horas úteis.</p>

</div>

</div>

<div class="faq-item">

<div class="faq-question">

<h3>Como funciona o sistema de recomendações de óleo?</h3>

<i class="fas fa-chevron-down"></i>

</div>

<div class="faq-answer">

<p>Nosso sistema utiliza as informações do seu veículo (marca, modelo, ano e tipo de motor) cruzadas com as especificações dos fabricantes e condições de uso para recomendar o óleo mais adequado. Também consideramos seu estilo de direção e as condições climáticas da sua região.</p>

</div>

</div>

<div class="faq-item">

<div class="faq-question">

<h3>As oficinas parceiras são confiáveis?</h3>

<i class="fas fa-chevron-down"></i>

</div>

<div class="faq-answer">

<p>Todas as oficinas em nossa rede passam por um processo de verificação que inclui análise de certificações, avaliações de clientes e visita técnica. Mantemos um sistema de avaliações para garantir a qualidade contínua dos serviços prestados.</p>

</div>

</div>

</div>

</div>

</div>

</main>

<!-- Footer -->

<footer class="footer">

<div class="container">

<div class="footer-content">

<div class="footer-brand">

<img src="../img/LogoIOil.png" alt="OilSmart">

<p>Conectando você às melhores oficinas e produtos para o cuidado do seu veículo.</p>

</div>

<div class="footer-links">

<h3>Links Rápidos</h3>

<ul>

<li><a href="index.html">Início</a></li>

<li><a href="servicos.html">Serviços</a></li>

<li><a href="sobre.html">Sobre Nós</a></li>

<li><a href="agenda.html">Minha Agenda</a></li>

<li><a href="contato.html">Contato</a></li>

</ul>

</div>

<div class="footer-links">

<h3>Suporte</h3>

<ul>

<li><a href="faq.html">Perguntas Frequentes</a></li>

<li><a href="politicas.html">Política de Privacidade</a></li>

<li><a href="termos.html">Termos de Serviço</a></li>

<li><a href="contato.html">Fale Conosco</a></li>

</ul>

</div>

<div class="footer-contact">

<h3>Contato</h3>

<p><i class="fas fa-envelope"></i> contato@oilsmart.com.br</p>

<p><i class="fas fa-phone"></i> (11) 3456-7890</p>

<p><i class="fas fa-map-marker-alt"></i> São Paulo, Brasil</p>

</div>

</div>

<div class="copyright">

<p>&copy; 2025 OilSmart. Todos os direitos reservados.</p>

</div>

</div>

</footer>

<!-- Login Modal -->

<!-- Loading Overlay -->

<div id="loading-overlay">

<div class="loader"></div>

</div>

<!-- Toast Notification -->

<div class="toast" id="toast"></div>

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

<script>

// FAQ toggle functionality

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

const faqItems = document.querySelectorAll('.faq-item');

faqItems.forEach(item => {

const question = item.querySelector('.faq-question');

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

// Close all other items

faqItems.forEach(otherItem => {

if (otherItem !== item && otherItem.classList.contains('active')) {

otherItem.classList.remove('active');

}

});

// Toggle current item

item.classList.toggle('active');

});

});

// Initialize modals and other functionality

initModals();

initMobileMenu();

});

function initModals() {

// Modal functionality would be implemented here

}

function initMobileMenu() {

// Mobile menu functionality would be implemented here

}

</script>

</body>

</html>

# Arquivo: public/html/index.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>OilSmart - Serviços Automotivos Especializados</title>

<meta name="description"

content="Agendamento de troca de óleo e serviços automotivos. Encontre os melhores produtos e oficinas parceiras.">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap"

rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<link rel="stylesheet" href="/css/main.css">

<link rel="stylesheet" href="/css/auth.css">

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

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon"></head>

<body>

<header class="header">

<div class="container">

<div class="navbar">

<button class="hamburger" id="hamburger">

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

</button>

<div class="logo">

<a href="index.html"><img src="/img/oilamarelo.png" alt="OilSmart"></a>

</div>

<nav class="nav" id="nav">

<ul class="nav\_\_list">

<li><a href="/html/index.html" class="active">Início</a></li>

<li><a href="/html/servicos.html">Serviços</a></li>

<li><a href="/html/agenda.html">Agenda</a></li>

<li><a href="/html/sobre.html">Sobre</a></li>

<li><a href="/html/contato.html">Contato</a></li>

</ul>

</nav>

<div class="action-icons">

<div class="user-dropdown" id="user-dropdown">

<div class="user-icon" id="login-btn">

<i class="fas fa-user"></i>

<span class="login-status" id="login-status">Login</span>

</div>

<div class="dropdown-menu" id="dropdown-menu">

<div class="user-info">

<strong id="user-display-name">Usuário</strong>

<span id="user-email">email@exemplo.com</span>

</div>

<div class="dropdown-divider"></div>

<a href="/html/agenda.html" class="dropdown-item">

<i class="fas fa-calendar-alt"></i> Meus Agendamentos

</a>

<div class="dropdown-divider"></div>

<a href="/html/index.html" id="logout-btn" class="dropdown-item">

<i class="fas fa-sign-out-alt"></i> Sair

</a>

</div>

</div>

</div>

</header>

<!-- <li><a href="/html/index.html" class="active">Início</a></li> -->

<main class="main-content">

<!-- Banner Hero com Carrossel e Texto -->

<section class="hero-banner">

<!-- Carrossel como fundo -->

<div class="hero-carousel">

<div class="carousel-track">

<div class="carousel-slide"><img src="/img/carro1.jpg" alt="Carro 1"></div>

<div class="carousel-slide"><img src="/img/carro2.jpg" alt="Carro 2"></div>

<div class="carousel-slide"><img src="/img/carro3.jpg" alt="Carro 3"></div>

<div class="carousel-slide"><img src="/img/carro4.jpg" alt="Carro 4"></div>

</div>

</div>

<!-- Overlay escuro com blur -->

<div class="hero-overlay"></div>

<!-- Conteúdo do hero em destaque -->

<div class="hero-content container">

<h1>Troca de Óleo com Qualidade e Agilidade</h1>

<p>Agende online e ganhe 10% de desconto na primeira troca</p>

<a href="/html/servicos.html" class="btn btn-primary">Agendar Serviço</a>

</div>

</section>

<div class="hero-divider"></div>

<style>

/\* ================= HERO BANNER ================= \*/

.hero-banner {

position: relative;

width: 100%;

height: 65vh;

min-height: 450px;

max-height: 650px;

overflow: hidden;

display: flex;

align-items: center;

justify-content: center;

color: white;

text-align: center;

isolation: isolate;

}

/\* Carrossel de fundo \*/

.hero-carousel {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

z-index: 1;

overflow: hidden;

}

.hero-carousel .carousel-track {

display: flex;

height: 100%;

transition: transform 0.8s cubic-bezier(0.23, 1, 0.32, 1);

}

.hero-carousel .carousel-slide {

min-width: 100%;

height: 100%;

position: relative;

}

.hero-carousel img {

width: 100%;

height: 100%;

object-fit: cover;

filter: brightness(0.7) contrast(1.1);

transition: transform 10s linear;

}

.hero-carousel .carousel-slide.active img {

transform: scale(1.05);

}

/\* Overlay escuro com blur \*/

.hero-overlay {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: linear-gradient(

135deg,

rgba(0, 0, 0, 0.5) 0%,

rgba(0, 0, 0, 0.3) 50%,

rgba(0, 0, 0, 0.5) 100%

);

backdrop-filter: blur(5px);

z-index: 2;

}

/\* Conteúdo do banner \*/

.hero-content {

position: relative;

z-index: 3;

padding: 2rem;

max-width: 900px;

animation: fadeInUp 1s ease-out 0.5s both;

}

.hero-content h1 {

font-size: 3rem;

font-weight: 700;

text-shadow: 0 2px 10px rgba(0,0,0,0.5);

}

.hero-content p {

font-size: 1.3rem;

margin: 1rem 0 2rem;

text-shadow: 0 2px 5px rgba(0,0,0,0.5);

}

/\* Botão principal \*/

.btn-primary {

background: linear-gradient(135deg, #f7b900, #E05D00);

border: none;

padding: 14px 35px;

font-size: 1.1rem;

border-radius: 50px;

transition: all 0.3s ease;

box-shadow: 0 5px 15px rgba(224, 93, 0, 0.4);

position: relative;

overflow: hidden;

}

.btn-primary::before {

content: '';

position: absolute;

top: 0;

left: -100%;

width: 100%;

height: 100%;

background: linear-gradient(90deg, transparent, rgba(255,255,255,0.2), transparent);

transition: 0.5s;

}

.btn-primary:hover {

transform: translateY(-3px);

box-shadow: 0 8px 20px rgba(224, 93, 0, 0.6);

}

.btn-primary:hover::before {

left: 100%;

}

/\* Indicadores do carrossel \*/

.carousel-indicators {

position: absolute;

bottom: 25px;

left: 50%;

transform: translateX(-50%);

display: flex;

gap: 10px;

z-index: 4;

}

.carousel-indicator {

width: 12px;

height: 12px;

border-radius: 50%;

background: rgba(255, 255, 255, 0.5);

cursor: pointer;

transition: all 0.3s ease;

}

.carousel-indicator.active {

background: #f7b900;

transform: scale(1.2);

box-shadow: 0 0 8px rgba(247, 185, 0, 0.8);

}

/\* Barra de progresso \*/

.progress-bar {

position: absolute;

bottom: 0;

left: 0;

height: 4px;

width: 100%;

background: rgba(255, 255, 255, 0.2);

z-index: 4;

}

.progress-fill {

height: 100%;

width: 0%;

background: #f7b900;

transition: width 0.1s linear;

}

/\* Animação do conteúdo \*/

@keyframes fadeInUp {

from {

opacity: 0;

transform: translateY(40px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

/\* ================= DIVISOR ENTRE BANNER E SITE ================= \*/

.hero-divider {

width: 100%;

height: 40px;

background: linear-gradient(to right, #f7b900, #E05D00);

box-shadow: 0 5px 15px rgba(0,0,0,0.2);

border-radius: 0 0 20px 20px;

margin-bottom: 2rem;

}

/\* ================= RESPONSIVO ================= \*/

@media (max-width: 992px) {

.hero-content h1 { font-size: 2.4rem; }

}

@media (max-width: 768px) {

.hero-banner { height: 55vh; min-height: 400px; }

.hero-content h1 { font-size: 2rem; }

.hero-content p { font-size: 1.1rem; }

.btn-primary { padding: 12px 30px; font-size: 1rem; }

}

@media (max-width: 576px) {

.hero-banner { min-height: 350px; }

.hero-content h1 { font-size: 1.7rem; }

.hero-content { padding: 1.5rem; }

}

</style>

<div class="container main-grid">

<!-- Conteúdo Principal -->

<div class="primary-content">

<!-- Carrossel de Serviços -->

<section class="services-carousel">

<h2 class="section-title">Nossos Serviços Automotivos</h2>

<div class="carousel-container">

<div class="carousel-track">

<div class="carousel-slide">

<div class="service-card">

<div class="service-icon">

<i class="fas fa-oil-can"></i>

</div>

<h3>Troca de Óleo Completa</h3>

<p>Óleo e filtro recomendados automaticamente para o seu veículo com base em marca,

modelo e quilometragem.</p>

<span class="price">Produtos homologados,<br>Agendamento online,<br>Histórico e

lembretes automáticos;</span>

<a href="/html/servicos.html" class="btn btn-secondary">Agendar</a>

</div>

</div>

<div class="carousel-slide">

<div class="service-card">

<div class="service-icon">

<i class="fas fa-tools"></i>

</div>

<h3>Revisão Lubrificação</h3>

<p>Verificamos o estado do óleo, filtro, fluido de freio e lubrificantes essenciais

para garantir o bom funcionamento do motor.</p>

<span class="price">Avaliação técnica,<br>Recomendação personalizada,<br>Integrado

ao seu histórico;</span>

<a href="/html/servicos.html" class="btn btn-secondary">Agendar</a>

</div>

</div>

<div class="carousel-slide">

<div class="service-card">

<div class="service-icon">

<i class="fas fa-search-plus"></i>

</div>

<h3>Diagnóstico de Óleo Inteligente</h3>

<p>Análise rápida para saber se está na hora de trocar — mesmo sem saber a

quilometragem.</p>

<span class="price">Consulta de cor, textura e viscosidade,<br>Apoio de IA para

recomendação,<br>Mais segurança para você e seu carro;

</span>

<a href="/html/servicos.html" class="btn btn-secondary">Agendar</a>

</div>

</div>

<div class="carousel-slide">

<div class="service-card">

<div class="service-icon">

<i class="fas fa-calendar-check"></i>

</div>

<h3>Controle de Trocas</h3>

<p>Fique tranquilo! A OilSmart te avisa quando for hora de trocar o óleo novamente.

</p>

<span class="price">Notificações por e-mail,<br>Baseado em uso real do

veículo,<br>Fidelização com oficinas parceiras;</span>

<a href="/html/servicos.html" class="btn btn-secondary">Agendar</a>

</div>

</div>

</div>

</div>

</section>

<!-- Seção de Informações sobre Troca de Óleo -->

<section class="oil-info-section">

<h2 class="section-title">A Importância da Troca de Óleo Regular</h2>

<div class="info-grid">

<div class="info-card">

<i class="fas fa-cog"></i>

<h3>Proteção do Motor</h3>

<p>O óleo lubrifica as peças móveis do motor, reduzindo o atrito e o desgaste, prolongando a

vida útil do seu veículo.</p>

</div>

<div class="info-card">

<i class="fas fa-temperature-high"></i>

<h3>Controle de Temperatura</h3>

<p>O óleo ajuda a dissipar o calor gerado pela combustão e pelo atrito, evitando

superaquecimento.</p>

</div>

<div class="info-card">

<i class="fas fa-gas-pump"></i>

<h3>Economia de Combustível</h3>

<p>Um motor bem lubrificado trabalha com mais eficiência, reduzindo o consumo de

combustível.</p>

</div>

<div class="info-card">

<i class="fas fa-shield-alt"></i>

<h3>Prevenção de Corrosão</h3>

<p>Aditivos no óleo protegem as peças metálicas contra oxidação e corrosão.</p>

</div>

</div>

</section>

<!-- Ofertas Especiais -->

<section class="special-offers">

<h2 class="section-title">Promoções Exclusivas OilSmart</h2>

<div class="offer-cards">

<!-- Oferta destaque -->

<div class="offer-card highlight">

<div class="offer-badge">Super Oferta</div>

<h3>Troca Completa Premium</h3>

<p>Troca de óleo + filtro original + inspeção rápida gratuita</p>

<div class="price-container">

<span class="old-price">R$ 199,90</span>

<span class="new-price">R$ 159,90</span>

<span class="discount">20% OFF</span>

</div>

<a href="/html/servicos.html" class="btn btn-primary">Agendar Agora</a>

</div>

<!-- Primeira troca -->

<div class="offer-card">

<h3>Primeira Troca</h3>

<p>Ganhe 15% de desconto na sua primeira troca de óleo na OilSmart</p>

<div class="price-container">

<span class="new-price">15% OFF</span>

</div>

<a href="/html/servicos.html" class="btn btn-secondary">Agendar</a>

</div>

<!-- Programa Fidelidade -->

<div class="offer-card">

<h3>Clube Fidelidade</h3>

<p>A cada 3 trocas de óleo, você ganha a 4ª troca com o filtro grátis</p>

<div class="price-container">

<span class="new-price">Filtro 100% Grátis</span>

</div>

<a href="/html/servicos.html" class="btn btn-secondary">Participar</a>

</div>

</div>

</section>

</div>

<!-- Sidebar -->

<aside class="sidebar">

</aside>

</div>

</main>

<footer class="footer">

<div class="container">

<div class="footer-content">

<div class="footer-brand">

<img src="/img/fjkaemfkemif.png" alt="OilSmart" class="footer-logo">

<p>Soluções completas para seu veículo</p>

</div>

<div class="footer-links">

<h3>Links Úteis</h3>

<ul>

<li><a href="/html/politicas.html">Política de Privacidade</a></li>

<li><a href="/html/termos.html">Termos de Serviço</a></li>

<li><a href="/html/faq.html">Perguntas Frequentes</a></li>

</ul>

</div>

<div class="footer-contact">

<h3>Contato</h3>

<p><i class="fas fa-phone"></i> (11) 4002-8922</p>

<p><i class="fas fa-envelope"></i> contato@OilSmart.com.br</p>

</div>

</div>

<div class="copyright">

<p>© 2025 OilSmart. Todos os direitos reservados.</p>

</div>

</div>

</footer>

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

<script>

// Script aprimorado para o carrossel

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

const track = document.querySelector(".hero-carousel .carousel-track");

const slides = document.querySelectorAll(".hero-carousel .carousel-slide");

const indicators = document.querySelectorAll(".carousel-indicator");

let index = 0;

let interval;

function updateCarousel() {

track.style.transform = `translateX(-${index \* 100}%)`;

// Atualiza classes ativas

slides.forEach(slide => slide.classList.remove('active'));

slides[index].classList.add('active');

indicators.forEach(indicator => indicator.classList.remove('active'));

indicators[index].classList.add('active');

}

// Inicia auto-play

function startAutoPlay() {

interval = setInterval(() => {

index = (index + 1) % slides.length;

updateCarousel();

}, 4000);

}

// Para auto-play ao interagir

function stopAutoPlay() {

clearInterval(interval);

}

// Adiciona eventos aos indicadores

indicators.forEach(indicator => {

indicator.addEventListener('click', function() {

stopAutoPlay();

index = parseInt(this.getAttribute('data-index'));

updateCarousel();

startAutoPlay();

});

});

// Pausa ao passar o mouse

const heroBanner = document.querySelector('.hero-banner');

heroBanner.addEventListener('mouseenter', stopAutoPlay);

heroBanner.addEventListener('mouseleave', startAutoPlay);

// Inicia o carrossel

startAutoPlay();

function checkLoginStatus() {

const isLoggedIn = localStorage.getItem('isLoggedIn') === 'true';

const userDropdown = document.getElementById('user-dropdown');

const loginStatus = document.getElementById('login-status');

if (userDropdown && loginStatus) {

if (isLoggedIn) {

userDropdown.classList.add('user-logged-in');

const userFullName = localStorage.getItem('userFullName');

const userEmail = localStorage.getItem('userEmail');

if (userFullName) {

loginStatus.textContent = userFullName;

} else if (userEmail) {

loginStatus.textContent = userEmail.split('@')[0];

} else {

loginStatus.textContent = 'Minha Conta';

}

} else {

userDropdown.classList.remove('user-logged-in');

loginStatus.textContent = 'Login';

}

}

}

// Inicializar componentes

initHeroCarousel();

checkLoginStatus();

});

</script>

</body>

</html>

# Arquivo: public/html/login.html

<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

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

<title>Login - OilSmart</title>

<link rel="stylesheet" href="/css/login.css">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon">

</head>

<body>

<div class="login-card">

<div class="logo-container">

<div class="logo">OS</div>

<h1 class="login-title">Bem-vindo de volta</h1>

</div>

<p class="login-subtitle">Acesse sua conta para continuar gerenciando seus veículos</p>

<div class="divider"></div>

<form id="loginForm">

<div class="form-group">

<label for="email" class="section-title">E-mail</label>

<input type="email" id="email" class="input-field" placeholder="seu.email@exemplo.com" required>

</div>

<div class="form-group">

<label for="password" class="section-title">Senha</label>

<input type="password" id="password" class="input-field" placeholder="Digite sua senha" required>

<a href="esqueci-senha.html" class="forgot-password">Esqueceu sua senha?</a>

</div>

<div class="password-options">

<label class="checkbox-container">

<input type="checkbox" id="remember-me">

<span class="checkmark"></span>

Manter conectado

</label>

</div>

<div class="divider"></div>

<div class="button-group">

<button type="button" class="btn btn-secondary" onclick="window.location.href='index.html'">Voltar</button>

<button type="submit" class="btn btn-primary">Acessar minha conta</button>

</div>

<p class="register-text">Não tem uma conta? <a href="cadastro.html" class="register-link">Cadastre-se gratuitamente</a></p>

</form>

</div>

<script>

document.getElementById('loginForm').addEventListener('submit', async function (e) {

e.preventDefault();

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

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

const rememberMe = document.getElementById('remember-me').checked;

try {

console.log('Tentando login com:', email);

// Primeiro teste se o servidor está respondendo

const testResponse = await fetch('http://localhost:3000/api/test');

if (!testResponse.ok) {

throw new Error('Servidor não está respondendo');

}

// Fazer login

const response = await fetch('http://localhost:3000/api/auth/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

email: email,

senha: senha

})

});

console.log('Status da resposta:', response.status);

// Verificar se a resposta é JSON

const contentType = response.headers.get('content-type');

if (!contentType || !contentType.includes('application/json')) {

const text = await response.text();

console.error('Resposta não é JSON:', text);

throw new Error('Resposta inválida do servidor');

}

const result = await response.json();

console.log('Resultado do login:', result);

const urlParams = new URLSearchParams(window.location.search);

const redirect = urlParams.get('redirect');

if (result.success) {

// Login bem-sucedido

alert(`Bem-vindo(a) de volta, ${result.user.nome || email.split('@')[0]}!`);

// Armazenar informações do usuário

localStorage.setItem('user', JSON.stringify(result.user));

localStorage.setItem('isLoggedIn', 'true');

localStorage.setItem('userId', result.user.id);

localStorage.setItem('userId', result.user.id);

localStorage.setItem('token', result.token);

// Lembrar email se solicitado

if (rememberMe) {

localStorage.setItem('rememberedEmail', email);

} else {

localStorage.removeItem('rememberedEmail');

}

// Redirecionar para a página inicial

window.location.href = 'index.html';

}

else {

alert(result.message);

}

} catch (error) {

console.error('Erro completo no login:', error);

alert('Erro ao conectar com o servidor. Verifique se o servidor está rodando: node server.js');

}

});

// Preencher email salvo ao carregar a página

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

const storedEmail = localStorage.getItem('rememberedEmail');

if (storedEmail) {

const emailInput = document.getElementById('email');

if (emailInput) {

emailInput.value = storedEmail;

document.getElementById('remember-me').checked = true;

}

}

// Testar conexão com servidor

testServerConnection();

});

async function testServerConnection() {

try {

const response = await fetch('http://localhost:3000/api/test');

if (response.ok) {

const data = await response.json();

console.log('✅ Servidor conectado:', data);

}

} catch (error) {

console.warn('❌ Não foi possível conectar ao servidor:', error.message);

}

}

</script>

</body>

</html>

# Arquivo: public/html/politicas.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>Política de Privacidade - OilSmart</title>

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<link rel="stylesheet" href="../css/main.css">

</head>

<body>

<!-- Header -->

<header class="header">

<div class="container">

<nav class="navbar">

<a href="index.html" class="logo">

<img src="../img/LogoIOil.png" alt="OilSmart">

</a>

<div class="nav" id="nav">

<ul class="nav\_\_list">

<li><a href="index.html">Início</a></li>

<li><a href="servicos.html">Serviços</a></li>

<li><a href="sobre.html">Sobre</a></li>

<li><a href="agenda.html">Agenda</a></li>

<li><a href="contatos.html">Contatos</a></li>

<li><a href="faq.html">FAQ</a></li>

</ul>

</div>

<div class="action-icons">

<div class="user-dropdown">

<div class="user-ico">

<i class="fas fa-user-circle"></i>

<span class="login-status">Entrar</span>

</div>

<div class="dropdown-menu">

<div class="user-info">

<strong>Visitante</strong>

<span>Entre ou cadastre-se</span>

</div>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item" data-modal="login">

<i class="fas fa-sign-in-alt"></i>

Fazer Login

</a>

<a href="#" class="dropdown-item" data-modal="register">

<i class="fas fa-user-plus"></i>

Criar Conta

</a>

</div>

</div>

<button class="hamburger" id="hamburger">

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

</button>

</div>

</nav>

</div>

</header>

<!-- Main Content -->

<main class="main-content">

<div class="container">

<h1 class="page-title">Política de Privacidade</h1>

<p class="page-subtitle">Entenda como coletamos, usamos e protegemos suas informações.</p>

<div class="privacy-content" style="background: var(--card-bg); padding: 40px; border-radius: var(--border-radius); box-shadow: var(--card-shadow); margin-top: 30px;">

<p>A sua privacidade é importante para nós. Esta política explica como coletamos, usamos, compartilhamos e protegemos suas informações pessoais quando você utiliza os serviços da OilSmart.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">1. Informações Coletadas</h2>

<p>Coletamos os seguintes tipos de informações quando você utiliza nossa plataforma:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li><strong>Informações de cadastro:</strong> nome, e-mail, telefone, endereço</li>

<li><strong>Dados do veículo:</strong> marca, modelo, ano, tipo de motor, quilometragem</li>

<li><strong>Informações de localização:</strong> coletadas com sua permissão para encontrar oficinas próximas</li>

<li><strong>Dados de uso:</strong> histórico de serviços, preferências, interações com a plataforma</li>

<li><strong>Informações de pagamento:</strong> processadas de forma segura por nossos parceiros</li>

</ul>

<h2 style="color: var(--secondary-color); margin-top: 30px;">2. Uso dos Dados</h2>

<p>Utilizamos suas informações para:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li>Fornecer recomendações personalizadas de óleo para seu veículo</li>

<li>Localizar e conectar você com oficinas parceiras próximas</li>

<li>Gerenciar agendamentos de serviços e enviar lembretes</li>

<li>Melhorar continuamente a experiência do usuário em nossa plataforma</li>

<li>Enviar comunicações relevantes sobre serviços, promoções e novidades (com sua permissão)</li>

<li>Garantir a segurança de sua conta e de nossos serviços</li>

</ul>

<h2 style="color: var(--secondary-color); margin-top: 30px;">3. Compartilhamento de Informações</h2>

<p>Seus dados pessoais não são vendidos a terceiros. Podemos compartilhar informações nas seguintes situações:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li><strong>Oficinas parceiras:</strong> informações necessárias para realizar os serviços agendados</li>

<li><strong>Prestadores de serviços:</strong> empresas que nos auxiliam na operação da plataforma (sempre com contratos de confidencialidade)</li>

<li><strong>Obrigações legais:</strong> quando exigido por lei, regulamento ou processo legal</li>

<li><strong>Proteção de direitos:</strong> para proteger os direitos, propriedade ou segurança da OilSmart, nossos usuários ou do público</li>

</ul>

<h2 style="color: var(--secondary-color); margin-top: 30px;">4. Segurança dos Dados</h2>

<p>Implementamos medidas técnicas e organizacionais robustas para proteger suas informações:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li>Criptografia de dados em trânsito e em repouso</li>

<li>Autenticação segura e controle de acesso</li>

<li>Monitoramento contínuo de segurança</li>

<li>Backups regulares e planos de recuperação de desastres</li>

<li>Treinamento de equipe em proteção de dados</li>

</ul>

<h2 style="color: var(--secondary-color); margin-top: 30px;">5. Seus Direitos</h2>

<p>Você tem o direito de:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li>Acessar e revisar suas informações pessoais</li>

<li>Corrigir dados inexatos ou incompletos</li>

<li>Solicitar a exclusão de seus dados pessoais</li>

<li>Opor-se ao processamento de seus dados para determinadas finalidades</li>

<li>Receber seus dados em formato estruturado e portátil</li>

<li>Retirar seu consentimento a qualquer momento</li>

</ul>

<p>Para exercer esses direitos, entre em contato conosco através do e-mail <strong>contato@oilsmart.com.br</strong>.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">6. Retenção de Dados</h2>

<p>Manteremos suas informações pessoais apenas pelo tempo necessário para cumprir as finalidades descritas nesta política, respeitando os prazos legais aplicáveis. Após esse período, seus dados serão anonimizados ou excluídos com segurança.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">7. Cookies e Tecnologias Semelhantes</h2>

<p>Utilizamos cookies e tecnologias similares para melhorar sua experiência, analisar o uso de nossos serviços e personalizar conteúdo e anúncios. Você pode gerenciar suas preferências de cookies através das configurações do seu navegador.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">8. Alterações nesta Política</h2>

<p>Podemos atualizar esta política periodicamente para refletir mudanças em nossas práticas ou por outros motivos operacionais, legais ou regulatórios. A versão mais recente estará sempre disponível em nosso site, com a data de última atualização.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">9. Contato</h2>

<p>Se você tiver dúvidas, preocupações ou solicitações relacionadas a esta Política de Privacidade ou ao tratamento de seus dados pessoais, entre em contato conosco:</p>

<p><strong>E-mail:</strong> contato@oilsmart.com.br<br>

<strong>Endereço:</strong> São Paulo, Brasil</p>

<p class="last-updated" style="margin-top: 40px; font-size: 0.9rem; color: var(--text-light); text-align: right;">Última atualização: Maio de 2025</p>

</div>

</div>

</main>

<!-- Footer -->

<footer class="footer">

<div class="container">

<div class="footer-content">

<div class="footer-brand">

<img src="../img/LogoIOil.png" alt="OilSmart">

<p>Conectando você às melhores oficinas e produtos para o cuidado do seu veículo.</p>

</div>

<div class="footer-links">

<h3>Links Rápidos</h3>

<ul>

<li><a href="index.html">Início</a></li>

<li><a href="servicos.html">Serviços</a></li>

<li><a href="sobre.html">Sobre Nós</a></li>

<li><a href="agenda.html">Minha Agenda</a></li>

<li><a href="contatos.html">Contato</a></li>

</ul>

</div>

<div class="footer-links">

<h3>Suporte</h3>

<ul>

<li><a href="faq.html">Perguntas Frequentes</a></li>

<li><a href="politicas.html">Política de Privacidade</a></li>

<li><a href="termos.html">Termos de Serviço</a></li>

<li><a href="contatos.html">Fale Conosco</a></li>

</ul>

</div>

<div class="footer-contact">

<h3>Contato</h3>

<p><i class="fas fa-envelope"></i> contato@oilsmart.com.br</p>

<p><i class="fas fa-phone"></i> (11) 3456-7890</p>

<p><i class="fas fa-map-marker-alt"></i> São Paulo, Brasil</p>

</div>

</div>

<div class="copyright">

<p>&copy; 2025 OilSmart. Todos os direitos reservados.</p>

</div>

</div>

</footer>

<!-- Login Modal -->

<div class="modal" id="login-modal">

<div class="modal-content">

<button class="close-modal">&times;</button>

<div class="modal-header">

<h2>Fazer Login</h2>

<p>Entre na sua conta para acessar todos os recursos</p>

</div>

<form class="login-form">

<div class="form-group">

<label for="login-email">E-mail</label>

<input type="email" id="login-email" placeholder="Seu e-mail" required>

</div>

<div class="form-group">

<label for="login-password">Senha</label>

<input type="password" id="login-password" placeholder="Sua senha" required>

</div>

<button type="submit" class="btn btn-primary btn-block">Entrar</button>

</form>

<div class="divider">ou</div>

<div class="social-login">

<button class="social-btn google">

<i class="fab fa-google"></i>

Entrar com Google

</button>

</div>

<div class="register-link">

Não tem uma conta? <a href="#" data-modal="register">Cadastre-se</a>

</div>

</div>

</div>

<!-- Register Modal -->

<div class="modal" id="register-modal">

<div class="modal-content">

<button class="close-modal">&times;</button>

<div class="modal-header">

<h2>Criar Conta</h2>

<p>Junte-se à nossa comunidade</p>

</div>

<form class="login-form">

<div class="form-group">

<label for="register-name">Nome Completo</label>

<input type="text" id="register-name" placeholder="Seu nome completo" required>

</div>

<div class="form-group">

<label for="register-email">E-mail</label>

<input type="email" id="register-email" placeholder="Seu e-mail" required>

</div>

<div class="form-group">

<label for="register-password">Senha</label>

<input type="password" id="register-password" placeholder="Crie uma senha" required>

</div>

<div class="form-group">

<label for="register-confirm-password">Confirmar Senha</label>

<input type="password" id="register-confirm-password" placeholder="Repita sua senha" required>

</div>

<button type="submit" class="btn btn-primary btn-block">Criar Conta</button>

</form>

<div class="divider">ou</div>

<div class="social-login">

<button class="social-btn google">

<i class="fab fa-google"></i>

Cadastrar com Google

</button>

</div>

<div class="register-link">

Já tem uma conta? <a href="#" data-modal="login">Faça login</a>

</div>

</div>

</div>

<!-- Loading Overlay -->

<div id="loading-overlay">

<div class="loader"></div>

</div>

<!-- Toast Notification -->

<div class="toast" id="toast"></div>

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

<script>

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

// Initialize modals and mobile menu

initModals();

initMobileMenu();

// Scroll to top

window.scrollTo({ top: 0, behavior: 'smooth' });

});

function initModals() {

// Modal functionality would be implemented here

}

function initMobileMenu() {

// Mobile menu functionality would be implemented here

}

</script>

</body>

</html>

# Arquivo: public/html/redefinir-senha.html

<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

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

<title>Redefinir Senha - OilSmart</title>

<link rel="stylesheet" href="/css/login.css">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon">

</head>

<body>

<div class="login-card">

<div class="logo-container">

<div class="logo">OS</div>

<h1 class="login-title">Redefinir Senha</h1>

</div>

<p class="login-subtitle">Crie uma nova senha para sua conta</p>

<div class="divider"></div>

<form id="resetForm">

<input type="hidden" id="reset-token" value="">

<div class="form-group">

<label for="reset-password" class="section-title">Nova Senha</label>

<input type="password" id="reset-password" class="input-field" placeholder="Crie uma nova senha segura" required>

</div>

<div class="form-group">

<label for="reset-confirm-password" class="section-title">Confirmar Nova Senha</label>

<input type="password" id="reset-confirm-password" class="input-field" placeholder="Digite a mesma senha" required>

</div>

<div class="divider"></div>

<div class="button-group">

<button type="button" class="btn btn-secondary" onclick="window.location.href='login.html'">Voltar</button>

<button type="submit" class="btn btn-primary">Redefinir Senha</button>

</div>

<p class="register-text">Lembrou sua senha? <a href="login.html" class="register-link">Faça login aqui</a></p>

</form>

</div>

<script>

// Extrair token da URL

const urlParams = new URLSearchParams(window.location.search);

const token = urlParams.get('token');

if (token) {

document.getElementById('reset-token').value = token;

} else {

alert('Token de redefinição não encontrado. Solicite uma nova recuperação.');

window.location.href = 'esqueci-senha.html';

}

document.getElementById('resetForm').addEventListener('submit', async function (e) {

e.preventDefault();

const token = document.getElementById('reset-token').value;

const senha = document.getElementById('reset-password').value;

const confirmarSenha = document.getElementById('reset-confirm-password').value;

if (!senha) {

alert('Por favor, informe a nova senha');

return;

}

if (senha.length < 6) {

alert('A senha deve ter pelo menos 6 caracteres');

return;

}

if (senha !== confirmarSenha) {

alert('As senhas não coincidem');

return;

}

try {

const response = await fetch('http://localhost:3000/api/auth/reset-password', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ token, senha })

});

const result = await response.json();

if (result.success) {

alert('Senha redefinida com sucesso!');

window.location.href = 'login.html';

} else {

alert(result.message);

}

} catch (error) {

console.error('Erro ao redefinir senha:', error);

alert('Erro ao conectar com o servidor. Tente novamente.');

}

});

</script>

</body>

</html>

# Arquivo: public/html/servicos.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>Agendamento - Troca de Óleo | OilSmart</title>

<meta name="description"

content="Agende sua troca de óleo com a OilSmart. Encontre o óleo e filtro perfeitos para seu veículo e oficinas parceiras próximas.">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css">

<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/html2canvas/1.4.1/html2canvas.min.js"></script>

<link rel="stylesheet" href="/css/main.css">

<link rel="stylesheet" href="/css/auth.css">

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

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon">

</head>

</head>

<script>

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

const hamburger = document.getElementById("hamburger-menu");

const nav = document.getElementById("nav");

if (hamburger && nav) {

hamburger.addEventListener("click", function () {

nav.classList.toggle("active");

hamburger.classList.toggle("active");

document.body.style.overflow = nav.classList.contains("active") ? "hidden" : "auto";

});

const navLinks = nav.querySelectorAll("a");

navLinks.forEach(link => {

link.addEventListener("click", function () {

if (window.innerWidth <= 992) {

nav.classList.remove("active");

hamburger.classList.remove("active");

document.body.style.overflow = "auto";

}

});

});

}

});

</script>

<body>

<header class="header">

<div class="container">

<div class="navbar">

<button class="hamburger-menu" id="hamburger">

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

</button>

<div class="logo">

<a href="index.html"><img src="/img/oilamarelo.png" alt="OilSmart"></a>

</div>

<nav class="nav" id="nav">

<ul class="nav\_\_list">

<li><a href="index.html">Início</a></li>

<li><a href="#" class="active">Serviços</a></li>

<li><a href="agenda.html">Agenda</a></li>

<li><a href="sobre.html">Sobre</a></li>

<li><a href="/html/contato.html">Contato</a></li>

</ul>

</nav>

<div class="action-icons">

<div class="user-dropdown" id="user-dropdown">

<div class="user-icon" id="login-btn">

<i class="fas fa-user"></i>

<span class="login-status" id="login-status">Login</span>

</div>

<div class="dropdown-menu" id="dropdown-menu">

<div class="user-info">

<strong id="user-display-name">Usuário</strong>

<span id="user-email">email@exemplo.com</span>

</div>

<div class="dropdown-divider"></div>

<a href="/html/agenda.html" class="dropdown-item">

<i class="fas fa-calendar-alt"></i> Meus Agendamentos

</a>

<div class="dropdown-divider"></div>

<a href="/html/index.html" id="logout-btn" class="dropdown-item">

<i class="fas fa-sign-out-alt"></i> Sair

</a>

</div>

</div>

</div>

</div>

</div>

</header>

<main class="main-content">

<div class="container">

<div class="progress-container">

<div class="progress-bar">

<div class="progress-step active">

<div class="step-bubble">1</div>

<span class="step-label">Veículo</span>

</div>

<div class="progress-step">

<div class="step-bubble">2</div>

<span class="step-label">Produtos</span>

</div>

<div class="progress-step">

<div class="step-bubble">3</div>

<span class="step-label">Agendamento</span>

</div>

<div class="progress-step">

<div class="step-bubble">4</div>

<span class="step-label">Confirmação</span>

</div>

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

</div>

</div>

<h1 class="page-title">Agendamento para Troca de Óleo</h1>

<p class="page-subtitle">Encontre o óleo e filtro perfeitos para seu veículo</p>

<div class="service-flow">

<!-- Passo 1: Dados do veículo e localização -->

<section class="service-step active" id="step1">

<h2><span class="step-number">1</span> Dados do Veículo e Localização</h2>

<form id="vehicle-form" class="vehicle-form">

<div class="form-row">

<div class="form-group">

<label for="vehicle-brand">Marca</label>

<select id="vehicle-brand" required>

<option value="">Selecione a marca</option>

</select>

</div>

<div class="form-group">

<label for="vehicle-model">Modelo</label>

<select id="vehicle-model" disabled required>

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

</select>

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="vehicle-year">Ano</label>

<select id="vehicle-year" disabled required>

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

</select>

</div>

<div class="form-group">

<label for="vehicle-mileage">Quilometragem (opcional)</label>

<input type="number" id="vehicle-mileage" placeholder="Ex: 50000">

<small style="display:block; color:#666; font-size:14px; margin-top:4px;">

Informe a quilometragem atual do veículo, se souber.

</small>

</div>

</div>

<div class="form-group">

<label for="location-input">Digite seu endereço</label>

<div class="location-search">

<div class="autocomplete-container">

<input type="text" id="location-input"

placeholder="Ex: Av. Paulista, 1000 - São Paulo/SP">

<div id="autocomplete-results" class="autocomplete-results"></div>

</div>

<button type="button" id="location-button" class="btn btn-primary">

<i class="fas fa-search"></i> Buscar

</button>

</div>

<div class="location-or">

<span>ou</span>

</div>

<button type="button" id="user-location-btn" class="btn btn-secondary">

<i class="fas fa-location-arrow"></i> Usar minha localização

</button>

</div>

<div class="map-container">

<div id="map"></div>

<div class="map-sidebar">

<div class="workshop-list"></div>

</div>

</div>

</form>

</section>

<!-- Passo 2: Recomendações -->

<section class="service-step" id="step2">

<h2><span class="step-number">2</span> Produtos Recomendados</h2>

<p class="step-description">Selecione os produtos que deseja incluir no serviço:</p>

<div class="recommendation-container">

<div class="product-card" id="recommended-oil">

<h3>Óleo Recomendado</h3>

<div class="product-selection">

<label class="checkbox-container">

<input type="checkbox" id="select-oil" checked>

<span class="checkmark"></span>

Incluir óleo no serviço

</label>

</div>

</div>

<div class="product-card" id="recommended-filter">

<h3>Filtro Recomendado</h3>

<div class="product-selection">

<label class="checkbox-container">

<input type="checkbox" id="select-filter" checked>

<span class="checkmark"></span>

Incluir filtro no serviço

</label>

</div>

</div>

</div>

<div class="selection-summary" id="selection-summary">

<h4>Resumo da Seleção:</h4>

<p id="selected-items">Nenhum produto selecionado</p>

</div>

<div class="step-buttons">

<button type="button" class="btn btn-secondary" onclick="goToStep(1)">Voltar</button>

<button type="button" class="btn btn-primary" onclick="goToStep(3)">Continuar</button>

</div>

</section>

<!-- Passo 3: Agendamento -->

<section class="service-step" id="step3">

<h2><span class="step-number">3</span> Agendamento</h2>

<div class="scheduling-container">

<div class="workshop-summary">

<h3>Oficina Selecionada</h3>

<div id="selected-workshop"></div>

</div>

<div class="schedule-form">

<div class="form-row">

<div class="form-group">

<label for="schedule-date">Data</label>

<input type="date" id="schedule-date" required>

</div>

<div class="form-group">

<label for="time">Horário:</label>

<select id="schedule-time" name="time" required>

<option value="" disabled selected>Selecione um horário</option>

<option value="08:00">08:00</option>

<option value="08:30">08:30</option>

<option value="09:00">09:00</option>

<option value="09:30">09:30</option>

<option value="10:00">10:00</option>

<option value="10:30">10:30</option>

<option value="11:00">11:00</option>

<option value="11:30">11:30</option>

<option value="12:00">12:00</option>

<option value="12:30">12:30</option>

<option value="13:00">13:00</option>

<option value="13:30">13:30</option>

<option value="14:00">14:00</option>

<option value="14:30">14:30</option>

<option value="15:00">15:00</option>

<option value="15:30">15:30</option>

<option value="16:00">16:00</option>

<option value="16:30">16:30</option>

<option value="17:00">17:00</option>

<option value="17:30">17:30</option>

<option value="18:00">18:00</option>

</select>

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="customer-name">Nome Completo</label>

<input type="text" id="customer-name" required>

</div>

<div class="form-group">

<label for="customer-cpf">CPF</label>

<input type="text" id="customer-cpf" placeholder="000.000.000-00" required

maxlength="14">

<div class="error-message"></div>

</div>

</div>

<div class="form-row">

<div class="form-group">

<label for="customer-phone">Telefone</label>

<input type="tel" id="customer-phone" placeholder="(00) 00000-0000" required>

</div>

<div class="form-group">

<label for="customer-email">E-mail</label>

<input type="email" id="customer-email" required>

</div>

</div>

<div class="step-buttons">

<button type="button" class="btn btn-secondary" onclick="goToStep(2)">Voltar</button>

<button type="button" class="btn btn-primary" onclick="goToStep(4)">Confirmar</button>

</div>

</div>

</div>

</section>

<!-- Passo 4: Confirmação -->

<section class="service-step" id="step4">

<div class="confirmation-container">

<div class="confirmation-icon">

<i class="fas fa-check-circle"></i>

</div>

<h2>Agendamento Confirmado!</h2>

<div class="confirmation-details">

<p><strong>Protocolo:</strong> #OIL20231115-004</p>

<p><strong>Serviço:</strong> Troca de Óleo Sintético</p>

<p><strong>Data:</strong> 15/11/2023 às 14:00</p>

<p><strong>Local:</strong> OilSmart - Unidade Paulista</p>

<p><strong>Endereço:</strong> Av. Paulista, 1000 - São Paulo/SP</p>

</div>

<p class="confirmation-text">Um e-mail de confirmação foi enviado para você com todos os

detalhes.</p>

<button id="download-pdf" class="btn btn-secondary" style="margin-top: 10px;">

<i class="fas fa-file-download"></i> Baixar Comprovante (PDF)

</button>

<button class="btn btn-primary" onclick="window.location.href='index.html'">Voltar ao

Início</button>

</div>

</section>

</div>

</div>

</main>

<footer class="footer">

<div class="container">

<div class="footer-content">

<div class="footer-brand">

<img src="/img/fjkaemfkemif.png" alt="OilSmart" class="footer-logo">

<p>Soluções completas para seu veículo</p>

</div>

<div class="footer-links">

<h3>Links Úteis</h3>

<ul>

<li><a href="#">Política de Privacidade</a></li>

<li><a href="#">Termos de Serviço</a></li>

<li><a href="#">Perguntas Frequentes</a></li>

</ul>

</div>

<div class="footer-contact">

<h3>Contato</h3>

<p><i class="fas fa-phone"></i> (11) 4002-8922</p>

<p><i class="fas fa-envelope"></i> contato@OilSmart.com.br</p>

</div>

</div>

<div class="copyright">

<p>© 2025 OilSmart. Todos os direitos reservados.</p>

</div>

</div>

</footer>

<div id="loading-overlay">

<div class="loader"></div>

</div>

<div id="toast" class="toast"></div>

<style>

/\* Estilos para a verificação de login \*/

.login-verification-overlay {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(33, 63, 87, 0.95);

display: flex;

justify-content: center;

align-items: center;

z-index: 9999;

backdrop-filter: blur(8px);

animation: overlayFadeIn 0.4s ease-out;

}

.login-verification-container {

background: var(--card-bg);

padding: 2.5rem;

border-radius: var(--border-radius);

box-shadow: var(--card-shadow);

text-align: center;

max-width: 450px;

width: 90%;

animation: modalSlideIn 0.5s ease-out;

border: 1px solid rgba(180, 148, 52, 0.2);

}

.login-verification-icon {

font-size: 3.5rem;

color: var(--secondary-color);

margin-bottom: 1.5rem;

animation: iconPulse 2s infinite;

}

.login-verification-title {

color: var(--primary-color);

font-size: 1.5rem;

margin-bottom: 1rem;

font-weight: 600;

}

.login-verification-message {

color: var(--text-color);

margin-bottom: 2rem;

line-height: 1.6;

}

.login-verification-btn {

background: linear-gradient(135deg, var(--secondary-color), var(--accent-color));

color: white;

border: none;

padding: 12px 28px;

border-radius: 50px;

font-weight: 600;

font-size: 1rem;

cursor: pointer;

transition: var(--transition);

box-shadow: 0 4px 15px rgba(180, 148, 52, 0.3);

display: inline-flex;

align-items: center;

gap: 8px;

}

.login-verification-btn:hover {

transform: translateY(-2px);

box-shadow: 0 6px 20px rgba(180, 148, 52, 0.4);

}

/\* Animações \*/

@keyframes overlayFadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

@keyframes modalSlideIn {

from {

opacity: 0;

transform: translateY(-30px) scale(0.95);

}

to {

opacity: 1;

transform: translateY(0) scale(1);

}

}

@keyframes iconPulse {

0% { transform: scale(1); }

50% { transform: scale(1.1); }

100% { transform: scale(1); }

}

/\* Responsividade \*/

@media (max-width: 768px) {

.login-verification-container {

padding: 2rem 1.5rem;

}

.login-verification-icon {

font-size: 3rem;

}

.login-verification-title {

font-size: 1.3rem;

}

}

@media (max-width: 480px) {

.login-verification-container {

padding: 1.5rem;

}

.login-verification-icon {

font-size: 2.5rem;

}

.login-verification-title {

font-size: 1.2rem;

}

.login-verification-btn {

padding: 10px 20px;

font-size: 0.9rem;

}

}

</style>

<script>

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

// Verificar se o usuário está logado usando o mesmo método do auth.js

const token = localStorage.getItem('token');

const userData = localStorage.getItem('user');

if (!token || !userData) {

// Criar overlay de verificação

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

overlay.className = 'login-verification-overlay';

// Criar conteúdo da verificação

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

container.className = 'login-verification-container';

container.innerHTML = `

<div class="login-verification-icon">🔒</div>

<h2 class="login-verification-title">Acesso Restrito</h2>

<p class="login-verification-message">Você precisa fazer login para agendar um serviço!</p>

<button class="login-verification-btn" onclick="window.location.href='login.html'">

<i class="fas fa-sign-in-alt"></i>

Fazer Login

</button>

`;

overlay.appendChild(container);

document.body.appendChild(overlay);

} else {

// Usuário está logado - preencher dados

const user = JSON.parse(userData);

const customerName = document.getElementById('customer-name');

const customerEmail = document.getElementById('customer-email');

if (customerName) customerName.value = user.nome || '';

if (customerEmail) customerEmail.value = user.email || '';

}

});

</script>

<script>

// Correção do menu hamburguer para a página de serviços

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

// Elementos do menu hamburguer

const hamburgerMenu = document.getElementById('hamburger-menu');

const mobileNav = document.getElementById('mobile-nav');

const body = document.body;

// Verifica se os elementos existem na página

if (hamburgerMenu && mobileNav) {

// Função para abrir/fechar o menu

function toggleMenu() {

const isMenuOpen = mobileNav.style.display === 'block';

mobileNav.style.display = isMenuOpen ? 'none' : 'block';

hamburgerMenu.classList.toggle('active', !isMenuOpen);

body.style.overflow = isMenuOpen ? 'auto' : 'hidden';

// Animações dos ícones do hamburguer

const hamburgerLines = hamburgerMenu.querySelectorAll('.hamburger-line');

if (!isMenuOpen) {

hamburgerLines[0].style.transform = 'rotate(45deg) translate(5px, 5px)';

hamburgerLines[1].style.opacity = '0';

hamburgerLines[2].style.transform = 'rotate(-45deg) translate(7px, -6px)';

} else {

hamburgerLines[0].style.transform = 'none';

hamburgerLines[1].style.opacity = '1';

hamburgerLines[2].style.transform = 'none';

}

}

// Adiciona o evento de clique ao hamburguer

hamburgerMenu.addEventListener('click', function (e) {

e.stopPropagation();

toggleMenu();

});

// Fecha o menu ao clicar fora dele

document.addEventListener('click', function (e) {

if (mobileNav.style.display === 'block' &&

!mobileNav.contains(e.target) &&

e.target !== hamburgerMenu) {

toggleMenu();

}

});

// Fecha o menu ao clicar em um link

const navLinks = mobileNav.querySelectorAll('a');

navLinks.forEach(link => {

link.addEventListener('click', function () {

toggleMenu();

});

});

// Ajusta o menu ao redimensionar a janela

window.addEventListener('resize', function () {

if (window.innerWidth > 992) {

mobileNav.style.display = 'none';

body.style.overflow = 'auto';

hamburgerMenu.classList.remove('active');

// Reseta as animações do hamburguer

const hamburgerLines = hamburgerMenu.querySelectorAll('.hamburger-line');

hamburgerLines[0].style.transform = 'none';

hamburgerLines[1].style.opacity = '1';

hamburgerLines[2].style.transform = 'none';

}

});

}

});

</script>

<style>

/\* Estilos para o menu hamburguer \*/

.hamburger-menu {

display: none;

flex-direction: column;

justify-content: space-around;

width: 30px;

height: 25px;

background: transparent;

border: none;

cursor: pointer;

padding: 0;

z-index: 1001;

}

.hamburger-line {

width: 100%;

height: 3px;

background: var(--f8f9fa);

border-radius: 2px;

transition: all 0.3s ease;

}

/\* Esconde o menu mobile por padrão \*/

#mobile-nav {

display: none;

position: fixed;

top: 80px;

left: 0;

width: 100%;

height: calc(100vh - 80px);

background: var(--light-color);

z-index: 1000;

overflow-y: auto;

padding: 20px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

}

/\* Mostra o menu hamburguer apenas em mobile \*/

@media (max-width: 992px) {

.hamburger-menu {

display: flex;

}

#mobile-nav.active {

display: block;

}

/\* Estilos para os links do menu mobile \*/

#mobile-nav .nav\_\_list {

flex-direction: column;

gap: 20px;

}

#mobile-nav .nav\_\_list a {

color: var(--text-color);

font-size: 1.2rem;

padding: 12px 0;

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

display: block;

}

#mobile-nav .nav\_\_list a:last-child {

border-bottom: none;

}

}

</style>

<script>

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

const downloadBtn = document.getElementById("download-pdf");

const confirmationSection = document.querySelector("#step4 .confirmation-container");

if (downloadBtn && confirmationSection) {

downloadBtn.addEventListener("click", async function () {

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

if (toast) {

toast.innerText = "Gerando PDF...";

toast.classList.add("show");

setTimeout(() => toast.classList.remove("show"), 2000);

}

html2canvas(confirmationSection, {

scale: 2,

backgroundColor: "#ffffff",

onclone: function (clonedDoc) {

const clonedSection = clonedDoc.querySelector(".confirmation-container");

if (clonedSection) {

clonedSection.style.width = "100%";

clonedSection.style.padding = "20px";

}

}

}).then(canvas => {

const imgData = canvas.toDataURL("image/png");

const { jsPDF } = window.jspdf;

const pdf = new jsPDF({

orientation: "portrait",

unit: "mm",

format: "a4"

});

const pageWidth = pdf.internal.pageSize.getWidth();

const pageHeight = pdf.internal.pageSize.getHeight();

const imgWidth = pageWidth - 20; // margem lateral

const imgHeight = (canvas.height \* imgWidth) / canvas.width;

let position = 10; // margem topo

let heightLeft = imgHeight;

// primeira página

pdf.addImage(imgData, "PNG", 10, position, imgWidth, imgHeight);

heightLeft -= pageHeight - 20;

// adiciona páginas extras se necessário

while (heightLeft > 0) {

pdf.addPage();

position = 10;

pdf.addImage(

imgData,

"PNG",

10,

position - (imgHeight - heightLeft),

imgWidth,

imgHeight

);

heightLeft -= pageHeight - 20;

}

// rodapé com data

const currentDate = new Date().toLocaleString('pt-BR');

pdf.setFontSize(10);

pdf.text(`Comprovante gerado em: ${currentDate}`, 10, pageHeight - 10);

pdf.save("Comprovante-Agendamento-OilSmart.pdf");

});

});

}

});

</script>

<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>

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

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

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

</body>

</html>

# Arquivo: public/html/sobre.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>Sobre Nós - OilSmart</title>

<meta name="description" content="Conheça a OilSmart, nossa história, missão e valores. A melhor escolha para serviços automotivos especializados.">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<link rel="stylesheet" href="/css/main.css">

<link rel="icon" href="/img/tentativa4.png" type="image/x-icon"></head>

<link rel="stylesheet" href="/css/auth.css">

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

</head>

<body>

<header class="header">

<div class="container">

<div class="navbar">

<button class="hamburger" id="hamburger">

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

</button>

<div class="logo">

<a href="index.html"><img src="/img/oilamarelo.png" alt="OilSmart"></a>

</div>

<nav class="nav" id="nav">

<ul class="nav\_\_list">

<li><a href="/html/index.html">Início</a></li>

<li><a href="/html/servicos.html">Serviços</a></li>

<li><a href="/html/agenda.html">Agenda</a></li>

<li><a href="/html/sobre.html" class="active">Sobre</a></li>

<li><a href="/html/contato.html">Contato</a></li>

</ul>

</nav>

<div class="action-icons">

<div class="user-dropdown" id="user-dropdown">

<div class="user-icon" id="login-btn">

<i class="fas fa-user"></i>

<span class="login-status" id="login-status">Login</span>

</div>

<div class="dropdown-menu" id="dropdown-menu">

<div class="user-info">

<strong id="user-display-name">Usuário</strong>

<span id="user-email">email@exemplo.com</span>

</div>

<div class="dropdown-divider"></div>

<a href="/html/agenda.html" class="dropdown-item">

<i class="fas fa-calendar-alt"></i> Meus Agendamentos

</a>

<div class="dropdown-divider"></div>

<a href="/html/index.html" id="logout-btn" class="dropdown-item">

<i class="fas fa-sign-out-alt"></i> Sair

</a>

</div>

</div>

</div>

</div>

</div>

</header>

<main class="main-content">

<section class="about-banner">

<div class="container">

<h1>Conheça a OilSmart</h1>

<p>Sua solução completa em serviços automotivos</p>

</div>

</section>

<div class="hero-divider"></div>

<div class="container about-container">

<section class="about-section">

<div class="about-content">

<h2 class="section-title">Nossa História</h2>

<p>A OilSmart nasceu da necessidade de transformar a rotina de motoristas e oficinas por todo o Brasil. Em um cenário onde a manutenção veicular ainda é cercada por incertezas — como escolher o óleo correto, encontrar oficinas confiáveis e evitar longas esperas — nós vimos uma oportunidade de inovar com tecnologia e inteligência.</p>

<p>Formado por um grupo de estudantes apaixonados por soluções práticas, nosso time desenvolveu uma plataforma digital inteligente que simplifica o processo de troca de óleo. A partir de um simples cadastro, o motorista informa o modelo e ano do seu veículo, e o sistema recomenda automaticamente o óleo ideal. Além disso, localizamos as oficinas mais próximas e permitimos o agendamento online, tudo em poucos cliques.</p>

<div class="milestones">

<div class="milestone">

<div class="number">2025</div>

<div class="description">Iniciamos nossa jornada conectando tecnologia e oficinas, com a criação do OilSmart</div>

</div>

</div>

</div>

</section>

<section class="mv-section">

<div class="mv-card">

<i class="fas fa-bullseye"></i>

<h3>Missão</h3>

<p>Oferecer serviços automotivos de excelência com transparência, agilidade e conveniência digital, proporcionando tranquilidade e satisfação aos nossos clientes.</p>

</div>

<div class="mv-card">

<i class="fas fa-eye"></i>

<h3>Visão</h3>

<p>Ser reconhecida como a melhor rede de serviços automotivos do Brasil, ampliando nossa presença nacional com inovação constante e atendimento humanizado.</p>

</div>

<div class="mv-card">

<i class="fas fa-heart"></i>

<h3>Valores</h3>

<ul>

<li>Transparência e honestidade</li>

<li>Qualidade acima de tudo</li>

<li>Inovação e tecnologia</li>

<li>Sustentabilidade</li>

<li>Paixão por automóveis</li>

</ul>

</div>

</section>

<section class="diferenciais-section">

<h2 class="section-title">Nossos Diferenciais</h2>

<div class="diferenciais-grid">

<div class="diferencial-card">

<i class="fas fa-clock"></i>

<h3>Agendamento Online</h3>

<p>Agende seu serviço em minutos pelo site ou app, sem ligações ou espera.</p>

</div>

<div class="diferencial-card">

<i class="fas fa-certificate"></i>

<h3>Produtos Premium</h3>

<p>Trabalhamos apenas com óleos e filtros das melhores marcas do mercado.</p>

</div>

<div class="diferencial-card">

<i class="fas fa-cogs" aria-hidden="true"></i>

<h3>Diagnóstico Inteligente</h3>

<p>Nosso sistema analisa o modelo do seu carro para recomendar o óleo perfeito e os melhores serviços.</p>

</div>

<div class="diferencial-card">

<i class="fas fa-history"></i>

<h3>Histórico Digital</h3>

<p>Acesse todo o histórico de serviços do seu veículo em qualquer lugar.</p>

</div>

</div>

</section>

<section class="testimonials-section">

<h2 class="section-title">O que nossos clientes dizem</h2>

<div class="testimonials-container">

<div class="testimonial">

<div class="quote">

<i class="fas fa-quote-left"></i>

<p>Atendimento rápido e profissional. Levei meu carro para troca de óleo e saí em menos de 30 minutos. Recomendo!</p>

</div>

<div class="client">

<img src="/img/client1.jpg" alt="Marcos Andrade">

<div class="client-info">

<h4>Marcos Andrade</h4>

<p>Cliente desde 2025</p>

</div>

</div>

</div>

<div class="testimonial">

<div class="quote">

<i class="fas fa-quote-left"></i>

<p>Adorei a praticidade de agendar pelo app e acompanhar todo o serviço pelo celular. Transparência total!</p>

</div>

<div class="client">

<img src="/img/motta.jpg" alt="Fernanda Lima">

<div class="client-info">

<h4>Flávio Mota</h4>

<p>Cliente desde 2025</p>

</div>

</div>

</div>

<div class="testimonial">

<div class="quote">

<i class="fas fa-quote-left"></i>

<p>Meu carro nunca rodou tão bem depois que comecei a fazer as revisões na OilSmart. Equipe nota 10!</p>

</div>

<div class="client">

<img src="/img/client3.jpg" alt="Roberto Alves">

<div class="client-info">

<h4>Roberto Alves</h4>

<p>Cliente desde 2025</p>

</div>

</div>

</div>

</div>

</section>

</div>

</main>

<footer class="footer">

<div class="container">

<div class="footer-content">

<div class="footer-brand">

<img src="/img/fjkaemfkemif.png" alt="OilSmart" class="footer-logo">

<p>Soluções completas para seu veículo</p>

</div>

<div class="footer-links">

<h3>Links Úteis</h3>

<ul>

<li><a href="/html/politicas.html">Política de Privacidade</a></li>

<li><a href="/html/termos.html">Termos de Serviço</a></li>

<li><a href="/html/faq.html">Perguntas Frequentes</a></li>

</ul>

</div>

<div class="footer-contact">

<h3>Contato</h3>

<p><i class="fas fa-phone"></i> (11) 4002-8922</p>

<p><i class="fas fa-envelope"></i> contato@OilSmart.com.br</p>

</div>

</div>

<div class="copyright">

<p>© 2025 OilSmart. Todos os direitos reservados.</p>

</div>

</div>

</footer>

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

</body>

</html>

# Arquivo: public/html/termos.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

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

<title>Termos de Serviço - OilSmart</title>

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<link rel="stylesheet" href="../css/main.css">

</head>

<body>

<!-- Header -->

<header class="header">

<div class="container">

<nav class="navbar">

<a href="index.html" class="logo">

<img src="../img/LogoIOil.png" alt="OilSmart">

</a>

<div class="nav" id="nav">

<ul class="nav\_\_list">

<li><a href="index.html">Início</a></li>

<li><a href="servicos.html">Serviços</a></li>

<li><a href="sobre.html">Sobre</a></li>

<li><a href="agenda.html">Agenda</a></li>

<li><a href="contatos.html">Contatos</a></li>

<li><a href="faq.html">FAQ</a></li>

</ul>

</div>

<div class="action-icons">

<div class="user-dropdown">

<div class="user-ico">

<i class="fas fa-user-circle"></i>

<span class="login-status">Entrar</span>

</div>

<div class="dropdown-menu">

<div class="user-info">

<strong>Visitante</strong>

<span>Entre ou cadastre-se</span>

</div>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item" data-modal="login">

<i class="fas fa-sign-in-alt"></i>

Fazer Login

</a>

<a href="#" class="dropdown-item" data-modal="register">

<i class="fas fa-user-plus"></i>

Criar Conta

</a>

</div>

</div>

<button class="hamburger" id="hamburger">

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

<span class="hamburger-line"></span>

</button>

</div>

</nav>

</div>

</header>

<!-- Main Content -->

<main class="main-content">

<div class="container">

<h1 class="page-title">Termos de Serviço</h1>

<p class="page-subtitle">Conheça os termos que regem o uso de nossa plataforma e serviços.</p>

<div class="terms-content" style="background: var(--card-bg); padding: 40px; border-radius: var(--border-radius); box-shadow: var(--card-shadow); margin-top: 30px;">

<p>Estes Termos de Serviço regulam o uso da plataforma OilSmart. Ao utilizar nossos serviços, você concorda com estes termos e com nossa <a href="politicas.html" style="color: var(--secondary-color);">Política de Privacidade</a>. Leia atentamente antes de prosseguir.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">1. Aceitação dos Termos</h2>

<p>Ao acessar ou usar a plataforma OilSmart, você concorda em ficar vinculado a estes Termos de Serviço e a nossa Política de Privacidade. Se você não concordar com qualquer parte destes termos, não deverá utilizar nossos serviços.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">2. Definições</h2>

<p>Para os fins destes Termos:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li><strong>"Plataforma"</strong> refere-se ao website, aplicativo e serviços da OilSmart</li>

<li><strong>"Usuário"</strong> é qualquer pessoa que acessa ou utiliza a Plataforma</li>

<li><strong>"Oficina Parceira"</strong> são estabelecimentos terceirizados que prestam serviços através da Plataforma</li>

<li><strong>"Serviços"</strong> são os serviços de intermediação e agendamento oferecidos pela OilSmart</li>

</ul>

<h2 style="color: var(--secondary-color); margin-top: 30px;">3. Cadastro e Conta do Usuário</h2>

<p>Para utilizar certos recursos da Plataforma, você precisará criar uma conta. Você concorda em:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li>Fornecer informações verdadeiras, precisas, atualizadas e completas</li>

<li>Manter a confidencialidade de sua senha e informações de conta</li>

<li>Ser responsável por toda atividade que ocorra em sua conta</li>

<li>Notificar-nos imediatamente sobre qualquer uso não autorizado de sua conta</li>

</ul>

<h2 style="color: var(--secondary-color); margin-top: 30px;">4. Serviços Prestados</h2>

<p>A OilSmart atua como intermediária entre usuários e oficinas parceiras, oferecendo:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li>Recomendações de óleo com base nas informações do veículo</li>

<li>Agendamento de serviços com oficinas parceiras</li>

<li>Informações sobre oficinas próximas e disponibilidade</li>

<li>Histórico de serviços realizados</li>

</ul>

<p><strong>Importante:</strong> A OilSmart não executa diretamente os serviços de troca de óleo ou manutenção veicular. A execução dos serviços é de responsabilidade exclusiva das oficinas parceiras.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">5. Responsabilidades do Usuário</h2>

<p>Você concorda em:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li>Fornecer informações precisas sobre seu veículo</li>

<li>Comparecer aos agendamentos no horário marcado</li>

<li>Respeitar as políticas de cancelamento das oficinas</li>

<li>Avaliar os serviços prestados de forma justa e construtiva</li>

<li>Não utilizar a Plataforma para fins ilegais ou não autorizados</li>

</ul>

<h2 style="color: var(--secondary-color); margin-top: 30px;">6. Pagamentos</h2>

<p>Os pagamentos pelos serviços agendados são processados através de nossos parceiros de pagamento. As oficinas parceiras são responsáveis por definir os preços dos serviços. A OilSmart pode cobrar uma taxa de intermediação, claramente informada durante o processo de agendamento.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">7. Cancelamentos e Reagendamentos</h2>

<p>Cancelamentos e reagendamentos estão sujeitos às políticas específicas de cada oficina parceira. Em geral:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li>Cancelamentos com até 1 hora de antecedência não geram custos</li>

<li>Cancelamentos em cima da hora podem estar sujeitos a taxas conforme política da oficina</li>

<li>Reagendamentos podem ser feitos através da Plataforma, sujeito à disponibilidade</li>

</ul>

<h2 style="color: var(--secondary-color); margin-top: 30px;">8. Limitação de Responsabilidade</h2>

<p>A OilSmart não será responsável por:</p>

<ul style="padding-left: 20px; margin-bottom: 20px;">

<li>Danos ou prejuízos resultantes de serviços prestados por oficinas parceiras</li>

<li>Informações incorretas fornecidas pelo usuário</li>

<li>Problemas técnicos ou de comunicação fora de nosso controle</li>

<li>Cancelamentos ou alterações feitas pelas oficinas parceiras</li>

</ul>

<p>Nosso papel é limitado à intermediação entre usuários e oficinas. Qualquer reclamação sobre serviços prestados deve ser direcionada diretamente à oficina responsável.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">9. Propriedade Intelectual</h2>

<p>Todos os direitos de propriedade intelectual relacionados à Plataforma, incluindo software, design, logotipos, textos e gráficos, são de propriedade da OilSmart ou de nossos licenciadores. Você concorda em não copiar, modificar ou distribuir qualquer parte da Plataforma sem autorização prévia por escrito.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">10. Modificações nos Termos</h2>

<p>Podemos modificar estes Termos a qualquer tempo. Alterações significativas serão notificadas por e-mail ou através de avisos na Plataforma. O uso continuado dos serviços após tais modificações constitui sua aceitação dos novos Termos.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">11. Rescisão</h2>

<p>Podemos suspender ou encerrar seu acesso à Plataforma a qualquer momento, sem aviso prévio, se violar estes Termos. Você também pode encerrar sua conta a qualquer momento através das configurações da Plataforma ou entrando em contato conosco.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">12. Lei Aplicável</h2>

<p>Estes Termos serão regidos e interpretados de acordo com as leis do Brasil. Qualquer disputa relacionada a estes Termos será submetida à jurisdição exclusiva dos tribunais da comarca de São Paulo/SP.</p>

<h2 style="color: var(--secondary-color); margin-top: 30px;">13. Contato</h2>

<p>Para questões relacionadas a estes Termos de Serviço, entre em contato conosco:</p>

<p><strong>E-mail:</strong> contato@oilsmart.com.br<br>

<strong>Endereço:</strong> São Paulo, Brasil</p>

<p class="last-updated" style="margin-top: 40px; font-size: 0.9rem; color: var(--text-light); text-align: right;">Última atualização: Maio de 2025</p>

</div>

</div>

</main>

<!-- Footer -->

<footer class="footer">

<div class="container">

<div class="footer-content">

<div class="footer-brand">

<img src="../img/LogoIOil.png" alt="OilSmart">

<p>Conectando você às melhores oficinas e produtos para o cuidado do seu veículo.</p>

</div>

<div class="footer-links">

<h3>Links Rápidos</h3>

<ul>

<li><a href="index.html">Início</a></li>

<li><a href="servicos.html">Serviços</a></li>

<li><a href="sobre.html">Sobre Nós</a></li>

<li><a href="agenda.html">Minha Agenda</a></li>

<li><a href="contatos.html">Contato</a></li>

</ul>

</div>

<div class="footer-links">

<h3>Suporte</h3>

<ul>

<li><a href="faq.html">Perguntas Frequentes</a></li>

<li><a href="politicas.html">Política de Privacidade</a></li>

<li><a href="termos.html">Termos de Serviço</a></li>

<li><a href="contatos.html">Fale Conosco</a></li>

</ul>

</div>

<div class="footer-contact">

<h3>Contato</h3>

<p><i class="fas fa-envelope"></i> contato@oilsmart.com.br</p>

<p><i class="fas fa-phone"></i> (11) 3456-7890</p>

<p><i class="fas fa-map-marker-alt"></i> São Paulo, Brasil</p>

</div>

</div>

<div class="copyright">

<p>&copy; 2025 OilSmart. Todos os direitos reservados.</p>

</div>

</div>

</footer>

<!-- Login Modal -->

<div class="modal" id="login-modal">

<div class="modal-content">

<button class="close-modal">&times;</button>

<div class="modal-header">

<h2>Fazer Login</h2>

<p>Entre na sua conta para acessar todos os recursos</p>

</div>

<form class="login-form">

<div class="form-group">

<label for="login-email">E-mail</label>

<input type="email" id="login-email" placeholder="Seu e-mail" required>

</div>

<div class="form-group">

<label for="login-password">Senha</label>

<input type="password" id="login-password" placeholder="Sua senha" required>

</div>

<button type="submit" class="btn btn-primary btn-block">Entrar</button>

</form>

<div class="divider">ou</div>

<div class="social-login">

<button class="social-btn google">

<i class="fab fa-google"></i>

Entrar com Google

</button>

</div>

<div class="register-link">

Não tem uma conta? <a href="#" data-modal="register">Cadastre-se</a>

</div>

</div>

</div>

<!-- Register Modal -->

<div class="modal" id="register-modal">

<div class="modal-content">

<button class="close-modal">&times;</button>

<div class="modal-header">

<h2>Criar Conta</h2>

<p>Junte-se à nossa comunidade</p>

</div>

<form class="login-form">

<div class="form-group">

<label for="register-name">Nome Completo</label>

<input type="text" id="register-name" placeholder="Seu nome completo" required>

</div>

<div class="form-group">

<label for="register-email">E-mail</label>

<input type="email" id="register-email" placeholder="Seu e-mail" required>

</div>

<div class="form-group">

<label for="register-password">Senha</label>

<input type="password" id="register-password" placeholder="Crie uma senha" required>

</div>

<div class="form-group">

<label for="register-confirm-password">Confirmar Senha</label>

<input type="password" id="register-confirm-password" placeholder="Repita sua senha" required>

</div>

<button type="submit" class="btn btn-primary btn-block">Criar Conta</button>

</form>

<div class="divider">ou</div>

<div class="social-login">

<button class="social-btn google">

<i class="fab fa-google"></i>

Cadastrar com Google

</button>

</div>

<div class="register-link">

Já tem uma conta? <a href="#" data-modal="login">Faça login</a>

</div>

</div>

</div>

<!-- Loading Overlay -->

<div id="loading-overlay">

<div class="loader"></div>

</div>

<!-- Toast Notification -->

<div class="toast" id="toast"></div>

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

<script>

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

// Initialize modals and mobile menu

initModals();

initMobileMenu();

// Scroll to top

window.scrollTo({ top: 0, behavior: 'smooth' });

});

function initModals() {

// Modal functionality would be implemented here

}

function initMobileMenu() {

// Mobile menu functionality would be implemented here

}

</script>

</body>

</html>

# Arquivo: public/js/adm.js

// OilSmart - Sistema Administrativo

// adm.js - Funcionalidades JavaScript com Backend Integrado

// Variáveis globais

let currentPage = 'dashboard';

let currentUser = null;

const API\_BASE = '/api';

// Inicialização

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

verificarAutenticacao();

atualizarDataHora();

setInterval(atualizarDataHora, 60000);

});

// Funções de autenticação

async function verificarAutenticacao() {

try {

const response = await fetch(`${API\_BASE}/usuario`, {

credentials: 'include'

});

if (response.ok) {

currentUser = await response.json();

mostrarDashboard();

} else {

mostrarLogin();

}

} catch (error) {

console.error('Erro ao verificar autenticação:', error);

mostrarLogin();

}

}

async function fazerLogin() {

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

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

const errorMsg = document.getElementById('loginError');

const btn = document.querySelector('#loginForm .auth-btn');

btn.disabled = true;

btn.innerHTML = '<div class="loading"></div> Entrando...';

try {

const response = await fetch(`${API\_BASE}/login`, {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

credentials: 'include',

body: JSON.stringify({ email, senha })

});

const data = await response.json();

if (response.ok) {

currentUser = data.user;

errorMsg.style.display = 'none';

mostrarDashboard();

} else {

errorMsg.textContent = data.error || 'Erro ao fazer login';

errorMsg.style.display = 'block';

}

} catch (error) {

errorMsg.textContent = 'Erro de conexão. Tente novamente.';

errorMsg.style.display = 'block';

} finally {

btn.disabled = false;

btn.innerHTML = 'Entrar no Sistema';

}

}

async function fazerCadastro() {

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

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

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

const confirmarSenha = document.getElementById('cadastroConfirmarSenha').value;

const errorMsg = document.getElementById('cadastroError');

const btn = document.querySelector('#cadastroForm .auth-btn');

// Validação básica

if (senha !== confirmarSenha) {

errorMsg.textContent = 'As senhas não coincidem';

errorMsg.style.display = 'block';

return;

}

btn.disabled = true;

btn.innerHTML = '<div class="loading"></div> Criando conta...';

try {

const response = await fetch(`${API\_BASE}/cadastro`, {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ nome, email, senha })

});

const data = await response.json();

if (response.ok) {

alert('Conta criada com sucesso! Faça login para continuar.');

voltarParaLogin();

} else {

errorMsg.textContent = data.error || 'Erro ao criar conta';

errorMsg.style.display = 'block';

}

} catch (error) {

errorMsg.textContent = 'Erro de conexão. Tente novamente.';

errorMsg.style.display = 'block';

} finally {

btn.disabled = false;

btn.innerHTML = 'Criar Conta';

}

}

async function fazerLogout() {

try {

await fetch(`${API\_BASE}/logout`, {

method: 'POST',

credentials: 'include'

});

} catch (error) {

console.error('Erro ao fazer logout:', error);

} finally {

currentUser = null;

mostrarLogin();

}

}

// Função para mostrar o formulário de cadastro

function mostrarCadastro() {

document.getElementById('authContainer').style.display = 'flex';

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

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

}

// Função para voltar ao login

function voltarParaLogin() {

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

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

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

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

}

function mostrarLogin() {

document.getElementById('authContainer').style.display = 'flex';

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

voltarParaLogin();

}

function mostrarDashboard() {

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

document.getElementById('adminContainer').style.display = 'flex';

// Atualizar informações do usuário na sidebar

if (currentUser) {

document.querySelector('.user-name').textContent = currentUser.nome;

document.querySelector('.user-role').textContent = currentUser.tipo === 'admin' ? 'Super Admin' : 'Usuário';

}

carregarDadosDashboard();

navegarPara('dashboard');

}

// Navegação

function navegarPara(pagina) {

// Esconder todas as seções

document.querySelectorAll('.content-section').forEach(section => {

section.classList.remove('active');

});

// Mostrar a seção selecionada

document.getElementById(pagina).classList.add('active');

// Atualizar o menu ativo

document.querySelectorAll('.nav-link').forEach(link => {

link.classList.remove('active');

});

document.querySelector(`[onclick="navegarPara('${pagina}')"]`).classList.add('active');

// Atualizar título da página

const titulos = {

'dashboard': 'Dashboard',

'produtos': 'Produtos',

'vendas': 'Vendas',

'funcionarios': 'Funcionários',

'relatorios': 'Relatórios'

};

document.getElementById('pageTitle').textContent = titulos[pagina] || 'Dashboard';

// Carregar dados específicos da página

if (pagina === 'produtos') {

carregarProdutos();

} else if (pagina === 'vendas') {

carregarVendas();

} else if (pagina === 'funcionarios') {

carregarFuncionarios();

}

currentPage = pagina;

}

// Dashboard

async function carregarDadosDashboard() {

try {

const response = await fetch(`${API\_BASE}/dashboard/metricas`, {

credentials: 'include'

});

if (response.ok) {

const metricas = await response.json();

atualizarMetricas(metricas);

inicializarGraficos(metricas);

carregarAtividadesRecentes(metricas);

} else {

console.error('Erro ao carregar métricas do dashboard');

}

} catch (error) {

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

}

}

function atualizarMetricas(metricas) {

document.getElementById('totalVendas').textContent = `R$ ${parseFloat(metricas.totalVendas.total).toFixed(2)}`;

document.getElementById('totalProdutos').textContent = metricas.totalProdutos.total;

document.getElementById('totalFuncionarios').textContent = metricas.totalFuncionarios.total;

document.getElementById('vendasPendentes').textContent = metricas.vendasPendentes.total;

}

function inicializarGraficos(metricas) {

const ctxVendas = document.getElementById('vendasChart').getContext('2d');

const ctxCategorias = document.getElementById('categoriasChart').getContext('2d');

// Gráfico de vendas mensais

new Chart(ctxVendas, {

type: 'line',

data: {

labels: metricas.vendasMes ? Object.keys(metricas.vendasMes).reverse() : ['Jan', 'Fev', 'Mar', 'Abr', 'Mai', 'Jun'],

datasets: [{

label: 'Vendas Mensais (R$)',

data: metricas.vendasMes ? Object.values(metricas.vendasMes).map(m => m.total).reverse() : [12000, 19000, 15000, 18000, 22000, 21000],

borderColor: '#b49434',

backgroundColor: 'rgba(180, 148, 52, 0.1)',

tension: 0.4,

fill: true

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

display: false

}

}

}

});

// Gráfico de categorias

new Chart(ctxCategorias, {

type: 'doughnut',

data: {

labels: metricas.produtosCategoria ? metricas.produtosCategoria.map(p => p.categoria) : ['Óleos', 'Filtros', 'Aditivos', 'Acessórios'],

datasets: [{

data: metricas.produtosCategoria ? metricas.produtosCategoria.map(p => p.quantidade) : [45, 25, 15, 15],

backgroundColor: ['#b49434', '#213f57', '#d4b15f', '#384d5e'],

borderWidth: 0

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

position: 'bottom'

}

}

}

});

}

function carregarAtividadesRecentes(metricas) {

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

atividadesList.innerHTML = '';

if (metricas.vendasRecentes && metricas.vendasRecentes.length > 0) {

metricas.vendasRecentes.forEach(venda => {

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

atividadeItem.className = 'activity-item';

atividadeItem.innerHTML = `

<div class="activity-icon">

<i class="fas fa-shopping-cart"></i>

</div>

<div class="activity-content">

<div class="activity-title">Nova venda #${venda.id}</div>

<div class="activity-time">${formatarData(venda.data\_venda)} - R$ ${venda.total.toFixed(2)}</div>

</div>

`;

atividadesList.appendChild(atividadeItem);

});

}

}

// Produtos

async function carregarProdutos() {

try {

const response = await fetch(`${API\_BASE}/produtos`, {

credentials: 'include'

});

if (response.ok) {

const produtos = await response.json();

exibirProdutos(produtos);

} else {

console.error('Erro ao carregar produtos');

}

} catch (error) {

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

}

}

function exibirProdutos(produtos) {

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

produtosGrid.innerHTML = '';

if (produtos.length === 0) {

produtosGrid.innerHTML = '<p class="text-center">Nenhum produto cadastrado.</p>';

return;

}

produtos.forEach(produto => {

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

produtoCard.className = 'produto-card';

produtoCard.innerHTML = `

<div class="produto-imagem">

${produto.imagem ? `<img src="${produto.imagem}" alt="${produto.nome}">` :

`<i class="fas fa-box-open"></i>`}

</div>

<div class="produto-info">

<h3 class="produto-nome">${produto.nome}</h3>

<p class="produto-detalhes">${produto.categoria} • ${produto.marca}</p>

<p class="produto-detalhes">Estoque: ${produto.estoque}</p>

<p class="produto-preco">R$ ${produto.preco.toFixed(2)}</p>

<div class="produto-acoes">

<button class="btn btn-outline btn-sm" onclick="editarProduto(${produto.id})">

<i class="fas fa-edit"></i> Editar

</button>

<button class="btn btn-danger btn-sm" onclick="excluirProduto(${produto.id})">

<i class="fas fa-trash"></i> Excluir

</button>

</div>

</div>

`;

produtosGrid.appendChild(produtoCard);

});

}

function abrirModalProduto() {

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

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

document.getElementById('produtoImagemPreview').innerHTML = '<div class="placeholder">Imagem do produto</div>';

document.getElementById('modalProduto').classList.add('active');

}

async function salvarProduto() {

const formData = new FormData();

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

const imagemInput = document.getElementById('produtoImagem');

formData.append('nome', document.getElementById('produtoNome').value);

formData.append('descricao', document.getElementById('produtoDescricao').value);

formData.append('categoria', document.getElementById('produtoCategoria').value);

formData.append('marca', document.getElementById('produtoMarca').value);

formData.append('preco', document.getElementById('produtoPreco').value);

formData.append('estoque', document.getElementById('produtoEstoque').value);

if (imagemInput.files[0]) {

formData.append('imagem', imagemInput.files[0]);

}

try {

const url = id ? `${API\_BASE}/produtos/${id}` : `${API\_BASE}/produtos`;

const method = id ? 'PUT' : 'POST';

const response = await fetch(url, {

method: method,

credentials: 'include',

body: formData

});

if (response.ok) {

fecharModal('modalProduto');

carregarProdutos();

carregarDadosDashboard();

mostrarMensagem('Produto salvo com sucesso!');

} else {

const error = await response.json();

mostrarMensagem(error.error || 'Erro ao salvar produto', 'error');

}

} catch (error) {

mostrarMensagem('Erro de conexão. Tente novamente.', 'error');

}

}

async function editarProduto(id) {

try {

const response = await fetch(`${API\_BASE}/produtos/${id}`, {

credentials: 'include'

});

if (response.ok) {

const produto = await response.json();

document.getElementById('produtoId').value = produto.id;

document.getElementById('produtoNome').value = produto.nome;

document.getElementById('produtoDescricao').value = produto.descricao || '';

document.getElementById('produtoCategoria').value = produto.categoria;

document.getElementById('produtoMarca').value = produto.marca;

document.getElementById('produtoPreco').value = produto.preco;

document.getElementById('produtoEstoque').value = produto.estoque;

const preview = document.getElementById('produtoImagemPreview');

if (produto.imagem) {

preview.innerHTML = `<img src="${produto.imagem}" alt="${produto.nome}">`;

} else {

preview.innerHTML = '<div class="placeholder">Imagem do produto</div>';

}

document.getElementById('modalProduto').classList.add('active');

}

} catch (error) {

mostrarMensagem('Erro ao carregar produto', 'error');

}

}

async function excluirProduto(id) {

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

try {

const response = await fetch(`${API\_BASE}/produtos/${id}`, {

method: 'DELETE',

credentials: 'include'

});

if (response.ok) {

carregarProdutos();

carregarDadosDashboard();

mostrarMensagem('Produto excluído com sucesso!');

} else {

mostrarMensagem('Erro ao excluir produto', 'error');

}

} catch (error) {

mostrarMensagem('Erro de conexão. Tente novamente.', 'error');

}

}

}

// Funcionários

async function carregarFuncionarios() {

try {

const response = await fetch(`${API\_BASE}/funcionarios`, {

credentials: 'include'

});

if (response.ok) {

const funcionarios = await response.json();

exibirFuncionarios(funcionarios);

} else {

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

}

} catch (error) {

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

}

}

function exibirFuncionarios(funcionarios) {

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

funcionariosGrid.innerHTML = '';

if (funcionarios.length === 0) {

funcionariosGrid.innerHTML = '<p class="text-center">Nenhum funcionário cadastrado.</p>';

return;

}

funcionarios.forEach(funcionario => {

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

funcionarioCard.className = 'funcionario-card';

funcionarioCard.innerHTML = `

<div class="funcionario-avatar">

<i class="fas fa-user"></i>

</div>

<div class="funcionario-info">

<h3 class="funcionario-nome">${funcionario.nome}</h3>

<p class="funcionario-detalhes">${funcionario.cargo}</p>

<p class="funcionario-detalhes">${funcionario.email}</p>

<p class="funcionario-detalhes">${funcionario.telefone}</p>

<p class="funcionario-detalhes">

<span class="status-badge status-${funcionario.status}">${funcionario.status}</span>

</p>

<div class="funcionario-acoes">

<button class="btn btn-outline btn-sm" onclick="editarFuncionario(${funcionario.id})">

<i class="fas fa-edit"></i> Editar

</button>

<button class="btn btn-danger btn-sm" onclick="excluirFuncionario(${funcionario.id})">

<i class="fas fa-trash"></i> Excluir

</button>

</div>

</div>

`;

funcionariosGrid.appendChild(funcionarioCard);

});

}

function abrirModalFuncionario() {

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

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

document.getElementById('modalFuncionario').classList.add('active');

}

async function salvarFuncionario() {

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

const funcionario = {

nome: document.getElementById('funcionarioNome').value,

email: document.getElementById('funcionarioEmail').value,

telefone: document.getElementById('funcionarioTelefone').value,

cargo: document.getElementById('funcionarioCargo').value,

departamento: document.getElementById('funcionarioDepartamento').value,

status: document.getElementById('funcionarioStatus').value

};

try {

const url = id ? `${API\_BASE}/funcionarios/${id}` : `${API\_BASE}/funcionarios`;

const method = id ? 'PUT' : 'POST';

const response = await fetch(url, {

method: method,

headers: {

'Content-Type': 'application/json'

},

credentials: 'include',

body: JSON.stringify(funcionario)

});

if (response.ok) {

fecharModal('modalFuncionario');

carregarFuncionarios();

carregarDadosDashboard();

mostrarMensagem('Funcionário salvo com sucesso!');

} else {

const error = await response.json();

mostrarMensagem(error.error || 'Erro ao salvar funcionário', 'error');

}

} catch (error) {

mostrarMensagem('Erro de conexão. Tente novamente.', 'error');

}

}

async function editarFuncionario(id) {

try {

const response = await fetch(`${API\_BASE}/funcionarios/${id}`, {

credentials: 'include'

});

if (response.ok) {

const funcionario = await response.json();

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

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

document.getElementById('funcionarioEmail').value = funcionario.email;

document.getElementById('funcionarioTelefone').value = funcionario.telefone;

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

document.getElementById('funcionarioDepartamento').value = funcionario.departamento;

document.getElementById('funcionarioStatus').value = funcionario.status;

document.getElementById('modalFuncionario').classList.add('active');

}

} catch (error) {

mostrarMensagem('Erro ao carregar funcionário', 'error');

}

}

async function excluirFuncionario(id) {

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

try {

const response = await fetch(`${API\_BASE}/funcionarios/${id}`, {

method: 'DELETE',

credentials: 'include'

});

if (response.ok) {

carregarFuncionarios();

carregarDadosDashboard();

mostrarMensagem('Funcionário excluído com sucesso!');

} else {

mostrarMensagem('Erro ao excluir funcionário', 'error');

}

} catch (error) {

mostrarMensagem('Erro de conexão. Tente novamente.', 'error');

}

}

}

// Vendas

async function carregarVendas() {

try {

const response = await fetch(`${API\_BASE}/vendas`, {

credentials: 'include'

});

if (response.ok) {

const vendas = await response.json();

exibirVendas(vendas);

atualizarResumoVendas(vendas);

} else {

console.error('Erro ao carregar vendas');

}

} catch (error) {

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

}

}

function exibirVendas(vendas) {

const vendasTable = document.getElementById('vendasTable').querySelector('tbody');

vendasTable.innerHTML = '';

if (vendas.length === 0) {

vendasTable.innerHTML = '<tr><td colspan="7" class="text-center">Nenhuma venda registrada.</td></tr>';

return;

}

vendas.forEach(venda => {

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

tr.innerHTML = `

<td>#${venda.id}</td>

<td>${formatarData(venda.data\_venda)}</td>

<td>${venda.cliente}</td>

<td>${venda.total\_itens} item(s)</td>

<td>R$ ${venda.total.toFixed(2)}</td>

<td><span class="status-badge status-${venda.status}">${venda.status}</span></td>

<td>

<button class="btn btn-outline btn-sm" onclick="detalhesVenda(${venda.id})">

<i class="fas fa-eye"></i> Detalhes

</button>

</td>

`;

vendasTable.appendChild(tr);

});

}

function atualizarResumoVendas(vendas) {

const totalVendas = vendas.reduce((total, venda) => total + venda.total, 0);

const vendasConfirmadas = vendas.filter(v => v.status === 'confirmado').length;

const vendasPendentes = vendas.filter(v => v.status === 'pendente').length;

const mediaVendas = vendas.length > 0 ? totalVendas / vendas.length : 0;

document.getElementById('resumoTotalVendas').textContent = `R$ ${totalVendas.toFixed(2)}`;

document.getElementById('resumoVendasConfirmadas').textContent = vendasConfirmadas;

document.getElementById('resumoVendasPendentes').textContent = vendasPendentes;

document.getElementById('resumoMediaVendas').textContent = `R$ ${mediaVendas.toFixed(2)}`;

}

async function detalhesVenda(id) {

try {

const response = await fetch(`${API\_BASE}/vendas/${id}`, {

credentials: 'include'

});

if (response.ok) {

const venda = await response.json();

exibirDetalhesVenda(venda);

}

} catch (error) {

mostrarMensagem('Erro ao carregar detalhes da venda', 'error');

}

}

function exibirDetalhesVenda(venda) {

const modal = document.getElementById('modalVendaDetalhes');

const detalhes = modal.querySelector('.modal-body');

detalhes.innerHTML = `

<div class="form-row">

<div class="form-group">

<label>ID da Venda</label>

<input type="text" value="#${venda.id}" readonly>

</div>

<div class="form-group">

<label>Data</label>

<input type="text" value="${formatarData(venda.data\_venda)}" readonly>

</div>

</div>

<div class="form-group">

<label>Cliente</label>

<input type="text" value="${venda.cliente}" readonly>

</div>

<div class="form-group">

<label>Itens</label>

<table class="data-table">

<thead>

<tr>

<th>Produto</th>

<th>Quantidade</th>

<th>Preço Unit.</th>

<th>Total</th>

</tr>

</thead>

<tbody>

${venda.itens.map(item => `

<tr>

<td>${item.produto\_nome}</td>

<td>${item.quantidade}</td>

<td>R$ ${item.preco\_unitario.toFixed(2)}</td>

<td>R$ ${(item.quantidade \* item.preco\_unitario).toFixed(2)}</td>

</tr>

`).join('')}

</tbody>

</table>

</div>

<div class="form-row">

<div class="form-group">

<label>Status</label>

<input type="text" value="${venda.status}" readonly>

</div>

<div class="form-group">

<label>Total</label>

<input type="text" value="R$ ${venda.total.toFixed(2)}" readonly>

</div>

</div>

`;

modal.classList.add('active');

}

// Utilitários

function formatarData(dataString) {

const data = new Date(dataString);

return data.toLocaleDateString('pt-BR');

}

function atualizarDataHora() {

const agora = new Date();

const data = agora.toLocaleDateString('pt-BR', {

weekday: 'long',

year: 'numeric',

month: 'long',

day: 'numeric'

});

const hora = agora.toLocaleTimeString('pt-BR');

document.getElementById('currentDate').textContent = data;

document.getElementById('currentTime').textContent = hora;

}

function fecharModal(modalId) {

document.getElementById(modalId).classList.remove('active');

}

function toggleSidebar() {

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

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

sidebar.classList.toggle('collapsed');

overlay.classList.toggle('active');

}

function mostrarMensagem(mensagem, tipo = 'success') {

// Criar elemento de mensagem

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

mensagemEl.className = `mensagem ${tipo}`;

mensagemEl.style.cssText = `

position: fixed;

top: 20px;

right: 20px;

padding: 15px 20px;

border-radius: 5px;

color: white;

z-index: 10000;

background-color: ${tipo === 'error' ? '#dc3545' : '#28a745'};

box-shadow: 0 5px 15px rgba(0,0,0,0.2);

`;

mensagemEl.textContent = mensagem;

document.body.appendChild(mensagemEl);

// Remover após 3 segundos

setTimeout(() => {

mensagemEl.remove();

}, 3000);

}

// Event Listeners

document.getElementById('loginForm').addEventListener('submit', function(e) {

e.preventDefault();

fazerLogin();

});

document.getElementById('cadastroForm').addEventListener('submit', function(e) {

e.preventDefault();

fazerCadastro();

});

document.getElementById('produtoForm').addEventListener('submit', function(e) {

e.preventDefault();

salvarProduto();

});

document.getElementById('funcionarioForm').addEventListener('submit', function(e) {

e.preventDefault();

salvarFuncionario();

});

// Fechar modais ao clicar no overlay

document.querySelectorAll('.modal').forEach(modal => {

modal.addEventListener('click', function(e) {

if (e.target === this) {

this.classList.remove('active');

}

});

});

// Fechar sidebar ao clicar no overlay

document.getElementById('sidebarOverlay').addEventListener('click', function() {

toggleSidebar();

});

// Preview de imagem

document.getElementById('produtoImagem').addEventListener('change', function(e) {

const file = e.target.files[0];

const preview = document.getElementById('produtoImagemPreview');

if (file) {

const reader = new FileReader();

reader.onload = function(e) {

preview.innerHTML = `<img src="${e.target.result}" alt="Preview">`;

};

reader.readAsDataURL(file);

}

});

# Arquivo: public/js/agenda.js

// agenda.js - Arquivo completo e funcional

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

// Elementos do DOM

const appointmentsList = document.querySelector('.appointments-list');

const filterPeriod = document.getElementById('filter-period');

const filterStatus = document.getElementById('filter-status');

const filterSort = document.getElementById('filter-sort');

const clearFiltersBtn = document.getElementById('clear-filters');

// Estados

let currentFilter = 'all';

let currentStatusFilter = 'all';

let currentSort = 'newest';

let allAppointments = [];

// Inicialização

init();

function init() {

if (!checkAuthentication()) {

return;

}

loadAppointments();

setupEventListeners();

verificarStatusPeriodicamente();

}

// Verificar se o usuário está logado

function checkAuthentication() {

const token = localStorage.getItem('token');

const userData = localStorage.getItem('user');

if (!token || !userData) {

showToast('Você precisa fazer login para ver seus agendamentos', 'error');

setTimeout(() => {

window.location.href = '/html/login.html?redirect=agenda.html';

}, 2000);

return false;

}

return true;

}

function setupEventListeners() {

// Filtro por período

if (filterPeriod) {

filterPeriod.addEventListener('change', function() {

currentFilter = this.value;

filterAppointments();

});

}

// Filtro por status

if (filterStatus) {

filterStatus.addEventListener('change', function() {

currentStatusFilter = this.value;

filterAppointments();

});

}

// Ordenação

if (filterSort) {

filterSort.addEventListener('change', function() {

currentSort = this.value;

filterAppointments();

});

}

// Limpar filtros

if (clearFiltersBtn) {

clearFiltersBtn.addEventListener('click', function() {

currentFilter = 'all';

currentStatusFilter = 'all';

currentSort = 'newest';

if (filterPeriod) filterPeriod.value = 'all';

if (filterStatus) filterStatus.value = 'all';

if (filterSort) filterSort.value = 'newest';

filterAppointments();

});

}

}

// Carregar agendamentos do backend

async function loadAppointments() {

showLoading(true);

try {

// Obter ID do usuário logado

const userData = localStorage.getItem('user');

if (!userData) {

throw new Error('Usuário não está logado');

}

const user = JSON.parse(userData);

const userId = user.id;

if (!userId) {

throw new Error('ID do usuário não encontrado');

}

// Buscar agendamentos específicos do usuário

const response = await fetch(`/api/agendamento\_simples/usuario/${userId}`);

if (!response.ok) {

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

}

const result = await response.json();

if (result.success) {

allAppointments = result.data || [];

displayAppointments(allAppointments);

} else {

throw new Error(result.message || 'Erro ao carregar agendamentos');

}

} catch (error) {

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

showError('Erro ao carregar agendamentos. Verifique se está logado.');

// Se não estiver logado, redirecionar para login

if (error.message.includes('não está logado')) {

setTimeout(() => {

window.location.href = '/html/login.html?redirect=agenda.html';

}, 2000);

}

} finally {

showLoading(false);

}

}

// agenda.js - Substitua a função loadLembreteAutomatico por esta versão melhorada

// Buscar lembrete inteligente baseado no status dos agendamentos

async function loadLembreteInteligente() {

try {

const userData = localStorage.getItem('user');

if (!userData) return;

const user = JSON.parse(userData);

const userId = user.id;

if (!userId) return;

// Buscar agendamentos do usuário

const response = await fetch(`/api/agendamento\_simples/usuario/${userId}`);

if (response.ok) {

const result = await response.json();

if (result.success && result.data && result.data.length > 0) {

const agendamentos = result.data;

// Verificar se existe agendamento futuro

const agora = new Date();

const agendamentoFuturo = agendamentos.find(ag => {

const dataAgendamento = new Date(ag.data\_hora);

return dataAgendamento > agora &&

ag.status !== 'cancelado' &&

ag.status !== 'fora\_prazo';

});

if (agendamentoFuturo) {

// Se tem agendamento futuro, mostrar informações dele

displayProximoAgendamento(agendamentoFuturo);

} else {

// Se não tem agendamento futuro, mostrar lembrete baseado no último serviço

const ultimoConcluido = agendamentos

.filter(ag => ag.status === 'concluido' || new Date(ag.data\_hora) < agora)

.sort((a, b) => new Date(b.data\_hora) - new Date(a.data\_hora))[0];

if (ultimoConcluido) {

displayLembreteAutomatico(ultimoConcluido);

} else {

displayLembretePadrao();

}

}

} else {

displayLembretePadrao();

}

} else {

displayLembretePadrao();

}

} catch (error) {

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

displayLembretePadrao();

}

}

// Exibir informações do próximo agendamento

function displayProximoAgendamento(agendamento) {

const reminderSection = document.querySelector('.reminder-section');

if (!reminderSection) return;

const dataAgendamento = new Date(agendamento.data\_hora);

const agora = new Date();

const diffTime = dataAgendamento - agora;

const diffDays = Math.ceil(diffTime / (1000 \* 60 \* 60 \* 24));

let statusText = '';

if (diffDays === 0) {

statusText = '<span style="color: #28a745;">⚡ É hoje!</span>';

} else if (diffDays === 1) {

statusText = '<span style="color: #ffc107;">⚠️ Amanhã!</span>';

} else if (diffDays <= 7) {

statusText = `<span style="color: #fd7e14;">📅 Em ${diffDays} dias</span>`;

} else {

statusText = `<span style="color: #17a2b8;">📅 Em ${diffDays} dias</span>`;

}

reminderSection.innerHTML = `

<div class="section-header">

<h2><i class="fas fa-calendar-check"></i> Próximo Agendamento</h2>

</div>

<div class="reminder-card next-appointment">

<div class="reminder-content">

<div class="reminder-icon" style="background: #e7f3ff; color: #0066cc;">

<i class="fas fa-clock"></i>

</div>

<div class="reminder-details">

<h3 style="color: #0066cc;">${agendamento.servicos || 'Serviço Agendado'}</h3>

<p><strong>Veículo:</strong> ${agendamento.veiculo || 'Veículo não informado'}</p>

<p><strong>Data e Horário:</strong> ${dataAgendamento.toLocaleDateString('pt-BR')} às ${dataAgendamento.toLocaleTimeString('pt-BR', { hour: '2-digit', minute: '2-digit' })}</p>

<p><strong>Local:</strong> ${agendamento.oficina\_nome || 'Oficina'} - ${agendamento.oficina\_endereco || 'Endereço não informado'}</p>

<p><strong>Status:</strong> ${statusText}</p>

${agendamento.total\_servico ? `<p><strong>Valor:</strong> R$ ${parseFloat(agendamento.total\_servico).toFixed(2)}</p>` : ''}

</div>

</div>

<div class="reminder-actions">

${diffDays > 1 ? `

` : ''}

<button class="btn btn-primary" onclick="reagendarServico(${agendamento.id})">

<i class="fas fa-copy"></i> Repetir Este Serviço

</button>

</div>

</div>

`;

}

// Atualize também a função displayLembreteAutomatico para ficar mais clara:

function displayLembreteAutomatico(ultimoServico) {

const reminderSection = document.querySelector('.reminder-section');

if (!reminderSection) return;

const dataServico = new Date(ultimoServico.data\_hora);

// Calcular próxima troca (6 meses após o serviço)

const proximaTrocaData = new Date(dataServico);

proximaTrocaData.setMonth(proximaTrocaData.getMonth() + 6);

// Verificar se já passou da data recomendada

const hoje = new Date();

const estaAtrasado = proximaTrocaData < hoje;

// Calcular dias de atraso ou até a próxima troca

const diffTime = Math.abs(proximaTrocaData - hoje);

const diffDays = Math.ceil(diffTime / (1000 \* 60 \* 60 \* 24));

let statusText = '';

if (estaAtrasado) {

statusText = `<span style="color: #dc3545;">⏰ Atrasado há ${diffDays} dias</span>`;

} else if (diffDays <= 30) {

statusText = `<span style="color: #fd7e14;">⚠️ Em ${diffDays} dias</span>`;

} else {

statusText = `<span style="color: #28a745;">📅 Em ${diffDays} dias</span>`;

}

reminderSection.innerHTML = `

<div class="section-header">

<h2><i class="fas fa-bell"></i> Lembrete de Manutenção</h2>

</div>

<div class="reminder-card ${estaAtrasado ? 'reminder-urgent' : ''}">

<div class="reminder-content">

<div class="reminder-icon ${estaAtrasado ? 'urgent' : ''}">

<i class="fas fa-oil-can"></i>

</div>

<div class="reminder-details">

<h3>${ultimoServico.servicos || 'Troca de Óleo e Filtro'}</h3>

<p><strong>Veículo:</strong> ${ultimoServico.veiculo || 'Seu veículo'}</p>

<p><strong>Último serviço:</strong> ${dataServico.toLocaleDateString('pt-BR')}</p>

<p><strong>Próxima troca recomendada:</strong> ${proximaTrocaData.toLocaleDateString('pt-BR')}</p>

<p><strong>Status:</strong> ${statusText}</p>

${estaAtrasado ?

'<p style="color: #dc3545; font-weight: bold;">⚠️ Sua troca de óleo está atrasada!</p>' :

'<p style="color: #28a745;">✅ Sua manutenção está em dia</p>'

}

</div>

</div>

<div class="reminder-actions">

<a href="/html/servicos.html" class="btn btn-primary">

<i class="fas fa-calendar-plus"></i> Agendar Nova Troca

</a>

<button class="btn btn-outline" onclick="reagendarUltimoServico(${ultimoServico.id})">

<i class="fas fa-redo"></i> Repetir Último Serviço

</button>

</div>

</div>

`;

}

// Função para ver detalhes do agendamento

function verDetalhesAgendamento(agendamentoId) {

// Rolar até o card do agendamento

const card = document.querySelector(`.appointment-card[data-id="${agendamentoId}"]`);

if (card) {

card.scrollIntoView({ behavior: 'smooth', block: 'center' });

// Efeito visual de destaque

card.style.animation = 'pulse 2s ease-in-out';

setTimeout(() => {

card.style.animation = '';

}, 2000);

}

}

// Função para reagendar qualquer serviço

async function reagendarServico(agendamentoId) {

try {

const response = await fetch(`/api/agendamento\_simples/${agendamentoId}`);

const result = await response.json();

if (result.success) {

const agendamento = result.data;

sessionStorage.setItem('ultimoAgendamento', JSON.stringify({

veiculo: agendamento.veiculo,

servicos: agendamento.servicos,

tipo\_oleo: agendamento.tipo\_oleo,

oficina: {

nome: agendamento.oficina\_nome,

endereco: agendamento.oficina\_endereco

}

}));

window.location.href = '/html/servicos.html';

}

} catch (error) {

console.error('Erro ao reagendar:', error);

showToast('Erro ao carregar dados do serviço', 'error');

}

}

// Na função init(), atualize para:

function init() {

if (!checkAuthentication()) {

return;

}

loadAppointments();

loadLembreteInteligente(); // ← TROQUEI PARA A NOVA FUNÇÃO

setupEventListeners();

verificarStatusPeriodicamente();

}

// Filtrar e ordenar agendamentos

function filterAppointments() {

let filtered = [...allAppointments];

// Filtro por período

if (currentFilter !== 'all') {

const now = new Date();

filtered = filtered.filter(appointment => {

const appointmentDate = new Date(appointment.data\_hora);

switch (currentFilter) {

case 'today':

return isToday(appointmentDate);

case 'week':

return isThisWeek(appointmentDate);

case 'month':

return isThisMonth(appointmentDate);

case 'last\_month':

return isLastMonth(appointmentDate);

case 'year':

return isThisYear(appointmentDate);

default:

return true;

}

});

}

// Filtro por status

if (currentStatusFilter !== 'all') {

filtered = filtered.filter(appointment => {

const appointmentDate = new Date(appointment.data\_hora);

const now = new Date();

switch (currentStatusFilter) {

case 'pendente':

return appointment.status === 'pendente' || (appointmentDate > now && (!appointment.status || appointment.status === 'pendente'));

case 'agendado':

return appointment.status === 'confirmado' || (appointmentDate > now && appointment.status !== 'cancelado');

case 'concluido':

return appointment.status === 'concluido';

case 'cancelado':

return appointment.status === 'cancelado';

case 'fora\_prazo':

return appointment.status === 'fora\_prazo' || (appointmentDate < now && appointment.status !== 'concluido' && appointment.status !== 'cancelado');

default:

return true;

}

});

}

// Ordenação

filtered.sort((a, b) => {

const dateA = new Date(a.data\_hora);

const dateB = new Date(b.data\_hora);

switch (currentSort) {

case 'newest':

return dateB - dateA;

case 'oldest':

return dateA - dateB;

case 'date\_asc':

return dateA - dateB;

case 'date\_desc':

return dateB - dateA;

default:

return dateB - dateA;

}

});

displayAppointments(filtered);

}

// Exibir agendamentos na página

function displayAppointments(appointments) {

if (!appointmentsList) return;

if (appointments.length === 0) {

appointmentsList.innerHTML = `

<div class="no-appointments">

<i class="fas fa-calendar-times"></i>

<h3>Nenhum agendamento encontrado</h3>

<p>Não há agendamentos para os filtros selecionados.</p>

</div>

`;

return;

}

appointmentsList.innerHTML = appointments.map(appointment => {

const statusClass = getStatusClass(appointment);

const statusText = getStatusText(appointment);

return `

<div class="appointment-card ${statusClass}" data-id="${appointment.id}">

<div class="appointment-header">

<span class="protocol">Protocolo: ${appointment.protocolo || 'N/A'}</span>

<span class="status ${statusClass}">${statusText}</span>

</div>

<div class="appointment-body">

<div class="service-info">

<h3>${appointment.servicos || 'Troca de Óleo'}</h3>

<p><i class="fas fa-car"></i> ${appointment.veiculo || 'Veículo não informado'}</p>

${appointment.servicos ? `<p><i class="fas fa-oil-can"></i> ${appointment.servicos.replace(/[\[\]"]/g, '')}</p>` : ''}

</div>

<div class="date-info">

<p><i class="far fa-calendar-alt"></i> ${formatDate(appointment.data\_hora)}</p>

<p><i class="far fa-clock"></i> ${formatTime(appointment.data\_hora)}</p>

</div>

<div class="location-info">

<p><i class="fas fa-map-marker-alt"></i> ${appointment.oficina\_nome || 'Oficina'}</p>

<p>${appointment.oficina\_endereco || 'Endereço não informado'}</p>

</div>

${appointment.total\_servico ? `

<div class="price-info">

<p><i class="fas fa-tag"></i> Total: R$ ${parseFloat(appointment.total\_servico).toFixed(2)}</p>

</div>

` : ''}

${appointment.motivo\_cancelamento ? `

<div class="cancel-info">

<p><i class="fas fa-info-circle"></i> Motivo: ${appointment.motivo\_cancelamento}</p>

</div>

` : ''}

</div>

<div class="appointment-footer">

${statusClass !== 'cancelled' && statusClass !== 'expired' ? `

<button class="btn btn-outline" onclick="repeatAppointment(${appointment.id})">

<i class="fas fa-redo"></i> Repetir Agendamento

</button>

${isFutureAppointment(appointment) ? `

<button class="btn btn-outline btn-cancel" onclick="cancelAppointment(${appointment.id})">

<i class="fas fa-times"></i> Cancelar

</button>

` : `

<button class="btn btn-outline" onclick="rateService(${appointment.id})">

<i class="fas fa-star"></i> Avaliar Serviço

</button>

`}

` : `

<button class="btn btn-outline" onclick="repeatAppointment(${appointment.id})">

<i class="fas fa-redo"></i> Novo Agendamento

</button>

<span class="text-muted">Ações indisponíveis</span>

`}

</div>

</div>

`;

}).join('');

}

// Funções auxiliares

function getStatusClass(appointment) {

const appointmentDate = new Date(appointment.data\_hora);

const now = new Date();

if (appointment.status === 'cancelado') {

return 'cancelled';

} else if (appointment.status === 'fora\_prazo') {

return 'expired';

} else if (appointmentDate > now) {

return 'pending';

} else if (appointment.status === 'concluido') {

return 'completed';

} else {

return 'expired';

}

}

function getStatusText(appointment) {

const appointmentDate = new Date(appointment.data\_hora);

const now = new Date();

if (appointment.status === 'cancelado') {

return 'Cancelado';

} else if (appointment.status === 'fora\_prazo') {

return 'Fora do Prazo';

} else if (appointmentDate > now) {

return 'Agendado';

} else if (appointment.status === 'concluido') {

return 'Concluído';

} else {

return 'Fora do Prazo';

}

}

function isFutureAppointment(appointment) {

const appointmentDate = new Date(appointment.data\_hora);

const now = new Date();

return appointmentDate > now && appointment.status !== 'cancelado';

}

function formatDate(dateString) {

const date = new Date(dateString);

return date.toLocaleDateString('pt-BR');

}

function formatTime(dateString) {

const date = new Date(dateString);

return date.toLocaleTimeString('pt-BR', {

hour: '2-digit',

minute: '2-digit'

});

}

// Funções de filtro por data

function isToday(date) {

const today = new Date();

return date.getDate() === today.getDate() &&

date.getMonth() === today.getMonth() &&

date.getFullYear() === today.getFullYear();

}

function isThisWeek(date) {

const today = new Date();

const startOfWeek = new Date(today);

startOfWeek.setDate(today.getDate() - today.getDay());

startOfWeek.setHours(0, 0, 0, 0);

const endOfWeek = new Date(startOfWeek);

endOfWeek.setDate(startOfWeek.getDate() + 6);

endOfWeek.setHours(23, 59, 59, 999);

return date >= startOfWeek && date <= endOfWeek;

}

function isThisMonth(date) {

const today = new Date();

return date.getMonth() === today.getMonth() &&

date.getFullYear() === today.getFullYear();

}

function isLastMonth(date) {

const today = new Date();

const lastMonth = today.getMonth() === 0 ? 11 : today.getMonth() - 1;

const year = today.getMonth() === 0 ? today.getFullYear() - 1 : today.getFullYear();

return date.getMonth() === lastMonth && date.getFullYear() === year;

}

function isThisYear(date) {

const today = new Date();

return date.getFullYear() === today.getFullYear();

}

function showLoading(show) {

// Implemente um overlay de loading se necessário

const loadingElement = document.getElementById('loading-agenda');

if (loadingElement) {

loadingElement.style.display = show ? 'block' : 'none';

}

}

function showError(message) {

if (typeof showToast === 'function') {

showToast(message, 'error');

} else {

alert(message);

}

}

// Verificar status periodicamente

function verificarStatusPeriodicamente() {

// Atualizar status no servidor a cada hora

setInterval(async () => {

try {

await fetch('/api/agendamento\_simples/atualizar-status/automatico', {

method: 'POST'

});

} catch (error) {

console.error('Erro ao verificar status:', error);

}

}, 60 \* 60 \* 1000); // 1 hora

// Recarregar agendamentos a cada 5 minutos

setInterval(() => {

loadAppointments();

}, 5 \* 60 \* 1000);

}

});

// Funções globais para os botões

async function repeatAppointment(appointmentId) {

try {

// Buscar dados do agendamento

const response = await fetch(`/api/agendamento\_simples/${appointmentId}`);

const result = await response.json();

if (result.success) {

const appointment = result.data;

// Armazenar dados para usar na página de serviços

sessionStorage.setItem('lastAppointment', JSON.stringify({

veiculo: appointment.veiculo,

servicos: appointment.servicos,

oficina: {

nome: appointment.oficina\_nome,

endereco: appointment.oficina\_endereco

}

}));

// Redirecionar para serviços

window.location.href = '/html/servicos.html';

}

} catch (error) {

console.error('Erro ao repetir agendamento:', error);

alert('Erro ao carregar dados do agendamento.');

}

}

// Função melhorada para download de comprovante

// Função melhorada para download de comprovante

async function downloadInvoice(appointmentId) {

try {

console.log('🎯 Iniciando download do comprovante para agendamento:', appointmentId);

// Mostrar estado de carregamento

const button = event.target.closest('.btn');

const originalText = button.innerHTML;

button.innerHTML = '<i class="fas fa-spinner fa-spin"></i> Gerando...';

button.disabled = true;

// Buscar dados completos do agendamento

const response = await fetch(`/api/agendamento\_simples/${appointmentId}`);

const result = await response.json();

if (result.success) {

// Gerar PDF com dados do agendamento

await generatePDF(result.data);

// Mostrar confirmação

showDownloadSuccess();

} else {

throw new Error(result.message || 'Erro ao buscar dados do agendamento');

}

} catch (error) {

console.error('❌ Erro ao gerar comprovante:', error);

showToast('❌ Erro ao gerar comprovante: ' + error.message, 'error');

} finally {

// Restaurar estado do botão

const button = event.target.closest('.btn');

button.innerHTML = '<i class="fas fa-file-invoice"></i> Comprovante';

button.disabled = false;

}

}

// Função aprimorada para gerar PDF

async function generatePDF(appointment) {

return new Promise((resolve, reject) => {

try {

const { jsPDF } = window.jspdf;

const pdf = new jsPDF();

// Configurações do documento

pdf.setFont('helvetica');

pdf.setFontSize(20);

// Cabeçalho

pdf.setTextColor(33, 63, 87); // Cor primária

pdf.text('OilSmart - Comprovante de Agendamento', 20, 30);

// Linha decorativa

pdf.setDrawColor(180, 148, 52); // Cor secundária

pdf.setLineWidth(1);

pdf.line(20, 35, 190, 35);

// Informações do protocolo

pdf.setFontSize(12);

pdf.setTextColor(0, 0, 0);

pdf.text(`Protocolo: ${appointment.protocolo || 'N/A'}`, 20, 50);

pdf.text(`Data de Emissão: ${new Date().toLocaleDateString('pt-BR')}`, 20, 60);

let yPosition = 80;

// Dados do agendamento

const details = [

`Data do Serviço: ${new Date(appointment.data\_hora).toLocaleDateString('pt-BR')}`,

`Horário: ${new Date(appointment.data\_hora).toLocaleTimeString('pt-BR', {

hour: '2-digit',

minute: '2-digit'

})}`,

`Oficina: ${appointment.oficina\_nome || 'Oficina não informada'}`,

`Endereço: ${appointment.oficina\_endereco || 'Endereço não informado'}`,

`Veículo: ${appointment.veiculo || 'Veículo não informado'}`,

`Serviços: ${appointment.servicos || 'Troca de óleo'}`,

`Status: ${getStatusText(appointment)}`

];

// Adicionar detalhes

pdf.setFontSize(11);

details.forEach(detail => {

pdf.text(detail, 20, yPosition);

yPosition += 8;

});

// Adicionar preço se disponível

if (appointment.total\_servico) {

yPosition += 10;

pdf.setFontSize(12);

pdf.setFont(undefined, 'bold');

pdf.text(`Valor Total: R$ ${parseFloat(appointment.total\_servico).toFixed(2)}`, 20, yPosition);

pdf.setFont(undefined, 'normal');

}

// Adicionar motivo do cancelamento se aplicável

if (appointment.motivo\_cancelamento) {

yPosition += 15;

pdf.setFontSize(10);

pdf.setTextColor(220, 53, 69); // Vermelho

pdf.text(`Motivo do Cancelamento: ${appointment.motivo\_cancelamento}`, 20, yPosition);

pdf.setTextColor(0, 0, 0);

}

// Informações de contato

yPosition += 20;

pdf.setFontSize(9);

pdf.setTextColor(100, 100, 100);

pdf.text('Para dúvidas ou cancelamentos, entre em contato:', 20, yPosition);

pdf.text('Telefone: (11) 4002-8922 | Email: contato@oilsmart.com.br', 20, yPosition + 5);

// Rodapé

yPosition = 270;

pdf.setFontSize(8);

pdf.setTextColor(128, 128, 128);

pdf.text('Este é um comprovante gerado automaticamente pelo sistema OilSmart.', 20, yPosition);

pdf.text(`Comprovante gerado em: ${new Date().toLocaleString('pt-BR')}`, 20, yPosition + 5);

// Salvar PDF

const fileName = `comprovante-${appointment.protocolo || appointment.id}.pdf`;

pdf.save(fileName);

resolve();

} catch (error) {

console.error('Erro na geração do PDF:', error);

reject(error);

}

});

}

// Função para mostrar confirmação de download

function showDownloadSuccess() {

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

modal.className = 'download-modal show';

modal.innerHTML = `

<div class="download-modal-content">

<div class="download-icon">

<i class="fas fa-check-circle"></i>

</div>

<h3>Download Concluído!</h3>

<p>O comprovante foi baixado com sucesso. Verifique sua pasta de downloads.</p>

<div class="download-modal-buttons">

<button class="btn btn-primary" onclick="closeDownloadModal()">

<i class="fas fa-check"></i> OK

</button>

</div>

</div>

`;

document.body.appendChild(modal);

// Fechar modal após 5 segundos automaticamente

setTimeout(() => {

closeDownloadModal();

}, 5000);

}

// Função para fechar o modal

function closeDownloadModal() {

const modal = document.querySelector('.download-modal');

if (modal) {

modal.remove();

}

}

// Adicionar evento de clique fora do modal para fechar

document.addEventListener('click', function(event) {

if (event.target.classList.contains('download-modal')) {

closeDownloadModal();

}

});

// Função para mostrar confirmação de download

function showDownloadSuccess() {

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

modal.className = 'download-modal show';

modal.innerHTML = `

<div class="download-modal-content">

<div class="download-icon">

<i class="fas fa-check-circle"></i>

</div>

<h3>Download Concluído!</h3>

<p>O comprovante foi baixado com sucesso. Verifique sua pasta de downloads.</p>

<div class="download-modal-buttons">

<button class="btn btn-primary" onclick="closeDownloadModal()">

<i class="fas fa-check"></i> OK

</button>

</div>

</div>

`;

document.body.appendChild(modal);

// Fechar modal após 5 segundos automaticamente

setTimeout(() => {

closeDownloadModal();

}, 5000);

}

// Função para fechar o modal

function closeDownloadModal() {

const modal = document.querySelector('.download-modal');

if (modal) {

modal.remove();

}

}

// Adicionar evento de clique fora do modal para fechar

document.addEventListener('click', function(event) {

if (event.target.classList.contains('download-modal')) {

closeDownloadModal();

}

});

async function cancelAppointment(appointmentId) {

console.log('🎯 Iniciando cancelamento do agendamento:', appointmentId);

// Verificar autenticação

const token = localStorage.getItem('token');

if (!token) {

showToast('Você precisa estar logado para cancelar agendamentos', 'error');

setTimeout(() => {

window.location.href = '/html/login.html?redirect=agenda.html';

}, 2000);

return;

}

try {

// Verificar se é um agendamento futuro

const response = await fetch(`/api/agendamento\_simples/${appointmentId}`);

const result = await response.json();

if (!result.success) {

throw new Error('Agendamento não encontrado');

}

const appointment = result.data;

const appointmentDate = new Date(appointment.data\_hora);

const now = new Date();

// Verificar se já passou da data

if (appointmentDate <= now) {

showToast('Não é possível cancelar agendamentos passados', 'error');

return;

}

// Verificar se já está cancelado

if (appointment.status === 'cancelado') {

showToast('Este agendamento já está cancelado', 'warning');

return;

}

} catch (error) {

console.error('Erro ao verificar agendamento:', error);

showToast('Erro ao verificar agendamento', 'error');

return;

}

// Pedir motivo do cancelamento com um modal mais amigável

const motivo = await showCancelReasonModal();

if (!motivo) {

showToast('Cancelamento não realizado', 'info');

return;

}

if (!confirm('Tem certeza que deseja cancelar este agendamento?\nEsta ação não pode ser desfeita.')) {

return;

}

try {

// Mostrar loading

showCancelLoading(true);

const response = await fetch(`/api/agendamento\_simples/${appointmentId}/cancelar`, {

method: 'POST',

headers: {

'Content-Type': 'application/json',

'Authorization': `Bearer ${token}`

},

body: JSON.stringify({

motivo: motivo.trim() || 'Cancelado pelo cliente'

})

});

const result = await response.json();

if (result.success) {

if (confirm('✅ Agendamento cancelado com sucesso!\n\nDeseja atualizar a página agora?')) {

location.reload();

}

} else {

throw new Error(result.message || 'Erro ao cancelar agendamento');

}

} catch (error) {

console.error('❌ Erro ao cancelar agendamento:', error);

showToast('❌ Erro ao cancelar agendamento: ' + error.message, 'error');

} finally {

showCancelLoading(false);

}

}

// Modal para inserir motivo do cancelamento

function showCancelReasonModal() {

return new Promise((resolve) => {

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

modal.style.cssText = `

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

display: flex;

justify-content: center;

align-items: center;

z-index: 10000;

`;

modal.innerHTML = `

<div style="background: white; padding: 30px; border-radius: 10px; width: 90%; max-width: 500px; box-shadow: 0 5px 15px rgba(0,0,0,0.3);">

<h3 style="margin-bottom: 15px; color: #333;">Motivo do Cancelamento</h3>

<p style="margin-bottom: 15px; color: #666;">Por favor, informe o motivo do cancelamento:</p>

<textarea

id="cancel-reason-input"

placeholder="Ex: Mudança de planos, problema no veículo, etc."

style="width: 100%; height: 100px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; resize: vertical; font-family: inherit;"

></textarea>

<div style="margin-top: 20px; display: flex; gap: 10px; justify-content: flex-end;">

<button id="cancel-btn" style="padding: 10px 20px; border: 1px solid #ddd; background: white; border-radius: 5px; cursor: pointer;">Cancelar</button>

<button id="confirm-btn" style="padding: 10px 20px; background: #dc3545; color: white; border: none; border-radius: 5px; cursor: pointer;">Confirmar Cancelamento</button>

</div>

</div>

`;

document.body.appendChild(modal);

const input = modal.querySelector('#cancel-reason-input');

const cancelBtn = modal.querySelector('#cancel-btn');

const confirmBtn = modal.querySelector('#confirm-btn');

input.focus();

cancelBtn.onclick = () => {

document.body.removeChild(modal);

resolve(null);

};

confirmBtn.onclick = () => {

const motivo = input.value.trim();

if (!motivo) {

showToast('Por favor, informe o motivo do cancelamento', 'warning');

input.focus();

return;

}

document.body.removeChild(modal);

resolve(motivo);

};

// Fechar modal ao clicar fora

modal.onclick = (e) => {

if (e.target === modal) {

document.body.removeChild(modal);

resolve(null);

}

};

// Enter para confirmar

input.onkeydown = (e) => {

if (e.key === 'Enter' && e.ctrlKey) {

confirmBtn.click();

}

};

});

}

function showSuccessModal() {

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

modal.style.cssText = `

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0,0,0,0.8);

display: flex;

justify-content: center;

align-items: center;

z-index: 10000;

`;

modal.innerHTML = `

<div style="

background: white;

padding: 30px;

border-radius: 10px;

text-align: center;

max-width: 400px;

box-shadow: 0 5px 15px rgba(0,0,0,0.3);

">

<div style="font-size: 48px; margin-bottom: 15px;">✅</div>

<h3 style="margin-bottom: 15px; color: #28a745;">Cancelado com Sucesso!</h3>

<p style="margin-bottom: 20px; color: #666;">

Seu agendamento foi cancelado.<br>

Atualize a página para ver as alterações.

</p>

<button onclick="location.reload()" style="

background: #28a745;

color: white;

border: none;

padding: 10px 20px;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

">

🔄 Atualizar Página

</button>

</div>

`;

document.body.appendChild(modal);

// Fechar modal ao clicar fora (opcional)

modal.onclick = function(e) {

if (e.target === modal) {

document.body.removeChild(modal);

}

};

}

// Função para mostrar/ocultar loading

function showCancelLoading(show) {

let loadingElement = document.getElementById('cancel-loading');

if (show) {

if (!loadingElement) {

loadingElement = document.createElement('div');

loadingElement.id = 'cancel-loading';

loadingElement.style.cssText = `

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

display: flex;

justify-content: center;

align-items: center;

z-index: 10001;

color: white;

font-size: 18px;

`;

loadingElement.innerHTML = `

<div style="text-align: center;">

<div class="spinner" style="border: 4px solid #f3f3f3; border-top: 4px solid #3498db; border-radius: 50%; width: 50px; height: 50px; animation: spin 1s linear infinite; margin: 0 auto 15px;"></div>

<p>Cancelando agendamento...</p>

</div>

`;

document.body.appendChild(loadingElement);

// Adicionar animação do spinner

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

style.textContent = `

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

`;

document.head.appendChild(style);

}

} else {

if (loadingElement) {

document.body.removeChild(loadingElement);

}

}

}

// Função para atualizar a interface do agendamento cancelado

async function updateAppointmentUI(appointmentId) {

try {

// Buscar dados atualizados do agendamento

const response = await fetch(`/api/agendamento\_simples/${appointmentId}`);

const result = await response.json();

if (result.success) {

const updatedAppointment = result.data;

// Encontrar o card do agendamento no DOM

const appointmentCard = document.querySelector(`.appointment-card[data-id="${appointmentId}"]`);

if (appointmentCard) {

// Atualizar o card com os novos dados

updateAppointmentCard(appointmentCard, updatedAppointment);

} else {

// Se não encontrar, recarregar a lista

console.log('Card não encontrado, recarregando lista...');

await loadAppointments();

}

}

} catch (error) {

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

// Em caso de erro, recarregar tudo

await loadAppointments();

}

}

// Função para atualizar um card específico

function updateAppointmentCard(cardElement, appointment) {

const statusClass = getStatusClass(appointment);

const statusText = getStatusText(appointment);

// Atualizar o status no header

const statusElement = cardElement.querySelector('.status');

if (statusElement) {

statusElement.className = `status ${statusClass}`;

statusElement.textContent = statusText;

}

// Atualizar a classe principal do card

cardElement.className = `appointment-card ${statusClass}`;

cardElement.setAttribute('data-id', appointment.id);

// Atualizar o protocolo se necessário

const protocolElement = cardElement.querySelector('.protocol');

if (protocolElement && appointment.protocolo) {

protocolElement.textContent = `Protocolo: ${appointment.protocolo}`;

}

// Adicionar/atualizar informações de cancelamento

let cancelInfo = cardElement.querySelector('.cancel-info');

if (appointment.motivo\_cancelamento) {

if (!cancelInfo) {

cancelInfo = document.createElement('div');

cancelInfo.className = 'cancel-info';

cardElement.querySelector('.appointment-body').appendChild(cancelInfo);

}

cancelInfo.innerHTML = `<p><i class="fas fa-info-circle"></i> Motivo: ${appointment.motivo\_cancelamento}</p>`;

} else if (cancelInfo) {

cancelInfo.remove();

}

// Atualizar os botões do footer

updateAppointmentFooter(cardElement, appointment);

// Adicionar efeito visual de atualização

cardElement.style.animation = 'pulse 0.5s ease-in-out';

setTimeout(() => {

cardElement.style.animation = '';

}, 500);

}

// Função para atualizar os botões do footer baseado no status

function updateAppointmentFooter(cardElement, appointment) {

const footer = cardElement.querySelector('.appointment-footer');

if (!footer) return;

const statusClass = getStatusClass(appointment);

if (statusClass === 'cancelled' || statusClass === 'expired') {

footer.innerHTML = `

<button class="btn btn-outline" onclick="repeatAppointment(${appointment.id})">

<i class="fas fa-redo"></i> Novo Agendamento

</button>

<button class="btn btn-outline" onclick="downloadInvoice(${appointment.id})">

<i class="fas fa-file-invoice"></i> Comprovante

</button>

<span class="text-muted">Ações indisponíveis</span>

`;

} else if (isFutureAppointment(appointment)) {

footer.innerHTML = `

<button class="btn btn-outline" onclick="repeatAppointment(${appointment.id})">

<i class="fas fa-redo"></i> Repetir Agendamento

</button>

<button class="btn btn-outline" onclick="downloadInvoice(${appointment.id})">

<i class="fas fa-file-invoice"></i> Comprovante

</button>

<button class="btn btn-outline btn-cancel" onclick="cancelAppointment(${appointment.id})">

<i class="fas fa-times"></i> Cancelar

</button>

`;

} else {

footer.innerHTML = `

<button class="btn btn-outline" onclick="repeatAppointment(${appointment.id})">

<i class="fas fa-redo"></i> Repetir Agendamento

</button>

<button class="btn btn-outline" onclick="downloadInvoice(${appointment.id})">

<i class="fas fa-file-invoice"></i> Comprovante

</button>

<button class="btn btn-outline" onclick="rateService(${appointment.id})">

<i class="fas fa-star"></i> Avaliar Serviço

</button>

`;

}

}

function rateService(appointmentId) {

alert('Sistema de avaliação em desenvolvimento!');

}

// Função auxiliar para mostrar toast (caso não exista no main.js)

function showToast(message, type = 'info') {

// Se a função showToast não existir, criar uma simples

if (typeof window.showToast !== 'function') {

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

toast.style.cssText = `

position: fixed;

top: 20px;

right: 20px;

padding: 15px 20px;

border-radius: 5px;

color: white;

z-index: 10000;

font-family: Arial, sans-serif;

max-width: 300px;

word-wrap: break-word;

`;

switch (type) {

case 'success':

toast.style.backgroundColor = '#28a745';

break;

case 'error':

toast.style.backgroundColor = '#dc3545';

break;

case 'warning':

toast.style.backgroundColor = '#ffc107';

toast.style.color = '#212529';

break;

default:

toast.style.backgroundColor = '#17a2b8';

}

toast.textContent = message;

document.body.appendChild(toast);

setTimeout(() => {

document.body.removeChild(toast);

}, 5000);

} else {

window.showToast(message, type);

}

}

// CSS adicional para estilização

const additionalCSS = `

/\* Status dos agendamentos \*/

.status.pending {

background: #fff3cd;

color: #856404;

border: 1px solid #ffeaa7;

}

.status.completed {

background: #d1edff;

color: #0c5460;

border: 1px solid #b3e0ff;

}

.status.expired {

background: #f8f9fa;

color: #6c757d;

border: 1px solid #dee2e6;

text-decoration: line-through;

}

.status.cancelled {

background: #f8d7da;

color: #721c24;

border: 1px solid #f5c6cb;

}

/\* Faixa de cancelado \*/

.appointment-card.cancelled::before {

content: "CANCELADO";

position: absolute;

top: 10px;

right: -30px;

background: #dc3545;

color: white;

padding: 5px 40px;

font-size: 12px;

font-weight: bold;

transform: rotate(45deg);

box-shadow: 0 2px 4px rgba(0,0,0,0.2);

z-index: 10;

}

.appointment-card.expired::before {

content: "FORA DO PRAZO";

position: absolute;

top: 10px;

right: -35px;

background: #6c757d;

color: white;

padding: 5px 40px;

font-size: 12px;

font-weight: bold;

transform: rotate(45deg);

box-shadow: 0 2px 4px rgba(0,0,0,0.2);

z-index: 10;

}

.appointment-card {

position: relative;

overflow: hidden;

}

/\* Desabilitar botões para agendamentos cancelados/vencidos \*/

.appointment-card.cancelled .btn,

.appointment-card.expired .btn {

opacity: 0.6;

cursor: not-allowed;

}

.appointment-card.cancelled .btn:hover,

.appointment-card.expired .btn:hover {

transform: none;

box-shadow: none;

}

/\* Estilo para quando não há agendamentos \*/

.no-appointments {

text-align: center;

padding: 60px 20px;

color: #6c757d;

}

.no-appointments i {

font-size: 64px;

margin-bottom: 20px;

color: #dee2e6;

}

.no-appointments h3 {

margin-bottom: 10px;

color: #495057;

}

.no-appointments p {

font-size: 16px;

}

/\* Botão de cancelar específico \*/

.btn-cancel {

background: #dc3545 !important;

color: white !important;

border-color: #dc3545 !important;

}

.btn-cancel:hover {

background: #c82333 !important;

border-color: #bd2130 !important;

}

/\* Informações de cancelamento \*/

.cancel-info {

background: #f8f9fa;

padding: 10px;

border-radius: 5px;

margin-top: 10px;

border-left: 4px solid #dc3545;

}

.cancel-info p {

margin: 0;

color: #721c24;

font-size: 14px;

}

/\* Filtros \*/

.filter-controls {

display: flex;

gap: 15px;

flex-wrap: wrap;

align-items: center;

margin-bottom: 20px;

}

.filter-group {

display: flex;

align-items: center;

gap: 8px;

}

.filter-select {

padding: 8px 12px;

border: 1px solid #ddd;

border-radius: 5px;

background: white;

font-size: 14px;

}

.btn-small {

padding: 6px 12px;

font-size: 13px;

}

/\* Animação para feedback visual \*/

@keyframes pulse {

0% { transform: scale(1); }

50% { transform: scale(1.02); }

100% { transform: scale(1); }

}

.appointment-card {

transition: all 0.3s ease;

}

/\* Estilo para texto desabilitado \*/

.text-muted {

color: #6c757d !important;

font-style: italic;

padding: 8px 12px;

}

/\* Garantir que os botões desabilitados tenham aparência correta \*/

.appointment-card.cancelled .btn,

.appointment-card.expired .btn {

opacity: 0.6;

cursor: not-allowed;

pointer-events: none;

}

.appointment-card.cancelled .btn:hover,

.appointment-card.expired .btn:hover {

transform: none;

box-shadow: none;

}

`;

// Adicione o CSS ao documento

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

style.textContent = additionalCSS;

document.head.appendChild(style);

# Arquivo: public/js/atualizarStatus.js

const db = require('./database/db');

async function atualizarStatusAgendamentos() {

const query = `

UPDATE agendamento\_simples

SET status = 'fora\_prazo'

WHERE data\_hora < NOW()

AND status IN ('pendente', 'confirmado')

`;

db.query(query, (err, result) => {

if (err) {

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

return;

}

console.log(`Status atualizados: ${result.affectedRows} agendamentos`);

});

}

// Executar imediatamente e depois a cada hora

atualizarStatusAgendamentos();

setInterval(atualizarStatusAgendamentos, 60 \* 60 \* 1000);

# Arquivo: public/js/auth.js

// auth.js - Sistema unificado de autenticação

class AuthManager {

constructor() {

this.init();

}

init() {

this.checkLoginStatus();

this.setupEventListeners();

}

checkLoginStatus() {

const isLoggedIn = localStorage.getItem('isLoggedIn') === 'true';

const userData = localStorage.getItem('userData');

this.updateUI(isLoggedIn, userData ? JSON.parse(userData) : null);

}

updateUI(isLoggedIn, userData = null) {

const loginStatus = document.getElementById('login-status');

const userDisplayName = document.getElementById('user-display-name');

const userEmail = document.getElementById('user-email');

const dropdownMenu = document.getElementById('dropdown-menu');

const userDropdown = document.getElementById('user-dropdown');

if (loginStatus) {

if (isLoggedIn && userData) {

loginStatus.textContent = userData.nome || userData.email.split('@')[0];

if (userDisplayName) userDisplayName.textContent = userData.nome || 'Usuário';

if (userEmail) userEmail.textContent = userData.email;

if (userDropdown) userDropdown.classList.add('logged-in');

} else {

loginStatus.textContent = 'Login';

if (userDropdown) userDropdown.classList.remove('logged-in');

}

}

}

setupEventListeners() {

// Login modal

const loginBtn = document.getElementById('login-btn');

const loginModal = document.getElementById('login-modal');

const closeModal = document.querySelector('.close-modal');

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

if (loginBtn && loginModal) {

loginBtn.addEventListener('click', (e) => {

e.preventDefault();

this.showModal(loginModal);

});

}

if (closeModal) {

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

this.hideModal(loginModal);

});

}

if (loginForm) {

loginForm.addEventListener('submit', (e) => {

e.preventDefault();

this.handleLogin(loginForm);

});

}

// Logout

const logoutBtn = document.getElementById('logout-btn');

if (logoutBtn) {

logoutBtn.addEventListener('click', (e) => {

e.preventDefault();

this.handleLogout();

});

}

// Close modal when clicking outside

document.addEventListener('click', (e) => {

if (loginModal && e.target === loginModal) {

this.hideModal(loginModal);

}

});

}

showModal(modal) {

if (modal) {

modal.style.display = 'block';

document.body.style.overflow = 'hidden';

}

}

hideModal(modal) {

if (modal) {

modal.style.display = 'none';

document.body.style.overflow = 'auto';

}

}

async handleLogin(form) {

const email = form.querySelector('#login-email').value;

const password = form.querySelector('#login-password').value;

const rememberMe = form.querySelector('#remember-me')?.checked;

try {

const response = await fetch('http://localhost:3000/api/auth/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ email, senha: password })

});

const data = await response.json();

if (data.success) {

localStorage.setItem('isLoggedIn', 'true');

localStorage.setItem('userData', JSON.stringify(data.user));

localStorage.setItem('token', data.token);

localStorage.setItem('user', JSON.stringify(data.user));

localStorage.setItem('userId', data.user.id); // Adicionado ID do usuário

if (rememberMe) {

localStorage.setItem('rememberedEmail', email);

} else {

localStorage.removeItem('rememberedEmail');

}

this.updateUI(true, data.user);

this.hideModal(document.getElementById('login-modal'));

// Show success message

this.showToast('Login realizado com sucesso!', 'success');

// Redirect or reload if needed

setTimeout(() => {

window.location.reload();

}, 1000);

} else {

this.showToast(data.message || 'Erro no login', 'error');

}

} catch (error) {

console.error('Login error:', error);

this.showToast('Erro ao conectar com o servidor', 'error');

}

}

handleLogout() {

localStorage.removeItem('isLoggedIn');

localStorage.removeItem('userData');

localStorage.removeItem('token');

this.updateUI(false);

this.showToast('Logout realizado com sucesso', 'success');

setTimeout(() => {

window.location.reload();

}, 1000);

}

showToast(message, type = 'info') {

// Create toast element if it doesn't exist

let toast = document.getElementById('auth-toast');

if (!toast) {

toast = document.createElement('div');

toast.id = 'auth-toast';

toast.style.cssText = `

position: fixed;

top: 20px;

right: 20px;

padding: 12px 20px;

border-radius: 4px;

color: white;

z-index: 10000;

transition: all 0.3s ease;

transform: translateX(100%);

`;

document.body.appendChild(toast);

}

// Set style based on type

const colors = {

success: '#4CAF50',

error: '#f44336',

warning: '#ff9800',

info: '#2196F3'

};

toast.style.backgroundColor = colors[type] || colors.info;

toast.textContent = message;

// Show toast

toast.style.transform = 'translateX(0)';

// Hide after 3 seconds

setTimeout(() => {

toast.style.transform = 'translateX(100%)';

}, 3000);

}

// auth.js - Adicione esta função na classe AuthManager

isUserLoggedIn() {

return localStorage.getItem('isLoggedIn') === 'true';

}

getUserData() {

const userData = localStorage.getItem('userData');

return userData ? JSON.parse(userData) : null;

}

getUserId() {

return localStorage.getItem('userId');

}

}

// Initialize auth manager when DOM is loaded

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

window.authManager = new AuthManager();

});

# Arquivo: public/js/cadastro.js

document.getElementById('register-form').addEventListener('submit', async (event) => {

event.preventDefault();

const nome = document.getElementById('register-name').value;

const email = document.getElementById('register-email').value;

const senha = document.getElementById('register-password').value;

const tipo = document.getElementById('register-type').value;

if (!nome || !email || !senha || !tipo) {

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

return;

}

try {

const response = await fetch('http://localhost:3000/api/auth/register', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({ nome, email, senha, tipo })

});

const data = await response.json();

if (response.ok) {

alert('Cadastro realizado com sucesso! Agora faça login.');

window.location.href = 'login.html';

} else {

alert(data.erro || 'Falha no cadastro');

}

} catch (error) {

console.error('Erro ao registrar usuário:', error);

alert('Erro ao tentar cadastrar. Tente novamente.');

}

});

# Arquivo: public/js/login.js

document.getElementById('login-form').addEventListener('submit', async (event) => {

event.preventDefault();

const email = document.getElementById('login-email').value;

const senha = document.getElementById('login-password').value;

if (!email || !senha) {

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

return;

}

try {

const response = await fetch('http://localhost:3000/api/auth/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({ email, senha })

});

const data = await response.json();

if (response.ok) {

localStorage.setItem('token', data.token);

localStorage.setItem('user', JSON.stringify(data.usuario));

window.location.href = 'agenda.html';

} else {

alert(data.erro || 'Falha no login');

}

} catch (error) {

console.error('Erro ao realizar login:', error);

alert('Erro ao tentar fazer login. Tente novamente.');

}

});

# Arquivo: public/js/main.js

// main.js - Arquivo JavaScript unificado para o site OilSmart (Adaptado para BD)

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

// =============================================

// FUNCIONALIDADES GLOBAIS (todas as páginas)

// =============================================

// 1. Menu Hamburguer

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

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

if (hamburger && nav) {

hamburger.addEventListener('click', function () {

nav.classList.toggle('active');

hamburger.classList.toggle('active');

document.body.style.overflow = nav.classList.contains('active') ? 'hidden' : 'auto';

});

// Fecha o menu quando um link é clicado (para mobile)

const navLinks = document.querySelectorAll('.nav\_\_list a');

navLinks.forEach(link => {

link.addEventListener('click', function () {

if (window.innerWidth <= 992) {

hamburger.classList.remove('active');

nav.classList.remove('active');

document.body.style.overflow = 'auto';

}

});

});

// Fecha o menu ao redimensionar para desktop

window.addEventListener('resize', function () {

if (window.innerWidth > 992) {

hamburger.classList.remove('active');

nav.classList.remove('active');

document.body.style.overflow = 'auto';

}

});

}

// =============================================

// CONTROLE DE USUÁRIO E LOGIN

// =============================================

// Elementos do DOM

const userDropdown = document.getElementById('user-dropdown');

const loginStatus = document.getElementById('login-status');

const logoutBtn = document.getElementById('logout-btn');

const dropdownMenu = document.getElementById('dropdown-menu');

const userDisplayName = document.getElementById('user-display-name');

const userEmail = document.getElementById('user-email');

const loginBtn = document.getElementById('login-btn');

// Verificar status de login

async function checkLoginStatus() {

const isLoggedIn = localStorage.getItem('isLoggedIn') === 'true';

const userData = JSON.parse(localStorage.getItem('user') || '{}');

const userId = localStorage.getItem('userId');

if (isLoggedIn && userData && userId) {

try {

// Verificar no servidor se o login ainda é válido

const response = await fetch('http://localhost:3000/api/auth/check-login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ userId: userId })

});

if (response.ok) {

const result = await response.json();

if (result.loggedIn) {

updateUserInterface(result.user);

userDropdown.classList.add('user-logged-in');

return;

}

}

} catch (error) {

console.error('Erro ao verificar login:', error);

// Se der erro, usa os dados locais

if (userData.nome || userData.email) {

updateUserInterface(userData);

userDropdown.classList.add('user-logged-in');

return;

}

}

}

// Se não estiver logado ou dados inválidos

userDropdown.classList.remove('user-logged-in');

loginStatus.textContent = 'Login';

}

// Atualizar interface com dados do usuário

function updateUserInterface(user) {

console.log('Atualizando interface com dados:', user); // Debug

// Atualizar o status de login no header

if (loginStatus) {

if (user.nome) {

loginStatus.textContent = user.nome.split(' ')[0]; // Primeiro nome apenas

} else if (user.email) {

const username = user.email.split('@')[0];

loginStatus.textContent = username;

} else {

loginStatus.textContent = 'Minha Conta';

}

}

// Atualizar nome no dropdown

if (userDisplayName) {

if (user.nome) {

userDisplayName.textContent = user.nome;

} else if (user.email) {

userDisplayName.textContent = user.email.split('@')[0];

} else {

userDisplayName.textContent = 'Usuário';

}

}

// Atualizar email no dropdown

if (userEmail) {

if (user.email) {

userEmail.textContent = user.email;

} else {

userEmail.textContent = 'email@exemplo.com';

}

}

// Marcar como logado

if (userDropdown) {

userDropdown.classList.add('user-logged-in');

}

}

// Logout

if (logoutBtn) {

logoutBtn.addEventListener('click', async function (e) {

e.preventDefault();

try {

// Chamar API de logout

const response = await fetch('http://localhost:3000/api/auth/logout', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

}

});

// Mesmo se a API falhar, faz logout localmente

localStorage.removeItem('isLoggedIn');

localStorage.removeItem('user');

localStorage.removeItem('userId');

showToast('Você foi deslogado com sucesso', 'success');

// Atualizar interface

userDropdown.classList.remove('user-logged-in');

loginStatus.textContent = 'Login';

// Fechar dropdown

dropdownMenu.style.display = 'none';

// Redirecionar para home após 1 segundo

setTimeout(() => {

window.location.href = 'index.html';

}, 1000);

} catch (error) {

console.error('Erro no logout:', error);

// Logout local mesmo com erro

localStorage.removeItem('isLoggedIn');

localStorage.removeItem('user');

localStorage.removeItem('userId');

userDropdown.classList.remove('user-logged-in');

loginStatus.textContent = 'Login';

dropdownMenu.style.display = 'none';

showToast('Desconectado', 'info');

}

});

}

// Abrir dropdown

if (userDropdown) {

userDropdown.addEventListener('click', function (e) {

e.stopPropagation();

const isLoggedIn = localStorage.getItem('isLoggedIn') === 'true';

const userData = JSON.parse(localStorage.getItem('userData') || localStorage.getItem('user') || '{}');

if (!isLoggedIn || !userData.email) {

// Se não estiver logado, abrir modal de login em vez de redirecionar

if (loginModal) {

showModal(loginModal);

} else {

window.location.href = '/html/login.html';

}

return;

}

// Se estiver logado, abrir/fechar dropdown

const isVisible = dropdownMenu.classList.contains('show') || dropdownMenu.style.display === 'block';

if (isVisible) {

dropdownMenu.classList.remove('show');

dropdownMenu.style.display = 'none';

} else {

// Atualizar dados do usuário antes de mostrar o dropdown

updateUserInterface(userData);

dropdownMenu.classList.add('show');

dropdownMenu.style.display = 'block';

}

});

}

// Fechar dropdown ao clicar fora

document.addEventListener('click', function (e) {

if (dropdownMenu && userDropdown && !userDropdown.contains(e.target)) {

dropdownMenu.style.display = 'none';

}

});

// Verificar login ao carregar a página

checkLoginStatus();

// Verificar login também quando a página ganha foco

window.addEventListener('focus', checkLoginStatus);

// Verificar login periodicamente (a cada 30 segundos)

setInterval(checkLoginStatus, 30000);

// 3. Modal de Login (comum a todas as páginas)

const loginModal = document.getElementById('login-modal');

const closeModalBtns = document.querySelectorAll('.close-modal');

const forgotModal = document.getElementById('forgot-modal');

const forgotPasswordLink = document.getElementById('forgot-password');

const backToLoginLink = document.getElementById('back-to-login');

const registerLink = document.getElementById('register-link');

const googleLoginBtn = document.getElementById('google-login');

const loginForm = document.getElementById('login-form') || document.querySelector('.login-form');

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

// Funções auxiliares para modais

function showModal(modal) {

if (modal) {

modal.classList.add('show');

document.body.style.overflow = 'hidden';

}

}

function hideModal(modal) {

if (modal) {

modal.classList.remove('show');

document.body.style.overflow = '';

}

}

// Abrir modal de login

if (loginBtn && loginModal) {

loginBtn.addEventListener('click', function (e) {

e.preventDefault();

showModal(loginModal);

});

}

// Mostrar modal de recuperação de senha

if (forgotPasswordLink) {

forgotPasswordLink.addEventListener('click', function (e) {

e.preventDefault();

hideModal(loginModal);

showModal(forgotModal);

});

}

// Voltar para o login

if (backToLoginLink) {

backToLoginLink.addEventListener('click', function (e) {

e.preventDefault();

hideModal(forgotModal);

showModal(loginModal);

});

}

// Fechar modais

if (closeModalBtns) {

closeModalBtns.forEach(btn => {

btn.addEventListener('click', function () {

const modal = this.closest('.modal');

hideModal(modal);

});

});

}

// Fechar ao clicar fora

window.addEventListener('click', function (e) {

if (e.target.classList.contains('modal')) {

hideModal(e.target);

}

});

// Mostrar/esconder senha

document.querySelectorAll('.toggle-password').forEach(btn => {

btn.addEventListener('click', function () {

const input = this.previousElementSibling || document.getElementById('login-password');

if (input) {

const type = input.getAttribute('type') === 'password' ? 'text' : 'password';

input.setAttribute('type', type);

this.innerHTML = type === 'password' ? '<i class="far fa-eye"></i>' : '<i class="far fa-eye-slash"></i>';

}

});

});

// Validação do formulário de login

if (loginForm) {

loginForm.addEventListener('submit', async function (e) {

e.preventDefault();

const email = this.querySelector('#login-email').value.trim();

const password = this.querySelector('#login-password').value.trim();

const rememberMe = this.querySelector('#remember-me') ? this.querySelector('#remember-me').checked : false;

// Reset erros

resetErrors();

// Validação

let isValid = true;

if (!email) {

showError('email-error', 'Por favor, insira seu e-mail');

isValid = false;

} else if (!validateEmail(email)) {

showError('email-error', 'Por favor, insira um e-mail válido');

isValid = false;

}

if (!password) {

showError('password-error', 'Por favor, insira sua senha');

isValid = false;

} else if (password.length < 6) {

showError('password-error', 'A senha deve ter pelo menos 6 caracteres');

isValid = false;

}

if (isValid) {

await loginWithDatabase(email, password, rememberMe);

}

});

}

// Validação do formulário de recuperação

if (forgotForm) {

forgotForm.addEventListener('submit', async function (e) {

e.preventDefault();

const email = document.getElementById('forgot-email').value.trim();

// Reset erros

document.getElementById('forgot-email-error').textContent = '';

// Validação

if (!email) {

showError('forgot-email-error', 'Por favor, insira seu e-mail');

} else if (!validateEmail(email)) {

showError('forgot-email-error', 'Por favor, insira um e-mail válido');

} else {

await resetPasswordWithDatabase(email);

}

});

}

// Login com Google

if (googleLoginBtn) {

googleLoginBtn.addEventListener('click', function (e) {

e.preventDefault();

showToast('Redirecionando para o login com Google...', 'info');

});

}

// Link de registro

if (registerLink) {

registerLink.addEventListener('click', function (e) {

e.preventDefault();

window.location.href = '/html/cadastro.html';

});

}

// Funções auxiliares para formulários

function resetErrors() {

document.querySelectorAll('.error-message').forEach(el => {

el.textContent = '';

});

}

function showError(id, message) {

const element = document.getElementById(id);

if (element) {

element.textContent = message;

}

}

function validateEmail(email) {

const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

return re.test(email);

}

// Login com banco de dados

async function loginWithDatabase(email, password, rememberMe) {

const submitBtn = loginForm.querySelector('button[type="submit"]');

const originalText = submitBtn.innerHTML;

// Mostrar loading

submitBtn.innerHTML = '<i class="fas fa-spinner fa-spin"></i> Entrando...';

submitBtn.disabled = true;

try {

// Enviar dados para o servidor

const response = await fetch('/api/auth/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

email: email,

senha: password

})

});

const result = await response.json();

if (result.success) {

// Login bem-sucedido

const userName = result.user.nome || email.split('@')[0];

// Armazenar informações do usuário

localStorage.setItem('isLoggedIn', 'true');

localStorage.setItem('user', JSON.stringify(result.user));

localStorage.setItem('userId', result.user.id);

// Lembrar email se solicitado

if (rememberMe) {

localStorage.setItem('rememberedEmail', email);

} else {

localStorage.removeItem('rememberedEmail');

}

showToast(`Bem-vindo(a), ${userName}!`, 'success');

// Fechar modal

hideModal(loginModal);

// Atualizar status do usuário

checkLoginStatus();

// Verificar se é a página de agenda para redirecionar

if (window.location.pathname.includes('agenda.html')) {

// Mostrar modal de confirmação na página de agenda

const confirmationModal = document.getElementById('confirmation-modal');

if (confirmationModal) {

showModal(confirmationModal);

setTimeout(() => {

window.location.href = 'agenda.html';

}, 2000);

}

}

} else {

showToast(result.message, 'error');

}

} catch (error) {

console.error('Erro no login:', error);

showToast('Erro ao conectar com o servidor', 'error');

} finally {

// Restaurar botão

submitBtn.innerHTML = originalText;

submitBtn.disabled = false;

}

}

async function resetPasswordWithDatabase(email) {

const forgotButton = document.getElementById('forgot-button');

const originalText = forgotButton.innerHTML;

// Mostrar loading

forgotButton.innerHTML = '<i class="fas fa-spinner fa-spin"></i> Enviando...';

forgotButton.disabled = true;

try {

// Simular envio de recuperação (implementar API real posteriormente)

await new Promise(resolve => setTimeout(resolve, 1500));

showToast(`Link de recuperação enviado para ${email}`, 'success');

hideModal(forgotModal);

} catch (error) {

console.error('Erro ao enviar recuperação:', error);

showToast('Erro ao enviar email de recuperação', 'error');

} finally {

// Restaurar botão

forgotButton.innerHTML = originalText;

forgotButton.disabled = false;

}

}

// Mostrar toast notification

function showToast(message, type = 'success') {

// Remove toast existente se houver

const existingToast = document.querySelector('.toast');

if (existingToast) existingToast.remove();

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

toast.className = `toast ${type}`;

toast.textContent = message;

document.body.appendChild(toast);

setTimeout(() => {

toast.classList.add('show');

}, 10);

setTimeout(() => {

toast.classList.remove('show');

setTimeout(() => {

toast.remove();

}, 300);

}, 3000);

}

// Adicionar estilo para o toast se não existir

if (!document.getElementById('toast-style')) {

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

style.id = 'toast-style';

style.textContent = `

.toast {

position: fixed;

bottom: 20px;

left: 50%;

transform: translateX(-50%);

padding: 12px 24px;

border-radius: 8px;

color: white;

font-weight: 500;

z-index: 1002;

opacity: 0;

transition: opacity 0.3s, transform 0.3s;

}

.toast.show {

opacity: 1;

transform: translateX(-50%) translateY(-10px);

}

.toast.success {

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

}

.toast.error {

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

}

.toast.warning {

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

}

.toast.info {

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

}

`;

document.head.appendChild(style);

}

// Verificar se tem e-mail lembrado

const rememberedEmail = localStorage.getItem('rememberedEmail');

if (rememberedEmail && document.getElementById('login-email')) {

document.getElementById('login-email').value = rememberedEmail;

if (document.getElementById('remember-me')) {

document.getElementById('remember-me').checked = true;

}

}

// 3. PÁGINA SOBRE (sobre.html)

if (window.location.pathname.includes('sobre.html')) {

// Configurar animações para elementos da página

const setupAnimations = function () {

const elements = document.querySelectorAll('.milestone, .mv-card, .diferencial-card, .testimonial');

elements.forEach(element => {

element.style.opacity = '0';

element.style.transform = 'translateY(20px)';

element.style.transition = 'opacity 0.6s ease, transform 0.6s ease';

});

const animateOnScroll = function () {

elements.forEach(element => {

const elementPosition = element.getBoundingClientRect().top;

const screenPosition = window.innerHeight / 1.3;

if (elementPosition < screenPosition) {

element.style.opacity = '1';

element.style.transform = 'translateY(0)';

}

});

};

window.addEventListener('scroll', animateOnScroll);

animateOnScroll(); // Executa uma vez ao carregar

};

setupAnimations();

}

});

// Funções globais que podem ser chamadas de qualquer lugar

function validateCPF(cpf) {

cpf = cpf.replace(/\D/g, '');

if (cpf.length !== 11 || /^(\d)\1+$/.test(cpf)) {

return false;

}

let soma = 0;

for (let i = 0; i < 9; i++) {

soma += parseInt(cpf.charAt(i)) \* (10 - i);

}

let resto = (soma \* 10) % 11;

if (resto === 10 || resto === 11) {

resto = 0;

}

if (resto !== parseInt(cpf.charAt(9))) {

return false;

}

soma = 0;

for (let i = 0; i < 10; i++) {

soma += parseInt(cpf.charAt(i)) \* (11 - i);

}

resto = (soma \* 10) % 11;

if (resto === 10 || resto === 11) {

resto = 0;

}

return resto === parseInt(cpf.charAt(10));

}

// Carrossel do Hero Banner

function initHeroCarousel() {

const slides = document.querySelectorAll('.carousel-slide');

const prevBtn = document.querySelector('.carousel-prev');

const nextBtn = document.querySelector('.carousel-next');

let currentSlide = 0;

let autoSlideInterval;

const slideIntervalTime = 5000; // 5 segundos

// Iniciar carrossel automático

function startAutoSlide() {

autoSlideInterval = setInterval(() => {

nextSlide();

}, slideIntervalTime);

}

// Parar carrossel automático

function stopAutoSlide() {

clearInterval(autoSlideInterval);

}

// Mostrar slide

function showSlide(n) {

slides[currentSlide].classList.remove('active');

currentSlide = (n + slides.length) % slides.length;

slides[currentSlide].classList.add('active');

}

// Slide anterior

function prevSlide() {

showSlide(currentSlide - 1);

stopAutoSlide();

startAutoSlide();

}

// Próximo slide

function nextSlide() {

showSlide(currentSlide + 1);

}

// Event listeners

if (prevBtn && nextBtn) {

prevBtn.addEventListener('click', prevSlide);

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

nextSlide();

stopAutoSlide();

startAutoSlide();

});

// Iniciar

showSlide(0);

startAutoSlide();

// Pausar quando o mouse estiver sobre o carrossel

const heroBanner = document.querySelector('.hero-banner');

if (heroBanner) {

heroBanner.addEventListener('mouseenter', stopAutoSlide);

heroBanner.addEventListener('mouseleave', startAutoSlide);

}

}

}

// Inicializar quando o DOM estiver carregado

document.addEventListener('DOMContentLoaded', initHeroCarousel);

// Funções auxiliares para formulários (usadas na página de serviços)

function resetFormErrors(form) {

form.querySelectorAll('.error-message').forEach(el => {

el.textContent = '';

});

}

function showError(form, fieldId, message) {

const errorElement = form.querySelector(`#${fieldId}-error`);

if (errorElement) {

errorElement.textContent = message;

}

}

function clearError(form, fieldId) {

const errorElement = form.querySelector(`#${fieldId}-error`);

if (errorElement) {

errorElement.textContent = '';

}

}

// Mover ícone de login para dentro do menu hamburger em mobile - VERSÃO CORRIGIDA

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

const actionIcons = document.querySelector('.action-icons');

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

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

const navbar = document.querySelector('.navbar');

// Cria uma cópia do action-icons para o mobile

let mobileActionIcons = null;

function handleMobileMenu() {

if (window.innerWidth <= 768) {

// Mobile - cria cópia dentro do nav se não existir

if (actionIcons && nav && !mobileActionIcons) {

mobileActionIcons = actionIcons.cloneNode(true);

mobileActionIcons.classList.add('mobile-actions');

nav.appendChild(mobileActionIcons);

// Adiciona eventos aos elementos clonados

setupClonedActionIcons(mobileActionIcons);

}

// Controla visibilidade baseado no menu

if (mobileActionIcons) {

mobileActionIcons.style.display = nav.classList.contains('active') ? 'flex' : 'none';

}

// Esconde o original no mobile

if (actionIcons) {

actionIcons.style.display = 'none';

}

} else {

// Desktop - mostra o original e remove a cópia do mobile

if (actionIcons) {

actionIcons.style.display = 'flex';

}

if (mobileActionIcons) {

mobileActionIcons.remove();

mobileActionIcons = null;

}

}

}

// Configura eventos para os elementos clonados

function setupClonedActionIcons(clonedIcons) {

const dropdown = clonedIcons.querySelector('.user-dropdown');

const loginBtn = clonedIcons.querySelector('#login-btn');

if (dropdown) {

dropdown.addEventListener('click', function(e) {

e.stopPropagation();

const dropdownMenu = this.querySelector('.dropdown-menu');

if (dropdownMenu) {

dropdownMenu.style.display = dropdownMenu.style.display === 'block' ? 'none' : 'block';

}

});

}

}

// Controla visibilidade quando hamburger é clicado

if (hamburger && nav) {

hamburger.addEventListener('click', function() {

setTimeout(() => {

if (window.innerWidth <= 768 && mobileActionIcons) {

mobileActionIcons.style.display = nav.classList.contains('active') ? 'flex' : 'none';

}

}, 10);

});

}

// Fecha dropdown ao clicar fora (para versão mobile também)

document.addEventListener('click', function(e) {

if (mobileActionIcons) {

const dropdownMenu = mobileActionIcons.querySelector('.dropdown-menu');

const userDropdown = mobileActionIcons.querySelector('.user-dropdown');

if (dropdownMenu && userDropdown && !userDropdown.contains(e.target)) {

dropdownMenu.style.display = 'none';

}

}

});

// Executa inicialmente e no resize

handleMobileMenu();

window.addEventListener('resize', handleMobileMenu);

});

# Arquivo: public/js/servicos.js

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

// ==================== ELEMENTOS DO DOM ====================

const brandSelect = document.getElementById("vehicle-brand");

const modelSelect = document.getElementById("vehicle-model");

const yearSelect = document.getElementById("vehicle-year");

const mileageInput = document.getElementById("vehicle-mileage");

const vehicleForm = document.getElementById("vehicle-form");

const locationInput = document.getElementById("location-input");

const locationButton = document.getElementById("location-button");

const userLocationBtn = document.getElementById("user-location-btn");

const progressSteps = document.querySelectorAll(".progress-step");

const selectedWorkshopDiv = document.getElementById("selected-workshop");

const scheduleDate = document.getElementById("schedule-date");

// Elementos do modal de login

const loginBtn = document.getElementById("login-btn");

const modal = document.getElementById("login-modal");

const closeModal = document.querySelector(".close-modal");

const loginForm = document.querySelector(".login-form");

const togglePassword = document.querySelector(".toggle-password");

const passwordInput = document.getElementById("login-password");

const hamburgerMenu = document.getElementById("hamburger-menu");

const mobileNav = document.getElementById("nav");

// ==================== VARIÁVEIS GLOBAIS ====================

let userLocation = null;

let markers = [];

let currentStep = 1;

let selectedWorkshop = null;

let userVehicle = null;

let map = null;

let addressSearchTimeout = null;

let cachedAddresses = {};

let locationHistory = [];

let selectedProducts = {

oil: true,

filter: true

};

// ==================== FUNÇÕES DE HISTÓRICO DE LOCALIZAÇÃO ====================

function checkUserLoggedIn() {

const token = localStorage.getItem('token');

const userData = localStorage.getItem('user');

if (!token || !userData) {

window.location.href = 'login.html?redirect=servicos.html';

return false;

}

return true;

}

// Carrega o histórico de localizações do localStorage

function loadLocationHistory() {

try {

const savedHistory = localStorage.getItem('oilSmartLocationHistory');

if (savedHistory) {

locationHistory = JSON.parse(savedHistory);

}

} catch (error) {

console.error("Erro ao carregar histórico de localizações:", error);

locationHistory = [];

}

}

// Salva uma nova localização no histórico

function saveToLocationHistory(locationData) {

// Evita duplicatas

const exists = locationHistory.some(item =>

item.display\_name === locationData.display\_name

);

if (!exists) {

locationHistory.unshift(locationData);

// Mantém apenas as 5 localizações mais recentes

locationHistory = locationHistory.slice(0, 5);

// Salva no localStorage

try {

localStorage.setItem('oilSmartLocationHistory', JSON.stringify(locationHistory));

} catch (error) {

console.error("Erro ao salvar histórico de localizações:", error);

}

}

}

// Exibe sugestões de localização baseadas no histórico

function showLocationSuggestions() {

const suggestionsContainer = document.getElementById("location-suggestions");

if (!suggestionsContainer || locationHistory.length === 0) return;

suggestionsContainer.innerHTML = '<h4>Localizações recentes:</h4>';

locationHistory.forEach(location => {

const suggestionItem = document.createElement("div");

suggestionItem.className = "location-suggestion";

suggestionItem.innerHTML = `

<i class="fas fa-history"></i>

<span>${location.display\_name.split(',')[0]}</span>

<small>${location.display\_name.split(',').slice(1, 3).join(',').trim()}</small>

`;

suggestionItem.addEventListener("click", () => {

locationInput.value = location.display\_name;

suggestionsContainer.style.display = "none";

searchByAddress(location);

});

suggestionsContainer.appendChild(suggestionItem);

});

suggestionsContainer.style.display = "block";

}

// ==================== FUNÇÕES DE RESPONSIVIDADE ====================

// Ajusta o layout para diferentes tamanhos de tela

function handleResponsiveLayout() {

// Menu hamburguer para mobile

if (window.innerWidth <= 992) {

if (hamburgerMenu && mobileNav) {

hamburgerMenu.style.display = "block";

mobileNav.style.display = "none";

}

} else {

if (hamburgerMenu && mobileNav) {

hamburgerMenu.style.display = "none";

mobileNav.style.display = "flex";

}

}

// Ajuste do mapa para mobile

const mapContainer = document.getElementById("map");

if (mapContainer) {

if (window.innerWidth <= 768) {

mapContainer.style.height = "300px";

} else {

mapContainer.style.height = "400px";

}

if (map) {

setTimeout(() => map.invalidateSize(), 100);

}

}

// Ajuste dos cards de recomendação

const recommendationContainer = document.querySelector(".recommendation-container");

if (recommendationContainer) {

if (window.innerWidth <= 768) {

recommendationContainer.style.flexDirection = "column";

} else {

recommendationContainer.style.flexDirection = "row";

}

}

// Ajuste dos cards de oficina

const workshopItems = document.querySelectorAll(".workshop-item");

workshopItems.forEach(item => {

if (window.innerWidth <= 768) {

item.style.flexDirection = "column";

item.querySelector(".btn").style.width = "100%";

item.querySelector(".btn").style.marginTop = "10px";

} else {

item.style.flexDirection = "row";

item.querySelector(".btn").style.width = "auto";

item.querySelector(".btn").style.marginTop = "0";

}

});

// Ajuste do formulário de agendamento

const scheduleForm = document.getElementById("schedule-form");

if (scheduleForm) {

if (window.innerWidth <= 768) {

scheduleForm.style.padding = "15px";

} else {

scheduleForm.style.padding = "20px";

}

}

}

// Menu hamburguer para mobile

if (hamburgerMenu && mobileNav) {

hamburgerMenu.addEventListener("click", function() {

mobileNav.style.display = mobileNav.style.display === "block" ? "none" : "block";

document.body.style.overflow = mobileNav.style.display === "block" ? "hidden" : "auto";

});

// Fecha o menu ao clicar em um link

const navLinks = mobileNav.querySelectorAll("a");

navLinks.forEach(link => {

link.addEventListener("click", function() {

if (window.innerWidth <= 992) {

mobileNav.style.display = "none";

document.body.style.overflow = "auto";

}

});

});

}

// ==================== INICIALIZAÇÃO DO MAPA (RESPONSIVA) ====================

function initMap() {

const mapContainer = document.getElementById("map");

const mapHeight = window.innerWidth <= 768 ? "300px" : "400px";

mapContainer.style.height = mapHeight;

map = L.map("map").setView([-23.5505, -46.6333], 13);

L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {

attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'

}).addTo(map);

// Função para buscar as oficinas do seu backend e adicionar marcadores no mapa

async function fetchAndDisplayWorkshops() {

try {

const response = await fetch('/api/oficina');

const data = await response.json();

if (data.success && data.data) {

data.data.forEach(oficina => {

const lat = oficina.lat;

const lng = oficina.lng;

// Certifica-se de que a latitude e longitude são válidas

if (lat && lng) {

const marker = L.marker([lat, lng]).addTo(map);

// Adiciona um popup com o nome e endereço da oficina

marker.bindPopup(`

<b>${oficina.nome}</b><br>

${oficina.endereco}, ${oficina.cidade} - ${oficina.estado}

`);

}

});

}

} catch (error) {

console.error('Erro ao buscar as oficinas:', error);

}

}

// Redimensiona o mapa quando a janela é redimensionada

window.addEventListener("resize", function() {

const newHeight = window.innerWidth <= 768 ? "300px" : "400px";

mapContainer.style.height = newHeight;

setTimeout(() => map.invalidateSize(), 100);

});

}

// ==================== FUNÇÕES DE CARREGAMENTO DE DADOS ====================

// Carrega as marcas de veículos (versão responsiva)

async function populateBrands() {

showLoading(true);

brandSelect.innerHTML = '<option value="">Selecione a marca</option>';

try {

const response = await fetch("/api/marcas");

if (!response.ok) throw new Error("Falha ao carregar marcas");

const brands = await response.json();

brands.forEach(brand => {

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

option.value = brand.id;

option.textContent = brand.nome;

brandSelect.appendChild(option);

});

} catch (error) {

console.error("Erro ao carregar marcas:", error);

showToast("Erro ao carregar marcas. Tente recarregar a página.", "error");

} finally {

showLoading(false);

}

}

// Carrega os modelos baseado na marca selecionada

async function populateModels(brandId) {

showLoading(true);

modelSelect.innerHTML = '<option value="">Selecione o modelo</option>';

modelSelect.disabled = true;

yearSelect.innerHTML = '<option value="">Selecione o ano</option>';

yearSelect.disabled = true;

if (brandId) {

try {

const response = await fetch(`/api/modelos?marca\_id=${brandId}`);

if (!response.ok) throw new Error("Falha ao carregar modelos");

const models = await response.json();

models.forEach(model => {

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

option.value = model.id;

option.textContent = model.nome;

modelSelect.appendChild(option);

});

modelSelect.disabled = false;

} catch (error) {

console.error("Erro ao carregar modelos:", error);

showToast("Erro ao carregar modelos para esta marca.", "error");

} finally {

showLoading(false);

}

}

}

// Carrega os anos baseado no modelo selecionado

async function populateYears(modelId) {

showLoading(true);

yearSelect.innerHTML = '<option value="">Selecione o ano</option>';

yearSelect.disabled = true;

if (modelId) {

try {

const response = await fetch(`/api/modelo\_anos?modelo\_id=${modelId}`);

if (!response.ok) throw new Error("Falha ao carregar anos");

const years = await response.json();

years.forEach(year => {

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

option.value = year.id;

option.textContent = year.ano;

yearSelect.appendChild(option);

});

yearSelect.disabled = false;

} catch (error) {

console.error("Erro ao carregar anos:", error);

showToast("Erro ao carregar anos para este modelo.", "error");

} finally {

showLoading(false);

}

}

}

// ==================== FUNÇÕES DE RECOMENDAÇÃO ====================

// Obtém recomendação de óleo e filtro

// Obtém recomendação de óleo e filtro

async function getOilRecommendation(modeloAnoId) {

showLoading(true);

try {

const response = await fetch(`/api/recomendacoes?modelo\_ano\_id=${modeloAnoId}`);

const result = await response.json();

if (!result.success) {

showToast(result.message, "error");

return null;

}

// Armazena os dados da recomendação para uso posterior

sessionStorage.setItem('oilRecommendation', JSON.stringify(result.data));

return result.data;

} catch (error) {

console.error("Erro ao obter recomendação:", error);

showToast("Erro de comunicação ao obter recomendação.", "error");

return null;

} finally {

showLoading(false);

}

}

function showRecommendations(recommendation) {

const oilContainer = document.getElementById("recommended-oil");

const filterContainer = document.getElementById("recommended-filter");

// LIMPEZA COMPLETA dos containers

oilContainer.innerHTML = '';

filterContainer.innerHTML = '';

if (!recommendation) {

oilContainer.innerHTML = `

<div class="no-recommendation">

<i class="fas fa-exclamation-triangle"></i>

<p>Nenhuma recomendação de óleo disponível</p>

</div>

<div class="product-selection">

<label class="checkbox-container">

<input type="checkbox" id="select-oil" disabled>

<span class="checkmark"></span>

Incluir óleo no serviço

</label>

</div>

`;

filterContainer.innerHTML = `

<div class="no-recommendation">

<i class="fas fa-exclamation-triangle"></i>

<p>Nenhuma recomendação de filtro disponível</p>

</div>

<div class="product-selection">

<label class="checkbox-container">

<input type="checkbox" id="select-filter" disabled>

<span class="checkmark"></span>

Incluir filtro no serviço

</label>

</div>

`;

return;

}

// Exibe recomendação de óleo

if (recommendation.oleo) {

const oil = recommendation.oleo;

oilContainer.innerHTML = `

<div class="product-card-content">

<div class="product-image">

<img src="../img/oil-default.jpg" alt="${oil.nome}" loading="lazy">

</div>

<div class="product-details">

<h4>${oil.nome}</h4>

<div class="product-specs">

<p><strong>Tipo:</strong> ${oil.tipo || 'N/A'}</p>

<p><strong>Viscosidade:</strong> ${oil.viscosidade || 'N/A'}</p>

<p><strong>Especificação:</strong> ${oil.especificacao || 'N/A'}</p>

</div>

${oil.preco ? `

<div class="product-price">

<strong>Preço estimado:</strong>

<span class="price">R$ ${(oil.preco || 0).toFixed(2)}</span>

</div>

` : ''}

</div>

</div>

<div class="product-selection">

<label class="checkbox-container">

<input type="checkbox" id="select-oil" checked>

<span class="checkmark"></span>

Incluir óleo no serviço

</label>

</div>

`;

} else {

oilContainer.innerHTML = `

<div class="no-recommendation">

<i class="fas fa-exclamation-triangle"></i>

<p>Nenhuma recomendação de óleo disponível</p>

</div>

<div class="product-selection">

<label class="checkbox-container">

<input type="checkbox" id="select-oil" disabled>

<span class="checkmark"></span>

Incluir óleo no serviço

</label>

</div>

`;

}

// Exibe recomendação de filtro (CÓDIGO CORRIGIDO)

// Exibe recomendação de filtro (VERSÃO FINAL CORRIGIDA)

if (recommendation.filtro) {

const filter = recommendation.filtro;

const precoFiltro = filter.preco || 0;

const precoFormatado = precoFiltro > 0 ? `R$ ${parseFloat(precoFiltro).toFixed(2)}` : 'Preço não disponível';

filterContainer.innerHTML = `

<div class="product-card-content">

<div class="product-image">

<img src="../img/oil-filter.jpg" alt="${filter.nome}" loading="lazy">

</div>

<div class="product-details">

<h4>${filter.nome}</h4>

<div class="product-specs">

<p><strong>Tipo:</strong> ${filter.tipo || 'N/A'}</p>

<p><strong>Compatibilidade:</strong> ${filter.compatibilidade\_modelo || 'N/A'}</p>

<p><strong>Marca:</strong> ${filter.marca || 'N/A'}</p>

</div>

<div class="product-price">

<strong>Preço estimado:</strong>

<span class="price">${precoFormatado}</span>

</div>

</div>

</div>

<div class="product-selection">

<label class="checkbox-container">

<input type="checkbox" id="select-filter" checked>

<span class="checkmark"></span>

Incluir filtro no serviço

</label>

</div>

`;

} else {

filterContainer.innerHTML = `

<div class="no-recommendation">

<i class="fas fa-exclamation-triangle"></i>

<p>Nenhuma recomendação de filtro disponível</p>

</div>

<div class="product-selection">

<label class="checkbox-container">

<input type="checkbox" id="select-filter" disabled>

<span class="checkmark"></span>

Incluir filtro no serviço

</label>

</div>

`;

}

// Adiciona event listeners aos checkboxes

const oilCheckbox = document.getElementById("select-oil");

const filterCheckbox = document.getElementById("select-filter");

if (oilCheckbox) {

oilCheckbox.addEventListener("change", updateProductSelection);

}

if (filterCheckbox) {

filterCheckbox.addEventListener("change", updateProductSelection);

}

// Inicializa a seleção

updateProductSelection();

}

// ==================== FUNÇÕES DE SELEÇÃO DE PRODUTOS ====================

// Atualiza a seleção de produtos

// Atualiza a seleção de produtos

function updateProductSelection() {

const oilCheckbox = document.getElementById("select-oil");

const filterCheckbox = document.getElementById("select-filter");

selectedProducts.oil = oilCheckbox ? oilCheckbox.checked : false;

selectedProducts.filter = filterCheckbox ? filterCheckbox.checked : false;

// Salva no sessionStorage

sessionStorage.setItem('selectedProducts', JSON.stringify(selectedProducts));

updateSelectionSummary();

}

// Atualiza o resumo da seleção

function updateSelectionSummary() {

const summaryElement = document.getElementById("selected-items");

const selections = [];

if (selectedProducts.oil) {

selections.push("Óleo");

}

if (selectedProducts.filter) {

selections.push("Filtro");

}

if (selections.length === 0) {

summaryElement.textContent = "Nenhum produto selecionado";

summaryElement.style.color = "#e63946";

} else {

summaryElement.textContent = selections.join(" + ");

summaryElement.style.color = "#2a9d8f";

}

}

// Valida a seleção antes de avançar

function validateSelection() {

if (!selectedProducts.oil && !selectedProducts.filter) {

showToast("Selecione pelo menos um produto para continuar", "error");

return;

}

// Armazena a seleção para uso posterior

sessionStorage.setItem('selectedProducts', JSON.stringify(selectedProducts));

goToStep(3);

}

// Recupera a seleção de produtos do sessionStorage

function getSelectedProducts() {

const saved = sessionStorage.getItem('selectedProducts');

return saved ? JSON.parse(saved) : { oil: true, filter: true };

}

// ==================== FUNÇÕES DE LOCALIZAÇÃO E OFICINAS ====================

// Obtém localização do usuário via GPS (versão responsiva)

async function getUserLocation() {

if (!navigator.geolocation) {

showToast("Seu navegador não suporta geolocalização", "error");

return;

}

showLoading(true);

try {

const position = await new Promise((resolve, reject) => {

navigator.geolocation.getCurrentPosition(resolve, reject, {

enableHighAccuracy: true,

timeout: 10000,

maximumAge: 0

});

});

userLocation = {

lat: position.coords.latitude,

lng: position.coords.longitude

};

// Adiciona marcador no mapa

addUserMarker(userLocation.lat, userLocation.lng, "Sua localização");

// Centraliza o mapa

map.setView([userLocation.lat, userLocation.lng], 14);

// Busca oficinas próximas

await searchNearbyWorkshops(userLocation.lat, userLocation.lng);

showToast("Localização obtida com sucesso!");

} catch (error) {

console.error("Erro ao obter localização:", error);

showToast("Não foi possível obter sua localização", "error");

} finally {

showLoading(false);

}

}

// Busca por endereço com autocomplete (OTIMIZADO)

async function searchAddress(query) {

if (!query || query.length < 3) return [];

// Verifica se já temos resultados em cache

if (cachedAddresses[query]) {

return cachedAddresses[query];

}

try {

const response = await fetch(`https://nominatim.openstreetmap.org/search?format=json&q=${encodeURIComponent(query)}&countrycodes=br&limit=5&addressdetails=1`);

if (!response.ok) throw new Error("Erro ao buscar endereços");

const data = await response.json();

// Armazena no cache

cachedAddresses[query] = data;

return data;

} catch (error) {

console.error("Erro na busca de endereços:", error);

return [];

}

}

// Exibe sugestões de endereço (versão responsiva)

function showAddressSuggestions(addresses) {

const resultsContainer = document.getElementById("autocomplete-results");

resultsContainer.innerHTML = '';

if (!addresses || addresses.length === 0) {

resultsContainer.classList.remove("active");

return;

}

addresses.forEach(address => {

const item = document.createElement("div");

item.className = "autocomplete-item";

item.innerHTML = `

<div class="main-text">${address.display\_name.split(',')[0]}</div>

<div class="secondary-text">${address.display\_name.split(',').slice(1).join(',').trim()}</div>

`;

item.addEventListener("click", () => {

locationInput.value = address.display\_name;

resultsContainer.classList.remove("active");

searchByAddress();

});

resultsContainer.appendChild(item);

});

resultsContainer.classList.add("active");

}

// Busca por endereço usando Nominatim (OTIMIZADO)

async function searchByAddress() {

const address = locationInput.value.trim();

if (!address) {

showToast("Digite um endereço válido", "warning");

return;

}

showLoading(true);

try {

// Verifica se já temos esse endereço em cache

let locationData;

if (cachedAddresses[address] && cachedAddresses[address].length > 0) {

locationData = cachedAddresses[address][0];

} else {

// Busca o endereço usando Nominatim

const response = await fetch(`https://nominatim.openstreetmap.org/search?format=json&q=${encodeURIComponent(address)}&countrycodes=br&limit=1&addressdetails=1`);

if (!response.ok) throw new Error("Erro ao buscar endereço");

const data = await response.json();

if (!data || data.length === 0) {

throw new Error("Endereço não encontrado");

}

locationData = data[0];

}

const location = {

lat: parseFloat(locationData.lat),

lng: parseFloat(locationData.lon)

};

userLocation = location;

// Adiciona marcador no mapa

addUserMarker(location.lat, location.lng, locationData.display\_name);

// Centraliza o mapa

map.setView([location.lat, location.lng], 14);

// Busca oficinas próximas

await searchNearbyWorkshops(location.lat, location.lng);

showToast(`Oficinas próximas a: ${locationData.display\_name}`);

} catch (error) {

console.error("Erro na busca por endereço:", error);

showToast(error.message, "error");

} finally {

showLoading(false);

}

}

// Adiciona marcador da localização do usuário

function addUserMarker(lat, lng, popupText) {

// Remove marcadores antigos

markers.filter(m => m.options.icon?.options?.className === "user-marker")

.forEach(m => map.removeLayer(m));

const userIcon = L.divIcon({

className: "user-marker",

html: '<div style="background: #1d3557; width: 20px; height: 20px; border-radius: 50%; border: 3px solid white; box-shadow: 0 2px 5px rgba(0,0,0,0.3);"></div>',

iconSize: [20, 20],

iconAnchor: [10, 20]

});

const marker = L.marker([lat, lng], { icon: userIcon })

.addTo(map)

.bindPopup(popupText)

.openPopup();

markers.push(marker);

}

// Busca oficinas próximas

// Busca oficinas próximas e calcula distâncias

async function searchNearbyWorkshops(lat, lng) {

showLoading(true);

try {

console.log('Buscando oficinas do banco de dados...');

const response = await fetch(`/api/oficina`);

if (!response.ok) {

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

}

const data = await response.json();

console.log('Oficinas encontradas:', data);

if (!data.success || !data.data || data.data.length === 0) {

showToast("Nenhuma oficina encontrada", "info");

displayWorkshops([]);

return;

}

// Filtra apenas oficinas com coordenadas válidas

const oficinasValidas = data.data.filter(workshop => {

const latNum = parseFloat(workshop.lat);

const lngNum = parseFloat(workshop.lng);

return !isNaN(latNum) && !isNaN(lngNum);

});

if (oficinasValidas.length === 0) {

showToast("Oficinas encontradas, mas sem coordenadas válidas", "warning");

return;

}

// Calcula distância para cada oficina

const oficinasComDistancia = oficinasValidas.map(workshop => {

const workshopLat = parseFloat(workshop.lat);

const workshopLng = parseFloat(workshop.lng);

// Calcula distância usando fórmula de Haversine

const distancia = calcularDistancia(lat, lng, workshopLat, workshopLng);

return {

...workshop,

distancia: distancia

};

});

// Ordena por distância (mais próximo primeiro)

oficinasComDistancia.sort((a, b) => a.distancia - b.distancia);

console.log('Oficinas ordenadas por distância:', oficinasComDistancia);

displayWorkshops(oficinasComDistancia);

} catch (error) {

console.error("Erro ao buscar oficinas:", error);

showToast("Erro ao carregar oficinas. Tente novamente.", "error");

} finally {

showLoading(false);

}

}

// Função para calcular distância usando fórmula de Haversine

function calcularDistancia(lat1, lon1, lat2, lon2) {

const R = 6371; // Raio da Terra em km

const dLat = deg2rad(lat2 - lat1);

const dLon = deg2rad(lon2 - lon1);

const a =

Math.sin(dLat/2) \* Math.sin(dLat/2) +

Math.cos(deg2rad(lat1)) \* Math.cos(deg2rad(lat2)) \*

Math.sin(dLon/2) \* Math.sin(dLon/2);

const c = 2 \* Math.atan2(Math.sqrt(a), Math.sqrt(1-a));

const distancia = R \* c; // Distância em km

return distancia;

}

function deg2rad(deg) {

return deg \* (Math.PI/180);

}

// Exibe oficinas no mapa e na lista (versão responsiva)

function displayWorkshops(workshops) {

const mapContainer = document.querySelector(".map-container");

// Cria ou atualiza o container de lista de oficinas

let listContainer = document.querySelector(".workshop-list-container");

if (!listContainer) {

listContainer = document.createElement("div");

listContainer.className = "workshop-list-container";

mapContainer.parentNode.insertBefore(listContainer, mapContainer.nextSibling);

}

listContainer.innerHTML = "";

// Limpa marcadores antigos (exceto o do usuário)

markers.filter(m => m.options.icon?.options?.className !== "user-marker")

.forEach(m => map.removeLayer(m));

markers = markers.filter(m => m.options.icon?.options?.className === "user-marker");

if (!workshops || workshops.length === 0) {

listContainer.innerHTML = "<p class='no-workshops'>Nenhuma oficina encontrada</p>";

return;

}

// Ícone das oficinas

const workshopIcon = L.divIcon({

className: "workshop-marker",

html: '<div style="background: #e63946; width: 30px; height: 30px; border-radius: 50%; border: 3px solid white; box-shadow: 0 2px 5px rgba(0,0,0,0.3);"></div>',

iconSize: [30, 30],

iconAnchor: [15, 30],

popupAnchor: [0, -30]

});

// Adiciona cada oficina no mapa e na lista

workshops.forEach(workshop => {

const lat = parseFloat(workshop.lat);

const lng = parseFloat(workshop.lng);

// Verifica se as coordenadas são válidas

if (isNaN(lat) || isNaN(lng)) {

console.warn('Coordenadas inválidas para oficina:', workshop);

return;

}

// Formata a distância

const distanciaFormatada = workshop.distancia < 1

? `${(workshop.distancia \* 1000).toFixed(0)} m`

: `${workshop.distancia.toFixed(1)} km`;

// Marcador no mapa

const marker = L.marker([lat, lng], { icon: workshopIcon })

.addTo(map)

.bindPopup(`

<div class="workshop-popup">

<h4>${workshop.nome}</h4>

<p><i class="fas fa-map-marker-alt"></i> ${workshop.endereco}, ${workshop.cidade}/${workshop.estado}</p>

<p><i class="fas fa-phone"></i> ${workshop.telefone || 'Não informado'}</p>

<p><i class="fas fa-route"></i> ${distanciaFormatada}</p>

<button onclick="selectWorkshop(${workshop.id})" class="select-workshop-btn">

Selecionar

</button>

</div>

`);

markers.push(marker);

// Item na lista

const workshopItem = document.createElement("div");

workshopItem.className = "workshop-item";

workshopItem.dataset.id = workshop.id;

workshopItem.innerHTML = `

<div class="workshop-info">

<h4>${workshop.nome}</h4>

<p class="distance"><i class="fas fa-route"></i> ${distanciaFormatada}</p>

<p class="address"><i class="fas fa-map-marker-alt"></i> ${workshop.endereco}, ${workshop.cidade}/${workshop.estado}</p>

<p class="phone"><i class="fas fa-phone"></i> ${workshop.telefone || 'Não informado'}</p>

<p class="hours"><i class="fas fa-clock"></i> ${workshop.horario\_abertura} - ${workshop.horario\_fechamento}</p>

</div>

<button class="btn select-btn" onclick="selectWorkshop(${workshop.id})">Selecionar</button>

`;

listContainer.appendChild(workshopItem);

});

// Ajusta o zoom para mostrar todos os marcadores

if (markers.length > 0) {

const bounds = L.latLngBounds(markers.map(m => m.getLatLng()));

map.fitBounds(bounds.pad(0.2));

}

}

// ==================== FUNÇÕES DE OFICINAS ====================

// Seleciona uma oficina

window.selectWorkshop = async function(workshopId) {

showLoading(true);

try {

// Busca os detalhes completos da oficina

const response = await fetch(`/api/oficina/${workshopId}`);

if (!response.ok) throw new Error("Erro ao carregar oficina");

const workshopData = await response.json();

selectedWorkshop = workshopData;

// CONVERTE COORDENADAS PARA NÚMERO

if (selectedWorkshop.lat) selectedWorkshop.lat = parseFloat(selectedWorkshop.lat);

if (selectedWorkshop.lng) selectedWorkshop.lng = parseFloat(selectedWorkshop.lng);

// Destaca a oficina selecionada

document.querySelectorAll(".workshop-item").forEach(item => {

item.classList.remove("selected");

if (parseInt(item.dataset.id) === parseInt(workshopId)) {

item.classList.add("selected");

item.style.animation = "pulse 0.5s ease";

setTimeout(() => {

item.style.animation = "";

}, 500);

}

});

// Centraliza o mapa na oficina selecionada

if (selectedWorkshop.lat && selectedWorkshop.lng) {

map.setView([selectedWorkshop.lat, selectedWorkshop.lng], 15);

}

showToast(`Oficina "${workshopData.nome}" selecionada!`);

return true;

} catch (error) {

console.error("Erro ao selecionar oficina:", error);

showToast("Erro ao selecionar a oficina", "error");

return false;

} finally {

showLoading(false);

}

};

// Mostra detalhes da oficina selecionada (versão responsiva)

function showSelectedWorkshop(workshop) {

selectedWorkshopDiv.innerHTML = `

<div class="selected-workshop-card">

<h3>${workshop.nome}</h3>

<p><i class="fas fa-map-marker-alt"></i> ${workshop.endereco}, ${workshop.cidade}/${workshop.estado}</p>

<p><i class="fas fa-phone"></i> ${workshop.telefone || 'Não informado'}</p>

<p><i class="fas fa-clock"></i> ${workshop.horario\_abertura} - ${workshop.horario\_fechamento}</p>

<p>${workshop.dias\_funcionamento}</p>

</div>

`;

updateAvailableTimeSlots(workshop);

}

// Atualiza os horários disponíveis para agendamento

function updateAvailableTimeSlots(workshop) {

const timeSelect = document.getElementById("schedule-time");

timeSelect.innerHTML = '<option value="" disabled selected>Selecione um horário</option>';

// Gera horários disponíveis (simulação)

const slots = generateTimeSlots(

workshop.horario\_abertura || "08:00",

workshop.horario\_fechamento || "18:00",

30

);

slots.forEach(time => {

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

option.value = time;

option.textContent = time;

timeSelect.appendChild(option);

});

}

// Gera intervalos de tempo entre abertura e fechamento

function generateTimeSlots(start, end, interval) {

const slots = [];

const [startHour, startMinute] = start.split(":").map(Number);

const [endHour, endMinute] = end.split(":").map(Number);

let currentHour = startHour;

let currentMinute = startMinute;

while (currentHour < endHour || (currentHour === endHour && currentMinute <= endMinute)) {

const time = `${currentHour.toString().padStart(2, '0')}:${currentMinute.toString().padStart(2, '0')}`;

slots.push(time);

currentMinute += interval;

if (currentMinute >= 60) {

currentHour += Math.floor(currentMinute / 60);

currentMinute = currentMinute % 60;

}

}

return slots;

}

// ==================== FUNÇÕES DE AGENDAMENTO ====================

// Processa o agendamento e salva no banco - VERSÃO COMPLETA E CORRIGIDA

async function processScheduling() {

console.log('Iniciando processScheduling...');

// Verificar se o usuário está logado

const userData = localStorage.getItem('user');

if (!userData) {

showToast("Você precisa fazer login para agendar um serviço", "error");

setTimeout(() => {

window.location.href = 'login.html?redirect=servicos.html';

}, 2000);

return false;

}

const user = JSON.parse(userData);

const usuario\_id = user.id;

if (!usuario\_id) {

showToast("Erro: ID do usuário não encontrado", "error");

return false;

}

// Obter dados do formulário

const scheduleDateValue = document.getElementById("schedule-date").value;

const scheduleTime = document.getElementById("schedule-time").value;

const customerName = document.getElementById("customer-name").value.trim();

const customerCpf = document.getElementById("customer-cpf").value.trim();

const customerPhone = document.getElementById("customer-phone").value.trim();

const customerEmail = document.getElementById("customer-email").value.trim();

// Validação básica dos campos obrigatórios

if (!scheduleDateValue || !scheduleTime || !customerName || !customerCpf || !customerPhone || !customerEmail) {

showToast("Preencha todos os campos obrigatórios", "error");

return false;

}

if (!validateCPF(customerCpf)) {

showToast("CPF inválido", "error");

return false;

}

if (!validateEmail(customerEmail)) {

showToast("E-mail inválido", "error");

return false;

}

if (!selectedWorkshop) {

showToast("Selecione uma oficina", "error");

return false;

}

// Verificar se há dados do veículo

const userVehicle = JSON.parse(sessionStorage.getItem('userVehicle') || '{}');

if (!userVehicle.modelo\_ano\_id) {

showToast("Dados do veículo não encontrados. Por favor, volte ao passo 1.", "error");

return false;

}

showLoading(true);

try {

// Obter produtos selecionados

const selectedProducts = getSelectedProducts();

const oilRecommendation = JSON.parse(sessionStorage.getItem('oilRecommendation') || '{}');

// Calcular total do serviço

let totalOil = 0;

let totalFilter = 0;

if (selectedProducts.oil && oilRecommendation.oleo && oilRecommendation.oleo.preco) {

totalOil = parseFloat(oilRecommendation.oleo.preco) || 0;

}

if (selectedProducts.filter && oilRecommendation.filtro && oilRecommendation.filtro.preco) {

totalFilter = parseFloat(oilRecommendation.filtro.preco) || 0;

}

const totalService = totalOil + totalFilter;

// Preparar descrição dos serviços

const servicosArray = [];

if (selectedProducts.oil && oilRecommendation.oleo) {

servicosArray.push(`Troca de Óleo: ${oilRecommendation.oleo.nome} - R$ ${totalOil.toFixed(2)}`);

}

if (selectedProducts.filter && oilRecommendation.filtro) {

servicosArray.push(`Troca de Filtro: ${oilRecommendation.filtro.nome} - R$ ${totalFilter.toFixed(2)}`);

}

// CORREÇÃO: Verificar se é array antes de fazer join

if (Array.isArray(servicosArray) && servicosArray.length > 0) {

servicos = servicosArray.join(' | ');

} else if (typeof servicosArray === 'string') {

// Se já for string, usar diretamente (remover caracteres indesejados)

servicos = servicosArray.replace(/[\[\]"]/g, '').trim();

} else {

servicos = 'Serviços não especificados';

}

// Formatar data e hora para o formato do banco

const dataHora = `${scheduleDateValue} ${scheduleTime}:00`;

// Gerar protocolo único

const protocolo = `OIL${Date.now().toString().slice(-8)}`;

// Preparar dados para envio - ESTRUTURA COMPLETA E CORRETA

const agendamentoData = {

protocolo: protocolo,

data\_hora: dataHora,

oficina\_nome: selectedWorkshop.nome,

oficina\_endereco: `${selectedWorkshop.endereco}, ${selectedWorkshop.cidade}/${selectedWorkshop.estado}`,

oficina\_telefone: selectedWorkshop.telefone || 'Não informado',

veiculo: `${userVehicle.marca || ''} ${userVehicle.modelo || ''} ${userVehicle.ano || ''}`.trim(),

servicos: servicosArray.join(' | '), // Converter array para string

total\_servico: totalService,

cliente\_nome: customerName,

cliente\_cpf: customerCpf.replace(/\D/g, ''), // Apenas números

cliente\_telefone: customerPhone,

cliente\_email: customerEmail,

usuario\_id: usuario\_id

};

console.log('Dados do agendamento para salvar:', agendamentoData);

// Salvar no banco usando a rota correta

const result = await salvarAgendamentoNoBanco(agendamentoData);

if (result.success) {

// Armazenar dados para exibição na confirmação

sessionStorage.setItem('codigoConfirmacao', result.codigo\_confirmacao || protocolo);

sessionStorage.setItem('agendamentoId', result.agendamento\_id);

// Salvar dados do cliente para exibição

const customerData = {

name: customerName,

cpf: customerCpf,

phone: customerPhone,

email: customerEmail

};

sessionStorage.setItem('customerData', JSON.stringify(customerData));

// Salvar dados da oficina

sessionStorage.setItem('selectedWorkshop', JSON.stringify(selectedWorkshop));

// Salvar dados do serviço

const serviceData = {

servicos: servicosArray,

total: totalService,

data: dataHora

};

sessionStorage.setItem('serviceData', JSON.stringify(serviceData));

showToast('Agendamento salvo com sucesso!', 'success');

return true;

} else {

throw new Error(result.message || 'Erro ao salvar agendamento');

}

} catch (error) {

console.error('Erro no agendamento:', error);

showToast(error.message || 'Erro ao processar agendamento', 'error');

return false;

} finally {

showLoading(false);

}

}

// Função para salvar agendamento no banco - VERSÃO CORRIGIDA

async function salvarAgendamentoNoBanco(agendamentoData) {

console.log('🔄 Tentando salvar agendamento no banco...');

console.log('Dados enviados:', JSON.stringify(agendamentoData, null, 2));

try {

const response = await fetch('/api/agendamento\_simples', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify(agendamentoData)

});

console.log('Status da resposta:', response.status);

// Tentar ler a resposta mesmo em caso de erro

const responseText = await response.text();

console.log('Resposta do servidor:', responseText);

let result;

try {

result = JSON.parse(responseText);

} catch (e) {

throw new Error(`Resposta inválida do servidor: ${responseText}`);

}

if (!response.ok) {

throw new Error(result.message || `Erro HTTP ${response.status}`);

}

if (!result.success) {

throw new Error(result.message || 'Erro ao salvar agendamento');

}

console.log('✅ Agendamento salvo com sucesso!', result);

return result;

} catch (error) {

console.error('❌ Erro ao salvar agendamento no banco:', error);

throw error;

}

}

// Função auxiliar para verificar se o usuário está logado

function checkUserLoggedIn() {

const token = localStorage.getItem('token');

const userData = localStorage.getItem('user');

if (!token || !userData) {

showToast("Você precisa fazer login para agendar um serviço", "error");

setTimeout(() => {

window.location.href = 'login.html?redirect=servicos.html';

}, 2000);

return false;

}

return true;

}

// Função para mostrar detalhes da confirmação (atualizada)

function showConfirmationDetails() {

const codigoConfirmacao = sessionStorage.getItem('codigoConfirmacao') || `OS${Date.now().toString().slice(-8)}`;

const customerData = JSON.parse(sessionStorage.getItem('customerData') || '{}');

const userVehicle = JSON.parse(sessionStorage.getItem('userVehicle') || '{}');

const selectedWorkshop = JSON.parse(sessionStorage.getItem('selectedWorkshop') || '{}');

const serviceData = JSON.parse(sessionStorage.getItem('serviceData') || '{}');

// Formatar data para exibição

let dataFormatada = '';

if (serviceData.data) {

const data = new Date(serviceData.data);

dataFormatada = data.toLocaleDateString('pt-BR') + ' às ' + data.toLocaleTimeString('pt-BR', {hour: '2-digit', minute: '2-digit'});

}

const confirmationDetails = document.querySelector(".confirmation-details");

let html = `

<div class="confirmation-section">

<h4>📋 Informações do Agendamento</h4>

<p><strong>Protocolo:</strong> ${codigoConfirmacao}</p>

<p><strong>Data e Horário:</strong> ${dataFormatada}</p>

</div>

<div class="confirmation-section">

<h4>🏢 Oficina</h4>

<p><strong>Nome:</strong> ${selectedWorkshop.nome || 'Não informado'}</p>

<p><strong>Endereço:</strong> ${selectedWorkshop.endereco || 'Não informado'}, ${selectedWorkshop.cidade || ''}/${selectedWorkshop.estado || ''}</p>

<p><strong>Telefone:</strong> ${selectedWorkshop.telefone || 'Não informado'}</p>

</div>

`;

// Informações do veículo

if (userVehicle.marca || userVehicle.modelo || userVehicle.ano) {

html += `

<div class="confirmation-section">

<h4>🚗 Veículo</h4>

<p><strong>Modelo:</strong> ${userVehicle.marca || ''} ${userVehicle.modelo || ''} ${userVehicle.ano || ''}</p>

${userVehicle.quilometragem ? `<p><strong>Quilometragem:</strong> ${userVehicle.quilometragem} km</p>` : ''}

</div>

`;

}

// Serviços selecionados

if (serviceData.servicos && serviceData.servicos.length > 0) {

html += `

<div class="confirmation-section">

<h4>🔧 Serviços</h4>

${Array.isArray(serviceData.servicos) ?

serviceData.servicos.map(servico => `<p>• ${servico}</p>`).join('') :

`<p>• ${serviceData.servicos}</p>`}

${serviceData.total ? `<p><strong>Total:</strong> R$ ${serviceData.total.toFixed(2)}</p>` : ''}

</div>

`;

}

// Informações do cliente

html += `

<div class="confirmation-section">

<h4>👤 Cliente</h4>

<p><strong>Nome:</strong> ${customerData.name || 'Não informado'}</p>

<p><strong>CPF:</strong> ${customerData.cpf || 'Não informado'}</p>

<p><strong>Telefone:</strong> ${customerData.phone || 'Não informado'}</p>

<p><strong>E-mail:</strong> ${customerData.email || 'Não informado'}</p>

</div>

`;

confirmationDetails.innerHTML = html;

}

// Função para validar o passo 3 (atualizada)

async function validateStep3() {

const scheduleDateValue = document.getElementById("schedule-date").value;

const scheduleTime = document.getElementById("schedule-time").value;

const customerName = document.getElementById("customer-name").value.trim();

const customerCpf = document.getElementById("customer-cpf").value.trim();

const customerPhone = document.getElementById("customer-phone").value.trim();

const customerEmail = document.getElementById("customer-email").value.trim();

if (!scheduleDateValue || !scheduleTime || !customerName || !customerCpf || !customerPhone || !customerEmail) {

showToast("Preencha todos os campos obrigatórios", "error");

return false;

}

if (!validateCPF(customerCpf)) {

showToast("CPF inválido", "error");

return false;

}

if (!validateEmail(customerEmail)) {

showToast("E-mail inválido", "error");

return false;

}

if (!selectedWorkshop) {

showToast("Selecione uma oficina", "error");

return false;

}

// Verificar se a data é válida (não pode ser no passado)

const selectedDate = new Date(scheduleDateValue + 'T' + scheduleTime);

const now = new Date();

if (selectedDate < now) {

showToast("Selecione uma data e horário futuros", "error");

return false;

}

return true;

}

// Mostra detalhes do agendamento confirmado

function showConfirmationDetails() {

const codigoConfirmacao = sessionStorage.getItem('codigoConfirmacao') || `OS${Date.now().toString().slice(-8)}`;

const customerData = JSON.parse(sessionStorage.getItem('customerData') || '{}');

const userVehicle = JSON.parse(sessionStorage.getItem('userVehicle') || '{}');

const selectedProducts = getSelectedProducts();

const oilRecommendation = JSON.parse(sessionStorage.getItem('oilRecommendation') || '{}');

const scheduleDateValue = document.getElementById("schedule-date").value;

const scheduleTime = document.getElementById("schedule-time").value;

const formattedDate = formatDateForDisplay(scheduleDateValue);

// Cálculo dos totais

let totalOil = 0;

let totalFilter = 0;

if (selectedProducts.oil && oilRecommendation.oleo && oilRecommendation.oleo.preco) {

totalOil = parseFloat(oilRecommendation.oleo.preco) || 0;

}

if (selectedProducts.filter && oilRecommendation.filtro && oilRecommendation.filtro.preco) {

totalFilter = parseFloat(oilRecommendation.filtro.preco) || 0;

}

const totalService = totalOil + totalFilter;

const confirmationDetails = document.querySelector(".confirmation-details");

let html = `

<p><strong>Protocolo:</strong> ${codigoConfirmacao}</p>

<p><strong>Data:</strong> ${formattedDate} às ${scheduleTime}</p>

<p><strong>Oficina:</strong> ${selectedWorkshop.nome}</p>

<p><strong>Endereço:</strong> ${selectedWorkshop.endereco}, ${selectedWorkshop.cidade}/${selectedWorkshop.estado}</p>

<p><strong>Telefone:</strong> ${selectedWorkshop.telefone || 'Não informado'}</p>

`;

// Adiciona informações do veículo

if (userVehicle.marca && userVehicle.modelo && userVehicle.ano) {

html += `<p><strong>Veículo:</strong> ${userVehicle.marca} ${userVehicle.modelo} ${userVehicle.ano}</p>`;

}

if (userVehicle.quilometragem) {

html += `<p><strong>Quilometragem:</strong> ${userVehicle.quilometragem} km</p>`;

}

// Adiciona serviços selecionados

html += `<p><strong>Serviços selecionados:</strong></p>`;

if (selectedProducts.oil && oilRecommendation.oleo) {

html += `<p>• Troca de óleo: ${oilRecommendation.oleo.nome} - R$ ${totalOil.toFixed(2)}</p>`;

}

if (selectedProducts.filter && oilRecommendation.filtro) {

html += `<p>• Troca de filtro: ${oilRecommendation.filtro.nome} - R$ ${totalFilter.toFixed(2)}</p>`;

}

// Adiciona total

html += `<p><strong>Total do serviço:</strong> R$ ${totalService.toFixed(2)}</p>`;

// Adiciona informações do cliente

html += `

<p><strong>Cliente:</strong> ${customerData.name || 'Não informado'}</p>

<p><strong>CPF:</strong> ${customerData.cpf || 'Não informado'}</p>

<p><strong>Contato:</strong> ${customerData.phone || 'Não informado'}</p>

<p><strong>E-mail:</strong> ${customerData.email || 'Não informado'}</p>

`;

confirmationDetails.innerHTML = html;

}

// ==================== FUNÇÕES DE NAVEGAÇÃO ====================

// Navega entre os passos do formulário - VERSÃO ATUALIZADA

window.goToStep = async function(step) {

if (step > currentStep) {

// Validação antes de avançar

if (step === 2 && !validateStep1()) return;

if (step === 3 && !await validateStep2()) return;

if (step === 4) {

// No passo 4, processar e salvar o agendamento

const success = await processScheduling();

if (!success) return;

showConfirmationDetails();

}

}

// Atualiza a interface

document.querySelectorAll(".service-step").forEach(s => s.classList.remove("active"));

document.getElementById(`step${step}`).classList.add("active");

progressSteps.forEach((p, index) => {

p.classList.toggle("active", index < step);

});

currentStep = step;

// Rolagem suave para o passo atual

document.getElementById(`step${step}`).scrollIntoView({

behavior: "smooth",

block: "start"

});

};

// Validação do passo 1 (Dados do veículo)

function validateStep1() {

const isValid = brandSelect.value && modelSelect.value && yearSelect.value;

if (!isValid) {

showToast("Selecione marca, modelo e ano do veículo", "error");

return false;

}

// Valida quilometragem apenas se preenchida

if (mileageInput.value.trim() !== "") {

if (isNaN(mileageInput.value) || mileageInput.value < 0) {

showToast("Quilometragem inválida", "error");

return false;

}

}

return true;

}

// Validação do passo 2 (Seleção de oficina) - CORRIGIDA

// Validação do passo 2 (Seleção de produtos)

async function validateStep2() {

const selectedProducts = getSelectedProducts();

// Verifica se pelo menos um produto foi selecionado

if (!selectedProducts.oil && !selectedProducts.filter) {

showToast("Selecione pelo menos um produto (óleo ou filtro) para continuar", "error");

// Adiciona animação para chamar atenção nos cards

const oilCard = document.getElementById("recommended-oil");

const filterCard = document.getElementById("recommended-filter");

if (oilCard) {

oilCard.style.animation = "shake 0.5s ease";

setTimeout(() => { oilCard.style.animation = ""; }, 500);

}

if (filterCard) {

filterCard.style.animation = "shake 0.5s ease";

setTimeout(() => { filterCard.style.animation = ""; }, 500);

}

return false;

}

// Verifica se a oficina foi selecionada

if (!selectedWorkshop) {

showToast("Selecione uma oficina para continuar", "error");

const workshopList = document.querySelector(".workshop-list-container");

if (workshopList) {

workshopList.style.animation = "shake 0.5s ease";

setTimeout(() => {

workshopList.style.animation = "";

}, 500);

}

return false;

}

// Verifica se a oficina selecionada ainda está válida

try {

const response = await fetch(`/api/oficina/${selectedWorkshop.id}`);

if (!response.ok) throw new Error("Oficina não disponível");

// Atualiza os dados da oficina

selectedWorkshop = await response.json();

return true;

} catch (error) {

console.error("Erro ao validar oficina:", error);

showToast("A oficina selecionada não está mais disponível. Por favor, selecione outra.", "error");

selectedWorkshop = null;

return false;

}

}

// Validação do passo 3 (Dados do agendamento)

async function validateStep3() {

const scheduleDateValue = document.getElementById("schedule-date").value;

const scheduleTime = document.getElementById("schedule-time").value;

const customerName = document.getElementById("customer-name").value.trim();

const customerCpf = document.getElementById("customer-cpf").value.trim();

const customerPhone = document.getElementById("customer-phone").value.trim();

const customerEmail = document.getElementById("customer-email").value.trim();

if (!scheduleDateValue || !scheduleTime || !customerName || !customerCpf || !customerPhone || !customerEmail) {

showToast("Preencha todos os campos obrigatórios", "error");

return false;

}

if (!validateCPF(customerCpf)) {

showToast("CPF inválido", "error");

return false;

}

if (!validateEmail(customerEmail)) {

showToast("E-mail inválido", "error");

return false;

}

// SALVAR DADOS DO CLIENTE ANTES DE AVANÇAR

const customerData = {

name: customerName,

cpf: customerCpf,

phone: customerPhone,

email: customerEmail

};

sessionStorage.setItem('customerData', JSON.stringify(customerData));

// Mostrar os detalhes da confirmação

showConfirmationDetails();

return true;

}

// Função para debug detalhado antes do salvamento

function debugAgendamentoCompleto() {

const userVehicle = JSON.parse(sessionStorage.getItem('userVehicle') || '{}');

const selectedProducts = getSelectedProducts();

const oilRecommendation = JSON.parse(sessionStorage.getItem('oilRecommendation') || '{}');

const customerData = JSON.parse(sessionStorage.getItem('customerData') || '{}');

console.log('=== DEBUG COMPLETO DO AGENDAMENTO ===');

console.log('1. VEÍCULO:', userVehicle);

console.log('2. PRODUTOS SELECIONADOS:', selectedProducts);

console.log('3. RECOMENDAÇÃO:', oilRecommendation);

console.log('4. OFICINA SELECIONADA:', selectedWorkshop);

console.log('5. DADOS DO CLIENTE:', customerData);

console.log('6. DADOS DO FORMULÁRIO:');

console.log(' - Data:', document.getElementById("schedule-date").value);

console.log(' - Hora:', document.getElementById("schedule-time").value);

console.log(' - Nome:', document.getElementById("customer-name").value);

console.log(' - CPF:', document.getElementById("customer-cpf").value);

console.log(' - Telefone:', document.getElementById("customer-phone").value);

console.log(' - Email:', document.getElementById("customer-email").value);

console.log('=====================================');

// Retorna os dados formatados para envio

return {

protocolo: `OIL${Date.now().toString().slice(-8)}`,

data\_hora: `${document.getElementById("schedule-date").value} ${document.getElementById("schedule-time").value}:00`,

oficina\_nome: selectedWorkshop?.nome || 'Oficina não selecionada',

oficina\_endereco: selectedWorkshop ? `${selectedWorkshop.endereco}, ${selectedWorkshop.cidade}/${selectedWorkshop.estado}` : 'Endereço não informado',

oficina\_telefone: selectedWorkshop?.telefone || 'Telefone não informado',

veiculo: userVehicle.marca && userVehicle.modelo && userVehicle.ano ?

`${userVehicle.marca} ${userVehicle.modelo} ${userVehicle.ano}` : 'Veículo não informado',

servicos: getServicosFormatados(selectedProducts, oilRecommendation),

total\_servico: calcularTotalServico(selectedProducts, oilRecommendation),

cliente\_nome: customerData.name || document.getElementById("customer-name").value,

cliente\_cpf: customerData.cpf || document.getElementById("customer-cpf").value,

cliente\_telefone: customerData.phone || document.getElementById("customer-phone").value,

cliente\_email: customerData.email || document.getElementById("customer-email").value

};

}

// Funções auxiliares para formatação

function getServicosFormatados(selectedProducts, oilRecommendation) {

const servicos = [];

if (selectedProducts.oil && oilRecommendation.oleo) {

servicos.push(`Óleo: ${oilRecommendation.oleo.nome}`);

}

if (selectedProducts.filter && oilRecommendation.filtro) {

servicos.push(`Filtro: ${oilRecommendation.filtro.nome}`);

}

return servicos.length > 0 ? servicos.join(' | ') : 'Serviços não especificados';

}

function calcularTotalServico(selectedProducts, oilRecommendation) {

let total = 0;

if (selectedProducts.oil && oilRecommendation.oleo && oilRecommendation.oleo.preco) {

total += parseFloat(oilRecommendation.oleo.preco) || 0;

}

if (selectedProducts.filter && oilRecommendation.filtro && oilRecommendation.filtro.preco) {

total += parseFloat(oilRecommendation.filtro.preco) || 0;

}

return total;

}

// Função para debug - verificar o que está no sessionStorage

function checkSessionStorage() {

console.log('=== DEBUG SESSION STORAGE ===');

console.log('customerData:', sessionStorage.getItem('customerData'));

console.log('userVehicle:', sessionStorage.getItem('userVehicle'));

console.log('oilRecommendation:', sessionStorage.getItem('oilRecommendation'));

console.log('selectedProducts:', sessionStorage.getItem('selectedProducts'));

console.log('=============================');

}

// Função auxiliar para formatar data

function formatDateForDisplay(dateString) {

const date = new Date(dateString);

return date.toLocaleDateString('pt-BR', {

day: '2-digit',

month: '2-digit',

year: 'numeric'

});

}

// ==================== FUNÇÕES AUXILIARES ====================

// Mostra/oculta o loading

function showLoading(show) {

const loadingOverlay = document.getElementById("loading-overlay");

if (loadingOverlay) {

loadingOverlay.style.display = show ? "flex" : "none";

}

}

// Exibe mensagens toast (versão responsiva)

function showToast(message, type = "success") {

const toast = document.getElementById("toast");

if (!toast) return;

toast.textContent = message;

toast.className = `toast ${type} show`;

// Ajuste de estilo para mobile

if (window.innerWidth <= 768) {

toast.style.width = "90%";

toast.style.left = "5%";

toast.style.transform = "none";

} else {

toast.style.width = "auto";

toast.style.left = "50%";

toast.style.transform = "translateX(-50%)";

}

setTimeout(() => {

toast.classList.remove("show");

}, 3000);

}

// Validação de CPF (simplificada)

function validateCPF(cpf) {

cpf = cpf.replace(/\D/g, '');

return cpf.length === 11;

}

// Validação de e-mail

function validateEmail(email) {

const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

return re.test(email);

}

// Formata CPF para exibição

function formatCPF(cpf) {

if (!cpf) return '';

cpf = cpf.replace(/\D/g, '');

return cpf.replace(/(\d{3})(\d{3})(\d{3})(\d{2})/, '$1.$2.$3-$4');

}

// Formata telefone para exibição

function formatPhone(phone) {

if (!phone) return '';

phone = phone.replace(/\D/g, '');

if (phone.length === 11) {

return phone.replace(/(\d{2})(\d{5})(\d{4})/, '($1) $2-$3');

}

return phone.replace(/(\d{2})(\d{4})(\d{4})/, '($1) $2-$3');

}

// Mostra/oculta o loading

function showLoading(show) {

document.getElementById("loading-overlay").style.display = show ? "flex" : "none";

}

// Exibe mensagens toast (versão responsiva)

function showToast(message, type = "success") {

const toast = document.getElementById("toast");

toast.textContent = message;

toast.className = `toast ${type} show`;

// Ajuste de estilo para mobile

if (window.innerWidth <= 768) {

toast.style.width = "90%";

toast.style.left = "5%";

toast.style.transform = "none";

} else {

toast.style.width = "auto";

toast.style.left = "50%";

toast.style.transform = "translateX(-50%)";

}

setTimeout(() => {

toast.classList.remove("show");

}, 3000);

}

// Configura data mínima para agendamento

function setMinScheduleDate() {

const today = new Date();

const tomorrow = new Date(today);

tomorrow.setDate(today.getDate() + 1);

scheduleDate.min = tomorrow.toISOString().split("T")[0];

scheduleDate.max = new Date(today.setMonth(today.getMonth() + 3)).toISOString().split("T")[0];

}

// ==================== EVENT LISTENERS ====================

// Seleção de marca -> carrega modelos

brandSelect.addEventListener("change", () => populateModels(brandSelect.value));

// Seleção de modelo -> carrega anos

modelSelect.addEventListener("change", () => populateYears(modelSelect.value));

// Formulário do veículo

vehicleForm.addEventListener("submit", async function(e) {

e.preventDefault();

if (!validateStep1()) return;

showLoading(true);

try {

// Armazena os dados do veículo

userVehicle = {

modelo\_ano\_id: yearSelect.value,

quilometragem: mileageInput.value.trim(),

marca: brandSelect.options[brandSelect.selectedIndex].text,

modelo: modelSelect.options[modelSelect.selectedIndex].text,

ano: yearSelect.options[yearSelect.selectedIndex].text

};

// Salva no sessionStorage para uso posterior

sessionStorage.setItem('userVehicle', JSON.stringify(userVehicle));

// Obtém recomendações

const recommendation = await getOilRecommendation(yearSelect.value);

showRecommendations(recommendation);

// Avança para o próximo passo

goToStep(2);

} catch (error) {

console.error("Erro:", error);

showToast("Erro ao processar veículo", "error");

} finally {

showLoading(false);

}

});

// Busca por endereço

locationButton.addEventListener("click", searchByAddress);

// Usar localização atual

userLocationBtn.addEventListener("click", getUserLocation);

// Máscara para CPF

document.getElementById("customer-cpf").addEventListener("input", function(e) {

let value = e.target.value.replace(/\D/g, "");

if (value.length > 9) {

value = value.replace(/^(\d{3})(\d{3})(\d{3})(\d{2})$/, "$1.$2.$3-$4");

} else if (value.length > 6) {

value = value.replace(/^(\d{3})(\d{3})(\d{3})$/, "$1.$2.$3");

} else if (value.length > 3) {

value = value.replace(/^(\d{3})(\d{3})$/, "$1.$2");

}

e.target.value = value;

});

// Máscara para telefone

document.getElementById("customer-phone").addEventListener("input", function(e) {

let value = e.target.value.replace(/\D/g, "");

if (value.length > 10) {

value = value.replace(/^(\d\d)(\d{5})(\d{4}).\*/, "($1) $2-$3");

} else if (value.length > 5) {

value = value.replace(/^(\d\d)(\d{4})(\d{0,4}).\*/, "($1) $2-$3");

} else if (value.length > 2) {

value = value.replace(/^(\d\d)(\d{0,5}).\*/, "($1) $2");

} else {

value = value.replace(/^(\d\*)/, "($1");

}

e.target.value = value;

});

// Modal de login (versão responsiva)

if (loginBtn && modal) {

loginBtn.addEventListener("click", function(e) {

e.preventDefault();

modal.style.display = "block";

document.body.style.overflow = "hidden";

});

closeModal.addEventListener("click", function() {

modal.style.display = "none";

document.body.style.overflow = "auto";

});

window.addEventListener("click", function(e) {

if (e.target === modal) {

modal.style.display = "none";

document.body.style.overflow = "auto";

}

});

// Ajuste do modal para mobile

const modalContent = document.querySelector(".modal-content");

if (modalContent) {

if (window.innerWidth <= 768) {

modalContent.style.width = "90%";

modalContent.style.padding = "20px";

} else {

modalContent.style.width = "400px";

modalContent.style.padding = "30px";

}

}

}

// Mostrar/esconder senha

if (togglePassword && passwordInput) {

togglePassword.addEventListener("click", function() {

const type = passwordInput.getAttribute("type") === "password" ? "text" : "password";

passwordInput.setAttribute("type", type);

this.querySelector("i").classList.toggle("fa-eye");

this.querySelector("i").classList.toggle("fa-eye-slash");

});

}

// Formulário de login (simulação)

if (loginForm) {

loginForm.addEventListener("submit", function(e) {

e.preventDefault();

const email = this.querySelector("#login-email").value.trim();

const password = this.querySelector("#login-password").value.trim();

if (!email || !password) {

showToast("Preencha todos os campos", "error");

return;

}

if (!validateEmail(email)) {

showToast("E-mail inválido", "error");

return;

}

showToast("Login realizado com sucesso!", "success");

modal.style.display = "none";

document.body.style.overflow = "auto";

});

}

// Autocomplete para busca de endereço (OTIMIZADO)

if (locationInput) {

locationInput.addEventListener("input", function(e) {

const query = e.target.value.trim();

const resultsContainer = document.getElementById("autocomplete-results");

// Limpa timeout anterior

if (addressSearchTimeout) {

clearTimeout(addressSearchTimeout);

}

if (query.length < 3) {

resultsContainer.classList.remove("active");

return;

}

// Adiciona debounce para evitar muitas requisições

addressSearchTimeout = setTimeout(async () => {

const addresses = await searchAddress(query);

showAddressSuggestions(addresses);

}, 300); // Reduzido de 500ms para 300ms

});

// Feche o autocomplete ao clicar fora

document.addEventListener("click", function(e) {

if (e.target !== locationInput) {

document.getElementById("autocomplete-results").classList.remove("active");

}

});

}

// ==================== INICIALIZAÇÃO ====================

function init() {

initMap();

populateBrands();

setMinScheduleDate();

handleResponsiveLayout();

// Adiciona listener para redimensionamento da janela

window.addEventListener("resize", handleResponsiveLayout);

}

init();

});

async function getOilRecommendation(modeloAnoId) {

showLoading(true);

try {

const response = await fetch(`/api/recomendacoes?modelo\_ano\_id=${modeloAnoId}`);

const result = await response.json();

console.log("DEBUG - Resposta completa da API:", result); // ADICIONE ESTA LINHA

if (!result.success) {

showToast(result.message, "error");

return null;

}

// DEBUG: Verificar especificamente os preços

if (result.data.oleo) {

console.log("DEBUG - Preço do óleo:", result.data.oleo.preco);

}

if (result.data.filtro) {

console.log("DEBUG - Preço do filtro:", result.data.filtro.preco);

}

// Armazena os dados da recomendação para uso posterior

sessionStorage.setItem('oilRecommendation', JSON.stringify(result.data));

return result.data;

} catch (error) {

console.error("Erro ao obter recomendação:", error);

showToast("Erro de comunicação ao obter recomendação.", "error");

return null;

} finally {

showLoading(false);

}

}

// Atualiza a barra de progresso

function updateProgressBar(step) {

const progressSteps = document.querySelectorAll('.progress-step');

// Atualiza as classes dos passos

progressSteps.forEach((stepElement, index) => {

const stepNumber = index + 1;

if (stepNumber < step) {

stepElement.classList.add('completed');

stepElement.classList.remove('active');

} else if (stepNumber === step) {

stepElement.classList.add('active');

stepElement.classList.remove('completed');

} else {

stepElement.classList.remove('active', 'completed');

}

});

}

// Na função displayWorkshops, você pode adicionar cores diferentes baseadas na distância:

function getDistanceColor(distancia) {

if (distancia < 0.5) return '#2a9d8f'; // Super perto

if (distancia < 1) return '#3a86ff'; // Muito perto

if (distancia < 3) return '#f4a261'; // Perto

if (distancia < 8) return '#ff9f1c'; // Moderado

if (distancia < 15) return '#ff6b6b'; // Um pouco longe

return '#e63946'; // Muito longe

}

// E modifique o item da lista para usar cores diferentes:

workshopItem.innerHTML = `

<div class="workshop-info">

<h4>${workshop.nome}</h4>

<p class="distance" style="color: ${getDistanceColor(workshop.distancia)};">

<i class="fas fa-route"></i> ${distanciaFormatada}

</p>

<p class="address"><i class="fas fa-map-marker-alt"></i> ${workshop.endereco}, ${workshop.cidade}/${workshop.estado}</p>

<p class="phone"><i class="fas fa-phone"></i> ${workshop.telefone || 'Não informado'}</p>

<p class="hours"><i class="fas fa-clock"></i> ${workshop.horario\_abertura} - ${workshop.horario\_fechamento}</p>

</div>

<button class="btn select-btn" onclick="selectWorkshop(${workshop.id})">Selecionar</button>

`;

// Atualiza o resumo da seleção

function updateSelectionSummary() {

const summaryElement = document.getElementById("selected-items");

const summaryContainer = document.querySelector(".selection-summary");

const selections = [];

if (selectedProducts.oil) {

selections.push("Óleo");

}

if (selectedProducts.filter) {

selections.push("Filtro");

}

if (selections.length === 0) {

summaryElement.textContent = "Nenhum produto selecionado";

summaryElement.style.color = "#e63946";

summaryContainer.classList.add("empty");

summaryContainer.classList.remove("highlight");

} else {

summaryElement.textContent = selections.join(" + ");

summaryElement.style.color = "#2a9d8f";

summaryContainer.classList.remove("empty");

summaryContainer.classList.add("highlight");

// Remove a classe de destaque após a animação

setTimeout(() => {

summaryContainer.classList.remove("highlight");

}, 500);

}

}

// Adicione esta função para inicializar o summary

function initSelectionSummary() {

const selectionContainer = document.createElement("div");

selectionContainer.className = "selection-summary";

selectionContainer.innerHTML = `

<h4>Resumo da Seleção</h4>

<p id="selected-items">Carregando...</p>

`;

// Insere o summary após os cards de recomendação

const recommendationContainer = document.querySelector(".recommendation-container");

if (recommendationContainer) {

recommendationContainer.parentNode.insertBefore(selectionContainer, recommendationContainer.nextSibling);

}

// Inicializa o summary

updateSelectionSummary();

}

// Função para debug - verificar dados antes do salvamento

function debugAgendamentoData() {

const userVehicle = JSON.parse(sessionStorage.getItem('userVehicle') || '{}');

const selectedProducts = getSelectedProducts();

const oilRecommendation = JSON.parse(sessionStorage.getItem('oilRecommendation') || '{}');

console.log('=== DEBUG AGENDAMENTO ===');

console.log('Veículo:', userVehicle);

console.log('Produtos selecionados:', selectedProducts);

console.log('Recomendação:', oilRecommendation);

console.log('Oficina selecionada:', selectedWorkshop);

console.log('Dados do formulário:');

console.log('- Data:', document.getElementById("schedule-date").value);

console.log('- Hora:', document.getElementById("schedule-time").value);

console.log('- Nome:', document.getElementById("customer-name").value);

console.log('- CPF:', document.getElementById("customer-cpf").value);

console.log('- Telefone:', document.getElementById("customer-phone").value);

console.log('- Email:', document.getElementById("customer-email").value);

console.log('=========================');

}

// Chame esta função antes de salvar

// debugAgendamentoData();

// Chame esta função antes de salvar para verificar os dados

// Chame initSelectionSummary() quando a página carregar

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

// === VERIFICAÇÃO DE LOGIN ===

// Verifica se o usuário tem token e dados no localStorage

const token = localStorage.getItem('token');

const userData = localStorage.getItem('user');

// Se não estiver logado, redireciona para login

if (!token || !userData) {

alert('Você precisa fazer login para agendar um serviço!');

window.location.href = 'login.html';

return; // Isso para a execução do código aqui

} initSelectionSummary();

});

// Função para melhorar a experiência do input de data

function enhanceDateInput() {

const dateInput = document.querySelector('input[type="date"]');

if (dateInput) {

// Adicionar um label dinâmico

const today = new Date();

const minDate = new Date();

minDate.setDate(today.getDate() + 1);

// Formatar a data mínima para o formato YYYY-MM-DD

const minDateString = minDate.toISOString().split('T')[0];

// Definir atributos min e max

dateInput.min = minDateString;

const maxDate = new Date();

maxDate.setMonth(today.getMonth() + 3);

dateInput.max = maxDate.toISOString().split('T')[0];

// Adicionar placeholder personalizado

dateInput.addEventListener('focus', function() {

this.classList.add('focused');

});

dateInput.addEventListener('blur', function() {

this.classList.remove('focused');

});

}

}

// Inicializar quando o DOM estiver carregado

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

enhanceDateInput();

});

// Validação do passo 3 (Dados do agendamento)

async function validateStep3() {

const scheduleDateValue = document.getElementById("schedule-date").value;

const scheduleTime = document.getElementById("schedule-time").value;

const customerName = document.getElementById("customer-name").value.trim();

const customerCpf = document.getElementById("customer-cpf").value.trim();

const customerPhone = document.getElementById("customer-phone").value.trim();

const customerEmail = document.getElementById("customer-email").value.trim();

if (!scheduleDateValue || !scheduleTime || !customerName || !customerCpf || !customerPhone || !customerEmail) {

showToast("Preencha todos os campos obrigatórios", "error");

return false;

}

if (!validateCPF(customerCpf)) {

showToast("CPF inválido", "error");

return false;

}

if (!validateEmail(customerEmail)) {

showToast("E-mail inválido", "error");

return false;

}

// SALVAR DADOS DO CLIENTE ANTES DE AVANÇAR

const customerData = {

name: customerName,

cpf: customerCpf,

phone: customerPhone,

email: customerEmail

};

sessionStorage.setItem('customerData', JSON.stringify(customerData));

return true;

}

// Mover ícone de login para dentro do menu hamburger em mobile - VERSÃO CORRIGIDA

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

const actionIcons = document.querySelector('.action-icons');

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

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

const navbar = document.querySelector('.navbar');

// Cria uma cópia do action-icons para o mobile

let mobileActionIcons = null;

function handleMobileMenu() {

if (window.innerWidth <= 768) {

// Mobile - cria cópia dentro do nav se não existir

if (actionIcons && nav && !mobileActionIcons) {

mobileActionIcons = actionIcons.cloneNode(true);

mobileActionIcons.classList.add('mobile-actions');

nav.appendChild(mobileActionIcons);

// Adiciona eventos aos elementos clonados

setupClonedActionIcons(mobileActionIcons);

}

// Controla visibilidade baseado no menu

if (mobileActionIcons) {

mobileActionIcons.style.display = nav.classList.contains('active') ? 'flex' : 'none';

}

// Esconde o original no mobile

if (actionIcons) {

actionIcons.style.display = 'none';

}

} else {

// Desktop - mostra o original e remove a cópia do mobile

if (actionIcons) {

actionIcons.style.display = 'flex';

}

if (mobileActionIcons) {

mobileActionIcons.remove();

mobileActionIcons = null;

}

}

}

// Configura eventos para os elementos clonados

function setupClonedActionIcons(clonedIcons) {

const dropdown = clonedIcons.querySelector('.user-dropdown');

const loginBtn = clonedIcons.querySelector('#login-btn');

if (dropdown) {

dropdown.addEventListener('click', function(e) {

e.stopPropagation();

const dropdownMenu = this.querySelector('.dropdown-menu');

if (dropdownMenu) {

dropdownMenu.style.display = dropdownMenu.style.display === 'block' ? 'none' : 'block';

}

});

}

}

// Controla visibilidade quando hamburger é clicado

if (hamburger && nav) {

hamburger.addEventListener('click', function() {

setTimeout(() => {

if (window.innerWidth <= 768 && mobileActionIcons) {

mobileActionIcons.style.display = nav.classList.contains('active') ? 'flex' : 'none';

}

}, 10);

});

}

// Fecha dropdown ao clicar fora (para versão mobile também)

document.addEventListener('click', function(e) {

if (mobileActionIcons) {

const dropdownMenu = mobileActionIcons.querySelector('.dropdown-menu');

const userDropdown = mobileActionIcons.querySelector('.user-dropdown');

if (dropdownMenu && userDropdown && !userDropdown.contains(e.target)) {

dropdownMenu.style.display = 'none';

}

}

});

// Executa inicialmente e no resize

handleMobileMenu();

window.addEventListener('resize', handleMobileMenu);

});

# Arquivo: public/js/teste.js

console.log('teste.js carregado');

# Arquivo: routes/agendamentoSimples.js

const express = require('express');

const router = express.Router();

const db = require('../database/db');

// Função para gerar código de confirmação único

function gerarCodigoConfirmacao() {

const timestamp = Date.now().toString().slice(-6);

const random = Math.floor(Math.random() \* 1000).toString().padStart(3, '0');

return `OIL${timestamp}${random}`;

}

// Função para limpar serviços

function limparServicos(servicos) {

if (!servicos) return 'Serviços não especificados';

// Remover caracteres indesejados

return servicos.toString()

.replace(/[\[\]"]/g, '')

.replace(/\s+/g, ' ')

.trim();

}

// Rota POST para criar agendamento

router.post("/", (req, res) => {

console.log('=== DADOS RECEBIDOS NO BACKEND ===');

console.log('Body completo:', JSON.stringify(req.body, null, 2));

console.log('==================================');

const {

protocolo,

data\_hora,

oficina\_nome,

oficina\_endereco,

oficina\_telefone,

veiculo,

servicos,

total\_servico,

cliente\_nome,

cliente\_cpf,

cliente\_telefone,

cliente\_email,

usuario\_id

} = req.body;

// VALIDAÇÃO MAIS FLEXÍVEL - apenas campos realmente essenciais

if (!cliente\_nome || !cliente\_telefone || !data\_hora) {

console.log('Campos obrigatórios faltando:', {

cliente\_nome: !!cliente\_nome,

cliente\_telefone: !!cliente\_telefone,

data\_hora: !!data\_hora

});

return res.status(400).json({

success: false,

message: 'Campos obrigatórios: nome, telefone e data/hora',

campos\_recebidos: req.body

});

}

// Usar protocolo gerado ou gerar um novo

const protocoloFinal = protocolo || gerarCodigoConfirmacao();

const servicosLimpos = limparServicos(servicos);

// PREPARAR VALORES COM VALIDAÇÕES INDIVIDUAIS

const values = [

protocoloFinal,

data\_hora || null,

oficina\_nome || 'Oficina não especificada',

oficina\_endereco || 'Endereço não informado',

oficina\_telefone || 'Telefone não informado',

veiculo ? (typeof veiculo === 'string' ? veiculo : JSON.stringify(veiculo)) : 'Veículo não informado',

servicosLimpos,

parseFloat(total\_servico) || 0.00,

cliente\_nome,

cliente\_cpf ? cliente\_cpf.replace(/\D/g, '') : 'CPF não informado',

cliente\_telefone,

cliente\_email || 'email@naoinformado.com',

usuario\_id || null

];

console.log('Valores para inserção:', values);

const query = `

INSERT INTO agendamento\_simples (

protocolo, data\_hora, oficina\_nome, oficina\_endereco, oficina\_telefone,

veiculo, servicos, total\_servico, cliente\_nome, cliente\_cpf, cliente\_telefone, cliente\_email, usuario\_id

) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)

`;

db.query(query, values, (err, result) => {

if (err) {

console.error('❌ Erro MySQL ao salvar agendamento:', err);

console.error('SQL:', err.sql);

return res.status(500).json({

success: false,

message: 'Erro ao salvar agendamento no banco de dados',

error: err.sqlMessage || err.message,

sql: err.sql,

values: values

});

}

console.log('✅ Agendamento salvo com sucesso. ID:', result.insertId);

res.status(201).json({

success: true,

agendamento\_id: result.insertId,

codigo\_confirmacao: protocoloFinal,

message: 'Agendamento salvo com sucesso!'

});

});

});

// Rota para buscar agendamentos por oficina

router.get('/oficina/:oficina\_nome', (req, res) => {

const { oficina\_nome } = req.params;

const query = `

SELECT \* FROM agendamento\_simples

WHERE oficina\_nome LIKE ?

ORDER BY data\_hora DESC

`;

db.query(query, [`%${oficina\_nome}%`], (err, results) => {

if (err) {

console.error('Erro ao buscar agendamentos:', err);

return res.status(500).json({

success: false,

message: 'Erro ao buscar agendamentos'

});

}

res.json({

success: true,

data: results

});

});

});

// Rota para buscar agendamentos por cliente

router.get("/cliente/:cliente\_email", (req, res) => {

const { cliente\_email } = req.params;

const query = `

SELECT \* FROM agendamento\_simples

WHERE cliente\_email = ?

ORDER BY data\_hora DESC

`;

db.query(query, [cliente\_email], (err, results) => {

if (err) {

console.error('Erro ao buscar agendamentos do cliente:', err);

return res.status(500).json({

success: false,

message: 'Erro ao buscar agendamentos do cliente'

});

}

res.json({

success: true,

data: results

});

});

});

// Rota para buscar agendamento por ID

router.get('/:id', (req, res) => {

const { id } = req.params;

const query = 'SELECT \* FROM agendamento\_simples WHERE id = ?';

db.query(query, [id], (err, results) => {

if (err) {

console.error('Erro ao buscar agendamento:', err);

return res.status(500).json({

success: false,

message: 'Erro ao buscar agendamento'

});

}

if (results.length === 0) {

return res.status(404).json({

success: false,

message: 'Agendamento não encontrado'

});

}

res.json({

success: true,

data: results[0]

});

});

});

// Rota para buscar todos os agendamentos

router.get('/', (req, res) => {

const query = 'SELECT \* FROM agendamento\_simples ORDER BY data\_hora DESC';

db.query(query, (err, results) => {

if (err) {

console.error('Erro ao buscar agendamentos:', err);

return res.status(500).json({

success: false,

message: 'Erro ao buscar agendamentos'

});

}

res.json({

success: true,

data: results

});

});

});

// Rota para atualizar agendamento

router.put('/:id', (req, res) => {

const { id } = req.params;

const {

data\_hora,

oficina\_nome,

oficina\_endereco,

oficina\_telefone,

veiculo,

servicos,

total\_servico,

cliente\_nome,

cliente\_cpf,

cliente\_telefone,

cliente\_email

} = req.body;

const servicosLimpos = limparServicos(servicos);

const query = `

UPDATE agendamento\_simples

SET data\_hora = ?, oficina\_nome = ?, oficina\_endereco = ?, oficina\_telefone = ?,

veiculo = ?, servicos = ?, total\_servico = ?, cliente\_nome = ?, cliente\_cpf = ?,

cliente\_telefone = ?, cliente\_email = ?

WHERE id = ?

`;

const values = [

data\_hora,

oficina\_nome,

oficina\_endereco,

oficina\_telefone,

veiculo ? (typeof veiculo === 'string' ? veiculo : JSON.stringify(veiculo)) : 'Veículo não informado',

servicosLimpos,

parseFloat(total\_servico) || 0.00,

cliente\_nome,

cliente\_cpf ? cliente\_cpf.replace(/\D/g, '') : 'CPF não informado',

cliente\_telefone,

cliente\_email,

id

];

db.query(query, values, (err, result) => {

if (err) {

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

return res.status(500).json({

success: false,

message: 'Erro ao atualizar agendamento'

});

}

if (result.affectedRows === 0) {

return res.status(404).json({

success: false,

message: 'Agendamento não encontrado'

});

}

res.json({

success: true,

message: 'Agendamento atualizado com sucesso!'

});

});

});

// Rota para deletar agendamento

router.delete('/:id', (req, res) => {

const { id } = req.params;

const query = 'DELETE FROM agendamento\_simples WHERE id = ?';

db.query(query, [id], (err, result) => {

if (err) {

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

return res.status(500).json({

success: false,

message: 'Erro ao deletar agendamento'

});

}

if (result.affectedRows === 0) {

return res.status(404).json({

success: false,

message: 'Agendamento não encontrado'

});

}

res.json({

success: true,

message: 'Agendamento deletado com sucesso!'

});

});

});

// Rota para buscar agendamentos por usuário ID

router.get('/usuario/:usuario\_id', (req, res) => {

const { usuario\_id } = req.params;

const query = `

SELECT \* FROM agendamento\_simples

WHERE usuario\_id = ?

ORDER BY data\_hora DESC

`;

db.query(query, [usuario\_id], (err, results) => {

if (err) {

console.error('Erro ao buscar agendamentos do usuário:', err);

return res.status(500).json({

success: false,

message: 'Erro ao buscar agendamentos'

});

}

res.json({

success: true,

data: results

});

});

});

// Adicione esta rota no final do arquivo agendamentoSimples.js

// Rota para cancelar agendamento

router.post('/:id/cancelar', (req, res) => {

const { id } = req.params;

const { motivo } = req.body;

console.log('🎯 Recebendo solicitação de cancelamento para agendamento:', id);

console.log('📝 Motivo:', motivo);

if (!motivo || motivo.trim() === '') {

return res.status(400).json({

success: false,

message: 'Motivo do cancelamento é obrigatório'

});

}

// Primeiro verificar se o agendamento existe e pode ser cancelado

const checkQuery = `

SELECT id, data\_hora, status

FROM agendamento\_simples

WHERE id = ?

`;

db.query(checkQuery, [id], (err, results) => {

if (err) {

console.error('❌ Erro ao verificar agendamento:', err);

return res.status(500).json({

success: false,

message: 'Erro ao verificar agendamento'

});

}

if (results.length === 0) {

return res.status(404).json({

success: false,

message: 'Agendamento não encontrado'

});

}

const agendamento = results[0];

const dataAgendamento = new Date(agendamento.data\_hora);

const agora = new Date();

// Verificar se já passou da data

if (dataAgendamento <= agora) {

return res.status(400).json({

success: false,

message: 'Não é possível cancelar agendamentos passados'

});

}

// Verificar se já está cancelado

if (agendamento.status === 'cancelado') {

return res.status(400).json({

success: false,

message: 'Este agendamento já está cancelado'

});

}

// Atualizar o agendamento

const updateQuery = `

UPDATE agendamento\_simples

SET status = 'cancelado',

motivo\_cancelamento = ?,

data\_cancelamento = NOW()

WHERE id = ?

`;

db.query(updateQuery, [motivo.trim(), id], (err, result) => {

if (err) {

console.error('❌ Erro ao cancelar agendamento:', err);

return res.status(500).json({

success: false,

message: 'Erro ao cancelar agendamento no banco de dados',

error: err.message

});

}

if (result.affectedRows === 0) {

return res.status(404).json({

success: false,

message: 'Agendamento não encontrado para cancelamento'

});

}

console.log('✅ Agendamento cancelado com sucesso. ID:', id);

res.json({

success: true,

message: 'Agendamento cancelado com sucesso!'

});

});

});

});

// Rota para atualizar status automaticamente (fora do prazo)

router.post('/atualizar-status/automatico', (req, res) => {

const query = `

UPDATE agendamento\_simples

SET status = 'fora\_prazo'

WHERE data\_hora < NOW()

AND status IN ('pendente', 'confirmado')

`;

db.query(query, (err, result) => {

if (err) {

console.error('Erro ao atualizar status automaticamente:', err);

return res.status(500).json({

success: false,

message: 'Erro ao atualizar status'

});

}

res.json({

success: true,

message: `${result.affectedRows} agendamentos atualizados para "fora do prazo"`

});

});

});

// agendamentoSimples.js - Adicione esta rota

// Rota para buscar último veículo agendado não cancelado do usuário

router.get('/usuario/:usuario\_id/ultimo-veiculo', (req, res) => {

const { usuario\_id } = req.params;

const query = `

SELECT \* FROM agendamento\_simples

WHERE usuario\_id = ?

AND status NOT IN ('cancelado', 'fora\_prazo')

AND data\_hora <= NOW()

ORDER BY data\_hora DESC

LIMIT 1

`;

db.query(query, [usuario\_id], (err, results) => {

if (err) {

console.error('Erro ao buscar último veículo agendado:', err);

return res.status(500).json({

success: false,

message: 'Erro ao buscar último veículo'

});

}

if (results.length === 0) {

return res.json({

success: true,

data: null,

message: 'Nenhum agendamento anterior encontrado'

});

}

res.json({

success: true,

data: results[0]

});

});

});

// agendamentoSimples.js - Adicione esta rota

// Rota para buscar último agendamento concluído do usuário

router.get('/usuario/:usuario\_id/ultimo-concluido', (req, res) => {

const { usuario\_id } = req.params;

const query = `

SELECT \* FROM agendamento\_simples

WHERE usuario\_id = ?

AND status IN ('concluido', 'confirmado')

AND data\_hora <= NOW()

ORDER BY data\_hora DESC

LIMIT 1

`;

db.query(query, [usuario\_id], (err, results) => {

if (err) {

console.error('Erro ao buscar último agendamento concluído:', err);

return res.status(500).json({

success: false,

message: 'Erro ao buscar último agendamento'

});

}

if (results.length === 0) {

return res.json({

success: true,

data: null,

message: 'Nenhum agendamento concluído encontrado'

});

}

res.json({

success: true,

data: results[0]

});

});

});

module.exports = router;

# Arquivo: routes/auth.js

const express = require('express');

const router = express.Router();

const db = require('../database/db');

const bcrypt = require('bcryptjs');

const jwt = require('jsonwebtoken');

const crypto = require('crypto');

const nodemailer = require('nodemailer');

const SECRET = 'oilsmart2025';

// Configuração do email (use Ethereal para teste)

async function createTransporter() {

// Para teste com Ethereal (email fake)

let testAccount = await nodemailer.createTestAccount();

return nodemailer.createTransport({

host: 'smtp.ethereal.email',

port: 587,

secure: false,

auth: {

user: testAccount.user,

pass: testAccount.pass

}

});

}

// Login

router.post('/login', (req, res) => {

const { email, senha } = req.body;

if (!email || !senha) {

return res.status(400).json({ success: false, message: 'Informe email e senha' });

}

db.query('SELECT \* FROM usuario WHERE email = ?', [email], async (err, results) => {

if (err) {

console.error('Erro no login:', err);

return res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

if (results.length === 0) {

return res.status(401).json({ success: false, message: 'Usuário não encontrado' });

}

const user = results[0];

try {

const senhaOk = await bcrypt.compare(senha, user.senha);

if (!senhaOk) {

return res.status(401).json({ success: false, message: 'Senha incorreta' });

}

const token = jwt.sign({

id: user.id,

tipo: user.tipo,

nome: user.nome

}, SECRET, { expiresIn: '2h' });

res.json({

success: true,

message: 'Login realizado com sucesso',

token,

user: {

id: user.id,

nome: user.nome,

email: user.email,

tipo: user.tipo

}

});

} catch (error) {

console.error('Erro ao comparar senhas:', error);

res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

});

});

// Registro

router.post('/register', async (req, res) => {

const { nome, email, senha, tipo, telefone, cpf, endereco, cep, cidade, estado } = req.body;

if (!nome || !email || !senha) {

return res.status(400).json({ success: false, message: 'Preencha todos os campos obrigatórios' });

}

try {

// Verificar se email já existe

db.query('SELECT id FROM usuario WHERE email = ?', [email], async (err, results) => {

if (err) {

console.error('Erro ao verificar email:', err);

return res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

if (results.length > 0) {

return res.status(400).json({ success: false, message: 'Email já cadastrado. Tente outro' });

}

// Criptografar senha

const senhaCriptografada = await bcrypt.hash(senha, 10);

// Inserir usuário

db.query(`INSERT INTO usuario (nome, email, senha)

VALUES (?, ?, ?)`,

[nome, email, senhaCriptografada, tipo || 'cliente', telefone, cpf, endereco, cep, cidade, estado],

(err, result) => {

if (err) {

console.error('Erro ao registrar usuário:', err);

return res.status(500).json({ success: false, message: 'Erro ao registrar usuário' });

}

res.status(201).json({

success: true,

message: 'Usuário registrado com sucesso',

userId: result.insertId

});

}

);

});

} catch (error) {

console.error('Erro no registro:', error);

res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

});

// Esqueci a senha

router.post('/forgot-password', async (req, res) => {

const { email } = req.body;

if (!email) {

return res.status(400).json({ success: false, message: 'Informe o e-mail' });

}

try {

// Verificar se o email existe

db.query('SELECT \* FROM usuario WHERE email = ?', [email], async (err, results) => {

if (err) {

console.error('Erro ao verificar email:', err);

return res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

if (results.length === 0) {

// Por segurança, não revelar se o email existe ou não

return res.json({

success: true,

message: 'Se o email estiver cadastrado, você receberá instruções de recuperação'

});

}

const user = results[0];

// Gerar token de recuperação

const resetToken = crypto.randomBytes(32).toString('hex');

const resetTokenExpiry = Date.now() + 3600000; // 1 hora

// Salvar token no banco

db.query('UPDATE usuario SET reset\_token = ?, reset\_token\_expiry = ? WHERE id = ?',

[resetToken, resetTokenExpiry, user.id],

async (err) => {

if (err) {

console.error('Erro ao salvar token:', err);

return res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

try {

// Criar transporter

const transporter = await createTransporter();

// Enviar email

const resetLink = `http://localhost:3000/html/redefinir-senha.html?token=${resetToken}`;

const mailOptions = {

to: email,

subject: 'Recuperação de Senha - OilSmart',

html: `

<div style="font-family: Arial, sans-serif; max-width: 600px; margin: 0 auto;">

<h2 style="color: #213f57;">Recuperação de Senha - OilSmart</h2>

<p>Você solicitou a redefinição de sua senha. Clique no link abaixo para criar uma nova senha:</p>

<div style="text-align: center; margin: 20px 0;">

<a href="${resetLink}" style="display: inline-block; padding: 12px 24px; background-color: #213f57; color: white; text-decoration: none; border-radius: 5px; font-weight: bold;">Redefinir Senha</a>

</div>

<p>Este link expira em 1 hora.</p>

<p>Se você não solicitou esta redefinição, ignore este email.</p>

<hr style="border: none; border-top: 1px solid #eee; margin: 20px 0;">

<p style="color: #b49434; font-size: 12px;">Equipe OilSmart</p>

</div>

`

};

const info = await transporter.sendMail(mailOptions);

console.log('Email enviado:', nodemailer.getTestMessageUrl(info));

res.json({

success: true,

message: 'Se o email estiver cadastrado, você receberá instruções de recuperação'

});

} catch (emailError) {

console.error('Erro ao enviar email:', emailError);

// Para desenvolvimento: retornar o link no console

const resetLink = `http://localhost:3000/redefinir-senha.html?token=${resetToken}`;

console.log('Link de recuperação (para teste):', resetLink);

res.json({

success: true,

message: 'Email de recuperação gerado (ver console para detalhes)',

testLink: resetLink // Apenas para desenvolvimento

});

}

}

);

});

} catch (error) {

console.error('Erro no processo de recuperação:', error);

res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

});

// Redefinir senha

router.post('/reset-password', async (req, res) => {

const { token, senha } = req.body;

if (!token || !senha) {

return res.status(400).json({ success: false, message: 'Token e senha são obrigatórios' });

}

if (senha.length < 6) {

return res.status(400).json({ success: false, message: 'A senha deve ter pelo menos 6 caracteres' });

}

try {

// Buscar usuário pelo token válido

db.query('SELECT \* FROM usuario WHERE reset\_token = ? AND reset\_token\_expiry > ?',

[token, Date.now()],

async (err, results) => {

if (err) {

console.error('Erro ao verificar token:', err);

return res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

if (results.length === 0) {

return res.status(400).json({ success: false, message: 'Token inválido ou expirado' });

}

const user = results[0];

try {

// Criptografar nova senha

const senhaCriptografada = await bcrypt.hash(senha, 10);

// Atualizar senha e limpar token

db.query('UPDATE usuario SET senha = ?, reset\_token = NULL, reset\_token\_expiry = NULL WHERE id = ?',

[senhaCriptografada, user.id],

(err, result) => {

if (err) {

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

return res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

res.json({ success: true, message: 'Senha redefinida com sucesso' });

}

);

} catch (hashError) {

console.error('Erro ao criptografar senha:', hashError);

res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

}

);

} catch (error) {

console.error('Erro no processo de redefinição:', error);

res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

});

// Verificar token de redefinição

router.get('/verify-reset-token/:token', (req, res) => {

const { token } = req.params;

db.query('SELECT id FROM usuario WHERE reset\_token = ? AND reset\_token\_expiry > ?',

[token, Date.now()],

(err, results) => {

if (err) {

console.error('Erro ao verificar token:', err);

return res.status(500).json({ success: false, message: 'Erro interno do servidor' });

}

if (results.length === 0) {

return res.status(400).json({ success: false, message: 'Token inválido ou expirado' });

}

res.json({ success: true, message: 'Token válido' });

}

);

});

module.exports = router;

# Arquivo: routes/contact.js

const express = require('express');

const router = express.Router();

const nodemailer = require('nodemailer');

const db = require('../database/db');

// Configuração do transporter do nodemailer

const transporter = nodemailer.createTransport({

service: 'gmail',

auth: {

user: process.env.EMAIL\_USER || 'oilsmartoficiall@gmail.com',

pass: process.env.EMAIL\_PASS || 'cyog lhar hnpm plgw'

}

});

// Verificar conexão com o serviço de email

transporter.verify(function(error, success) {

if (error) {

console.error('Erro na configuração do email:', error);

} else {

console.log('Servidor de email configurado com sucesso');

}

});

router.post('/', async (req, res) => {

try {

const { name, email, phone, subject, message } = req.body;

// Validação básica

if (!name || !email || !subject || !message) {

return res.status(400).json({

success: false,

message: 'Todos os campos obrigatórios devem ser preenchidos'

});

}

// Validação de email

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailRegex.test(email)) {

return res.status(400).json({

success: false,

message: 'Email inválido'

});

}

const query = `

INSERT INTO contatos (nome, email, telefone, assunto, mensagem, data\_contato)

VALUES (?, ?, ?, ?, ?, NOW())

`;

// Salvar no banco de dados

db.query(query, [name, email, phone, subject, message], (err, results) => {

if (err) {

console.error('Erro ao salvar contato no banco:', err);

return res.status(500).json({

success: false,

message: 'Erro ao salvar contato no sistema'

});

}

// Configurar opções do email

const mailOptions = {

from: {

name: 'OilSmart Contato',

address: process.env.EMAIL\_USER || 'oilsmartoficiall@gmail.com'

},

to: 'oilsmartoficiall@gmail.com',

subject: `📧 Contato OilSmart: ${subject}`,

html: `

<div style="font-family: Arial, sans-serif; max-width: 600px; margin: 0 auto;">

<h2 style="color: #1d3557;">Nova mensagem de contato - OilSmart</h2>

<div style="background: #f8f9fa; padding: 20px; border-radius: 8px;">

<p><strong>👤 Nome:</strong> ${name}</p>

<p><strong>📧 E-mail:</strong> ${email}</p>

<p><strong>📞 Telefone:</strong> ${phone || 'Não informado'}</p>

<p><strong>📋 Assunto:</strong> ${subject}</p>

<p><strong>💬 Mensagem:</strong></p>

<div style="background: white; padding: 15px; border-radius: 5px; border-left: 4px solid #1d3557;">

${message.replace(/\n/g, '<br>')}

</div>

</div>

<hr style="border: none; border-top: 1px solid #ddd; margin: 20px 0;">

<p style="color: #666; font-size: 12px;">

Enviado em: ${new Date().toLocaleString('pt-BR')}<br>

IP: ${req.ip}

</p>

</div>

`

};

// Enviar email

transporter.sendMail(mailOptions, (error, info) => {

if (error) {

console.error('Erro ao enviar e-mail:', error);

return res.status(500).json({

success: false,

message: 'Erro ao enviar mensagem. Tente novamente mais tarde.'

});

}

console.log('E-mail enviado com sucesso:', info.messageId);

res.json({

success: true,

message: 'Mensagem enviada com sucesso! Entraremos em contato em breve.',

messageId: info.messageId

});

});

});

} catch (error) {

console.error('Erro ao processar formulário:', error);

res.status(500).json({

success: false,

message: 'Erro interno do servidor. Tente novamente mais tarde.'

});

}

});

router.get('/', (req, res) => {

const query = 'SELECT \* FROM contatos ORDER BY data\_contato DESC';

db.query(query, (err, results) => {

if (err) {

console.error('Erro ao buscar contatos:', err);

return res.status(500).json({ error: 'Erro interno do servidor' });

}

res.json(results);

});

});

module.exports = router;

// Adicione no contact.js

router.get('/test-email', (req, res) => {

const mailOptions = {

from: process.env.EMAIL\_USER,

to: 'oilsmartoficiall@gmail.com',

subject: 'Teste de Email - OilSmart',

text: 'Este é um email de teste do sistema OilSmart'

};

transporter.sendMail(mailOptions, (error, info) => {

if (error) {

console.error('Erro no teste de email:', error);

return res.status(500).json({

success: false,

error: error.toString()

});

}

res.json({

success: true,

message: 'Email de teste enviado com sucesso',

messageId: info.messageId

});

});

});

# Arquivo: routes/geocode.js

const express = require("express");

const router = express.Router();

const axios = require("axios");

// Rota para geocodificação usando Nominatim (OpenStreetMap)

router.get("/", async (req, res) => {

const { address } = req.query;

if (!address) {

return res.status(400).json({ error: "Endereço não fornecido" });

}

try {

// URL do Nominatim para geocodificação

const response = await axios.get("https://nominatim.openstreetmap.org/search", {

params: {

q: address, // endereço a ser buscado

format: "json", // formato da resposta

addressdetails: 1,// detalhes do endereço

limit: 1 // retornar apenas o resultado mais próximo

},

headers: {

"User-Agent": "SeuAppNome/1.0 (seu-email@example.com)" // obrigatório pelo Nominatim

}

});

if (response.data.length === 0) {

return res.status(404).json({ error: "Endereço não encontrado" });

}

// Retorna latitude e longitude do primeiro resultado

const result = response.data[0];

res.json({

latitude: result.lat,

longitude: result.lon,

display\_name: result.display\_name,

address: result.address

});

} catch (error) {

console.error("Erro na geocodificação:", error.message);

res.status(500).json({ error: "Erro interno do servidor" });

}

});

module.exports = router;

# Arquivo: routes/lembreteTrocaOleo.js

const express = require('express');

const router = express.Router();

// "Banco de dados" em memória

let lembretes = [];

// Rota para buscar lembrete do usuário

router.get('/usuario/:usuario\_id', (req, res) => {

const { usuario\_id } = req.params;

const lembrete = lembretes.find(l => l.usuario\_id === parseInt(usuario\_id));

if (!lembrete) {

return res.json({

success: true,

data: null,

message: 'Nenhum lembrete encontrado'

});

}

res.json({

success: true,

data: lembrete

});

});

// Rota para criar/atualizar lembrete

router.post('/', (req, res) => {

const {

usuario\_id,

veiculo\_modelo = 'Meu Veículo',

ultima\_troca\_km,

ultima\_troca\_data,

proxima\_troca\_km,

proxima\_troca\_data,

tipo\_oleo = 'Óleo semissintético',

observacoes = ''

} = req.body;

if (!usuario\_id) {

return res.status(400).json({

success: false,

message: 'Usuário ID é obrigatório'

});

}

// Verifica se já existe lembrete para este usuário

const index = lembretes.findIndex(l => l.usuario\_id === parseInt(usuario\_id));

const novoLembrete = {

id: index >= 0 ? lembretes[index].id : Date.now(),

usuario\_id: parseInt(usuario\_id),

veiculo\_modelo,

ultima\_troca\_km: ultima\_troca\_km || 80000,

ultima\_troca\_data: ultima\_troca\_data || '2023-05-15',

proxima\_troca\_km: proxima\_troca\_km || 85000,

proxima\_troca\_data: proxima\_troca\_data || '2023-11-15',

tipo\_oleo,

observacoes,

updated\_at: new Date().toISOString()

};

if (index >= 0) {

// Atualizar existente

lembretes[index] = novoLembrete;

} else {

// Criar novo

lembretes.push(novoLembrete);

}

res.json({

success: true,

message: index >= 0 ? 'Lembrete atualizado com sucesso!' : 'Lembrete criado com sucesso!',

data: novoLembrete

});

});

// Rota para dados de exemplo (pré-popular)

router.post('/exemplo/:usuario\_id', (req, res) => {

const { usuario\_id } = req.params;

const lembreteExemplo = {

id: Date.now(),

usuario\_id: parseInt(usuario\_id),

veiculo\_modelo: 'Honda Civic 2020',

ultima\_troca\_km: 80000,

ultima\_troca\_data: '2023-05-15',

proxima\_troca\_km: 85000,

proxima\_troca\_data: '2023-11-15',

tipo\_oleo: 'Óleo semissintético 10W40',

observacoes: 'Próxima troca recomendada',

updated\_at: new Date().toISOString()

};

// Remove lembrete existente se houver

lembretes = lembretes.filter(l => l.usuario\_id !== parseInt(usuario\_id));

lembretes.push(lembreteExemplo);

res.json({

success: true,

message: 'Lembrete de exemplo criado!',

data: lembreteExemplo

});

});

// Rota para deletar lembrete

router.delete('/usuario/:usuario\_id', (req, res) => {

const { usuario\_id } = req.params;

const initialLength = lembretes.length;

lembretes = lembretes.filter(l => l.usuario\_id !== parseInt(usuario\_id));

if (lembretes.length < initialLength) {

res.json({

success: true,

message: 'Lembrete removido com sucesso!'

});

} else {

res.status(404).json({

success: false,

message: 'Lembrete não encontrado'

});

}

});

// Rota para listar todos os lembretes (apenas para debug)

router.get('/debug/todos', (req, res) => {

res.json({

success: true,

data: lembretes,

total: lembretes.length

});

});

module.exports = router;

# Arquivo: routes/marca.js

const express = require("express");

const router = express.Router();

const db = require("../database/db");

// Rota para obter todas as marcas

router.get("/", (req, res) => {

const sql = "SELECT \* FROM marca";

db.query(sql, (err, results) => {

if (err) {

console.error("Erro ao buscar marcas:", err);

res.status(500).json({ error: "Erro interno do servidor" });

return;

}

res.json(results);

});

});

module.exports = router;

# Arquivo: routes/modelo.js

const express = require("express");

const router = express.Router();

const db = require("../database/db");

// Rota para obter modelos por marca (via query parameter)

router.get("/", (req, res) => {

const { marca\_id } = req.query;

if (!marca\_id) {

return res.status(400).json({ error: "marca\_id é obrigatório" });

}

const sql = "SELECT \* FROM modelo WHERE marca\_id = ?";

db.query(sql, [marca\_id], (err, results) => {

if (err) {

console.error("Erro ao buscar modelos:", err);

res.status(500).json({ error: "Erro interno do servidor" });

return;

}

res.json(results);

});

});

// Rota para obter modelos por marca (via parâmetro de URL)

router.get("/:marcaId", (req, res) => {

const { marcaId } = req.params;

const sql = "SELECT \* FROM modelo WHERE marca\_id = ?";

db.query(sql, [marcaId], (err, results) => {

if (err) {

console.error("Erro ao buscar modelos:", err);

res.status(500).json({ error: "Erro interno do servidor" });

return;

}

res.json(results);

});

});

module.exports = router;

# Arquivo: routes/modeloAno.js

const express = require("express");

const router = express.Router();

const db = require("../database/db");

// Rota para obter anos de modelo por modelo (via query parameter)

router.get("/", (req, res) => {

const { modelo\_id } = req.query;

if (!modelo\_id) {

return res.status(400).json({ error: "modelo\_id é obrigatório" });

}

const sql = "SELECT \* FROM modelo\_ano WHERE modelo\_id = ?";

db.query(sql, [modelo\_id], (err, results) => {

if (err) {

console.error("Erro ao buscar anos de modelo:", err);

res.status(500).json({ error: "Erro interno do servidor" });

return;

}

res.json(results);

});

});

// Rota para obter anos de modelo por modelo (via parâmetro de URL)

router.get("/:modeloId", (req, res) => {

const { modeloId } = req.params;

const sql = "SELECT \* FROM modelo\_ano WHERE modelo\_id = ?";

db.query(sql, [modeloId], (err, results) => {

if (err) {

console.error("Erro ao buscar anos de modelo:", err);

res.status(500).json({ error: "Erro interno do servidor" });

return;

}

res.json(results);

});

});

module.exports = router;

# Arquivo: routes/oficina.js

const express = require('express');

const router = express.Router();

const db = require('../database/db');

// Rota para obter todas as oficinas, com filtros opcionais por cidade e estado

router.get('/', async (req, res) => {

try {

let sql = `

SELECT

id,

usuario\_id,

nome,

cep,

endereco,

cidade,

estado,

telefone,

horario\_abertura,

horario\_fechamento,

dias\_funcionamento,

lat,

lng

FROM oficina

`;

const params = [];

const conditions = [];

if (req.query.cidade) {

conditions.push('cidade = ?');

params.push(req.query.cidade);

}

if (req.query.estado) {

conditions.push('estado = ?');

params.push(req.query.estado);

}

if (conditions.length > 0) {

sql += ` WHERE ` + conditions.join(' AND ');

}

db.query(sql, params, (err, results) => {

if (err) {

console.error('Erro ao buscar oficinas:', err);

return res.status(500).json({

success: false,

message: 'Erro ao buscar oficinas no banco de dados'

});

}

res.json({

success: true,

data: results

});

});

} catch (error) {

console.error('Erro no endpoint /api/oficina:', error);

res.status(500).json({

success: false,

message: 'Erro interno no servidor'

});

}

});

// Rota para obter oficina por ID

router.get('/:id', (req, res) => {

const { id } = req.params;

const sql = `

SELECT

id,

usuario\_id,

nome,

cep,

endereco,

cidade,

estado,

telefone,

horario\_abertura,

horario\_fechamento,

dias\_funcionamento,

lat,

lng

FROM oficina

WHERE id = ?

`;

db.query(sql, [id], (err, results) => {

if (err) {

console.error('Erro ao buscar oficina:', err);

return res.status(500).json({

error: 'Erro interno do servidor'

});

}

if (results.length === 0) {

return res.status(404).json({

error: 'Oficina não encontrada'

});

}

res.json(results[0]);

});

});

module.exports = router;

# Arquivo: routes/recomendacao.js

const express = require("express");

const router = express.Router();

const db = require("../database/db");

router.get("/", (req, res) => {

const { modelo\_ano\_id } = req.query;

console.log("Recebida requisição para modelo\_ano\_id:", modelo\_ano\_id);

if (!modelo\_ano\_id) {

return res.status(400).json({

success: false,

message: "O 'modelo\_ano\_id' é obrigatório."

});

}

// Consulta otimizada para buscar recomendações

const sql = `

SELECT

r.id AS recomendacao\_id,

oleo.id AS oleo\_id,

oleo.nome AS oleo\_nome,

oleo.tipo AS oleo\_tipo,

oleo.viscosidade AS oleo\_viscosidade,

oleo.especificacao AS oleo\_especificacao,

oleo.marca AS oleo\_marca,

COALESCE(oleo.preco, 0) AS oleo\_preco,

filtro.id AS filtro\_id,

filtro.nome AS filtro\_nome,

filtro.tipo AS filtro\_tipo,

filtro.compatibilidade\_modelo AS filtro\_compatibilidade,

COALESCE(filtro.preco, 0) AS filtro\_preco,

m.nome AS modelo\_nome,

mar.nome AS marca\_nome,

ma.ano AS ano\_modelo

FROM recomendacao r

LEFT JOIN produto\_oleo oleo ON r.oleo\_id = oleo.id

LEFT JOIN produto\_filtro filtro ON r.filtro\_id = filtro.id

LEFT JOIN modelo\_ano ma ON r.modelo\_ano\_id = ma.id

LEFT JOIN modelo m ON ma.modelo\_id = m.id

LEFT JOIN marca mar ON m.marca\_id = mar.id

WHERE r.modelo\_ano\_id = ?

LIMIT 1

`;

console.log("Executando SQL para recomendação");

db.query(sql, [modelo\_ano\_id], (err, results) => {

if (err) {

console.error("Erro MySQL ao buscar recomendação:", err);

return res.status(500).json({

success: false,

message: "Erro interno do servidor ao buscar recomendação."

});

}

console.log("Resultados encontrados:", results.length);

if (results.length === 0) {

return res.status(404).json({

success: false,

message: "Nenhuma recomendação encontrada para o veículo selecionado."

});

}

const recomendacao = results[0];

// Formatar resposta

const response = {

success: true,

data: {

veiculo: {

modelo: recomendacao.modelo\_nome,

marca: recomendacao.marca\_nome,

ano: recomendacao.ano\_modelo

},

oleo: recomendacao.oleo\_id ? {

id: recomendacao.oleo\_id,

nome: recomendacao.oleo\_nome,

tipo: recomendacao.oleo\_tipo,

viscosidade: recomendacao.oleo\_viscosidade,

especificacao: recomendacao.oleo\_especificacao,

marca: recomendacao.oleo\_marca,

preco: parseFloat(recomendacao.oleo\_preco)

} : null,

filtro: recomendacao.filtro\_id ? {

id: recomendacao.filtro\_id,

nome: recomendacao.filtro\_nome,

tipo: recomendacao.filtro\_tipo,

compatibilidade\_modelo: recomendacao.filtro\_compatibilidade,

preco: parseFloat(recomendacao.filtro\_preco)

} : null,

total: (parseFloat(recomendacao.oleo\_preco || 0) + parseFloat(recomendacao.filtro\_preco || 0)).toFixed(2)

}

};

console.log("Recomendação enviada com sucesso");

res.json(response);

});

});

// Nova rota para buscar recomendações por modelo, marca e ano

router.get("/por-veiculo", (req, res) => {

const { marca\_id, modelo\_id, ano } = req.query;

if (!marca\_id || !modelo\_id || !ano) {

return res.status(400).json({

success: false,

message: "Os parâmetros 'marca\_id', 'modelo\_id' e 'ano' são obrigatórios."

});

}

const sql = `

SELECT

r.id AS recomendacao\_id,

oleo.id AS oleo\_id,

oleo.nome AS oleo\_nome,

oleo.tipo AS oleo\_tipo,

oleo.viscosidade AS oleo\_viscosidade,

oleo.especificacao AS oleo\_especificacao,

oleo.marca AS oleo\_marca,

COALESCE(oleo.preco, 0) AS oleo\_preco,

filtro.id AS filtro\_id,

filtro.nome AS filtro\_nome,

filtro.tipo AS filtro\_tipo,

filtro.compatibilidade\_modelo AS filtro\_compatibilidade,

COALESCE(filtro.preco, 0) AS filtro\_preco,

m.nome AS modelo\_nome,

mar.nome AS marca\_nome,

ma.ano AS ano\_modelo

FROM recomendacao r

LEFT JOIN produto\_oleo oleo ON r.oleo\_id = oleo.id

LEFT JOIN produto\_filtro filtro ON r.filtro\_id = filtro.id

LEFT JOIN modelo\_ano ma ON r.modelo\_ano\_id = ma.id

LEFT JOIN modelo m ON ma.modelo\_id = m.id

LEFT JOIN marca mar ON m.marca\_id = mar.id

WHERE m.marca\_id = ? AND m.id = ? AND ma.ano = ?

LIMIT 1

`;

db.query(sql, [marca\_id, modelo\_id, ano], (err, results) => {

if (err) {

console.error("Erro MySQL ao buscar recomendação por veículo:", err);

return res.status(500).json({

success: false,

message: "Erro interno do servidor."

});

}

if (results.length === 0) {

return res.status(404).json({

success: false,

message: "Nenhuma recomendação encontrada para o veículo especificado."

});

}

const recomendacao = results[0];

const response = {

success: true,

data: {

veiculo: {

modelo: recomendacao.modelo\_nome,

marca: recomendacao.marca\_nome,

ano: recomendacao.ano\_modelo

},

oleo: {

id: recomendacao.oleo\_id,

nome: recomendacao.oleo\_nome,

tipo: recomendacao.oleo\_tipo,

viscosidade: recomendacao.oleo\_viscosidade,

especificacao: recomendacao.oleo\_especificacao,

marca: recomendacao.oleo\_marca,

preco: parseFloat(recomendacao.oleo\_preco)

},

filtro: {

id: recomendacao.filtro\_id,

nome: recomendacao.filtro\_nome,

tipo: recomendacao.filtro\_tipo,

compatibilidade\_modelo: recomendacao.filtro\_compatibilidade,

preco: parseFloat(recomendacao.filtro\_preco)

},

total: (parseFloat(recomendacao.oleo\_preco || 0) + parseFloat(recomendacao.filtro\_preco || 0)).toFixed(2)

}

};

res.json(response);

});

});

module.exports = router;

# Arquivo: routes/veiculo.js

const express = require("express");

const router = express.Router();

const db = require("../database/db");

// Rota para obter todos os veículos

router.get("/", (req, res) => {

const sql = "SELECT \* FROM veiculo";

db.query(sql, (err, results) => {

if (err) {

console.error("Erro ao buscar veículos:", err);

res.status(500).json({ error: "Erro interno do servidor" });

return;

}

res.json(results);

});

});

module.exports = router;

# Arquivo: server.js

const express = require('express');

const bcrypt = require('bcryptjs');

const jwt = require('jsonwebtoken');

const path = require('path');

const cors = require('cors');

require('dotenv').config();

const app = express();

const PORT = process.env.PORT || 3000;

// Middlewares

app.use(cors());

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

app.use(express.static(path.join(\_\_dirname, 'public')));

// Conexão com banco

const db = require('./database/db');

// Middleware de autenticação

function authenticateToken(req, res, next) {

const authHeader = req.headers['authorization'];

const token = authHeader && authHeader.split(' ')[1];

if (!token) {

return res.status(401).json({

success: false,

message: 'Token de acesso necessário'

});

}

jwt.verify(token, process.env.JWT\_SECRET || 'seu\_segredo\_jwt', (err, user) => {

if (err) {

return res.status(403).json({

success: false,

message: 'Token inválido'

});

}

req.user = user;

next();

});

}

// Importar rotas

const authRoutes = require('./routes/auth');

const agendamentoSimplesRoutes = require('./routes/agendamentoSimples');

const oficinaRoutes = require('./routes/oficina');

const veiculoRoutes = require('./routes/veiculo');

const marcaRoutes = require('./routes/marca');

const modeloRoutes = require('./routes/modelo');

const modeloAnoRoutes = require('./routes/modeloAno');

const recomendacaoRoutes = require('./routes/recomendacao');

const geocodeRoutes = require('./routes/geocode');

const lembreteTrocaOleoRoutes = require('./routes/lembreteTrocaOleo');

// Usar rotas

app.use('/api/auth', authRoutes);

app.use('/api/oficina', oficinaRoutes);

app.use('/api/veiculos', veiculoRoutes);

app.use('/api/marcas', marcaRoutes);

app.use('/api/modelos', modeloRoutes);

app.use('/api/modelo\_anos', modeloAnoRoutes);

app.use('/api/recomendacoes', recomendacaoRoutes);

app.use('/api/geocode', geocodeRoutes);

app.use('/api/contact', require('./routes/contact'));

app.use('/api/agendamento\_simples', agendamentoSimplesRoutes);

app.use('/api/lembretes\_troca\_oleo', lembreteTrocaOleoRoutes);

// Rotas de produtos (do arquivo novo)

app.get('/api/produtos/oleo/:id', (req, res) => {

const { id } = req.params;

db.query('SELECT \* FROM produto\_oleo WHERE id = ?', [id], (err, results) => {

if (err) {

console.error('Erro ao buscar óleo:', err);

return res.status(500).json({ error: 'Erro interno do servidor' });

}

res.json(results[0] || null);

});

});

app.get('/api/produtos/filtro/:id', (req, res) => {

const { id } = req.params;

db.query('SELECT \* FROM produto\_filtro WHERE id = ?', [id], (err, results) => {

if (err) {

console.error('Erro ao buscar filtro:', err);

return res.status(500).json({ error: 'Erro interno do servidor' });

}

res.json(results[0] || null);

});

});

// Rotas gerais

app.get('/', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public/html/index.html'));

});

app.get('/api/test', (req, res) => {

res.json({ message: 'Servidor funcionando!', timestamp: new Date() });

});

// Iniciar servidor

app.listen(PORT, () => {

console.log(`✅ Servidor rodando em http://localhost:${PORT}`);

});

// Rota para criar agendamento

// app.post('/api/agendamentos', authenticateToken, async (req, res) => {

// try {

// const {

// cliente\_id,

// oficina\_id,

// veiculo\_id,

// data\_agendamento,

// servicos,

// produtos,

// observacoes

// } = req.body;

// // Verificar se o cliente\_id corresponde ao usuário logado

// if (cliente\_id !== req.user.id) {

// return res.status(403).json({

// success: false,

// message: 'Acesso não autorizado'

// });

// }

// // Verificar se a oficina existe e é do tipo correto

// const oficinaCheck = await new Promise((resolve, reject) => {

// db.query(`

// SELECT o.id, u.tipo

// FROM oficina o

// JOIN usuario u ON o.usuario\_id = u.id

// WHERE o.id = ? AND u.tipo = 'oficina'

// `, [oficina\_id], (err, results) => {

// if (err) reject(err);

// resolve(results);

// });

// });

// if (oficinaCheck.length === 0) {

// return res.status(404).json({

// success: false,

// message: 'Oficina não encontrada ou inválida'

// });

// }

// // Verificar se o veículo pertence ao cliente (se fornecido)

// if (veiculo\_id) {

// const veiculoCheck = await new Promise((resolve, reject) => {

// db.query('SELECT id FROM veiculo WHERE id = ? AND usuario\_id = ?',

// [veiculo\_id, cliente\_id], (err, results) => {

// if (err) reject(err);

// resolve(results);

// });

// });

// if (veiculoCheck.length === 0) {

// return res.status(404).json({

// success: false,

// message: 'Veículo não encontrado ou não pertence ao cliente'

// });

// }

// }

// // Gerar código único de agendamento

// const codigoConfirmacao = 'OS' + Date.now().toString().slice(-8);

// const query = `

// INSERT INTO agendamento

// (cliente\_id, oficina\_id, veiculo\_id, data\_agendamento, servicos, produtos, observacoes, codigo\_confirmacao, status)

// VALUES (?, ?, ?, ?, ?, ?, ?, ?, 'pendente')

// `;

// const result = await new Promise((resolve, reject) => {

// db.query(query, [

// cliente\_id,

// oficina\_id,

// veiculo\_id || null,

// data\_agendamento,

// JSON.stringify(servicos || {}),

// JSON.stringify(produtos || {}),

// observacoes || '',

// codigoConfirmacao

// ], (err, result) => {

// if (err) reject(err);

// resolve(result);

// });

// });

// res.json({

// success: true,

// message: 'Agendamento criado com sucesso',

// agendamento\_id: result.insertId,

// codigo\_confirmacao: codigoConfirmacao

// });

// } catch (error) {

// console.error('Erro no agendamento:', error);

// res.status(500).json({

// success: false,

// message: 'Erro interno do servidor'

// });

// }

// });

// Rota para buscar agendamentos do usuário

app.get('/api/agendamentos/usuario/:userId', authenticateToken, (req, res) => {

const { userId } = req.params;

// Verificar se o usuário está tentando acessar seus próprios agendamentos

if (parseInt(userId) !== req.user.id) {

return res.status(403).json({

success: false,

message: 'Acesso não autorizado'

});

}

const query = `

SELECT a.\*, o.nome as oficina\_nome, o.endereco as oficina\_endereco,

v.placa, m.nome as modelo\_nome, ma.nome as marca\_nome

FROM agendamento a

JOIN oficina o ON a.oficina\_id = o.id

LEFT JOIN veiculo v ON a.veiculo\_id = v.id

LEFT JOIN modelo\_ano moano ON v.modelo\_ano\_id = moano.id

LEFT JOIN modelo m ON moano.modelo\_id = m.id

LEFT JOIN marca ma ON m.marca\_id = ma.id

WHERE a.cliente\_id = ?

ORDER BY a.data\_agendamento DESC

`;

db.query(query, [userId], (err, results) => {

if (err) {

console.error('Erro ao buscar agendamentos:', err);

return res.status(500).json({

success: false,

message: 'Erro interno do servidor'

});

}

// Parse dos dados JSON armazenados como texto

const agendamentos = results.map(ag => ({

...ag,

servicos: JSON.parse(ag.servicos || '{}'),

produtos: JSON.parse(ag.produtos || '{}')

}));

res.json({

success: true,

agendamentos: agendamentos

});

});

});

// Rota para buscar agendamentos da oficina

app.get('/api/agendamentos/oficina/:oficinaId', authenticateToken, (req, res) => {

const { oficinaId } = req.params;

// Verificar se o usuário é dono da oficina

const verificarOficina = `

SELECT id FROM oficina WHERE id = ? AND usuario\_id = ?

`;

db.query(verificarOficina, [oficinaId, req.user.id], (err, oficinaResults) => {

if (err) {

console.error('Erro ao verificar oficina:', err);

return res.status(500).json({

success: false,

message: 'Erro interno do servidor'

});

}

if (oficinaResults.length === 0) {

return res.status(403).json({

success: false,

message: 'Acesso não autorizado à esta oficina'

});

}

const query = `

SELECT a.\*, u.nome as cliente\_nome, u.telefone as cliente\_telefone,

v.placa, m.nome as modelo\_nome, ma.nome as marca\_nome

FROM agendamento a

JOIN usuario u ON a.cliente\_id = u.id

LEFT JOIN veiculo v ON a.veiculo\_id = v.id

LEFT JOIN modelo\_ano moano ON v.modelo\_ano\_id = moano.id

LEFT JOIN modelo m ON moano.modelo\_id = m.id

LEFT JOIN marca ma ON m.marca\_id = ma.id

WHERE a.oficina\_id = ?

ORDER BY a.data\_agendamento DESC

`;

db.query(query, [oficinaId], (err, results) => {

if (err) {

console.error('Erro ao buscar agendamentos da oficina:', err);

return res.status(500).json({

success: false,

message: 'Erro interno do servidor'

});

}

// Parse dos dados JSON armazenados como texto

const agendamentos = results.map(ag => ({

...ag,

servicos: JSON.parse(ag.servicos || '{}'),

produtos: JSON.parse(ag.produtos || '{}')

}));

res.json({

success: true,

agendamentos: agendamentos

});

});

});

});

// Rota de teste para verificar a tabela agendamento\_simples

app.get('/api/test/agendamento-table', (req, res) => {

db.query('SELECT COUNT(\*) as total FROM agendamento\_simples', (err, results) => {

if (err) {

return res.status(500).json({

success: false,

error: err.message

});

}

res.json({

success: true,

total\_registros: results[0].total,

mensagem: 'Tabela agendamento\_simples está acessível'

});

});

});

module.exports.authenticateToken = authenticateToken;