# Estrutura

|  |  |  |
| --- | --- | --- |
| Caminho Completo | Nome do Arquivo | Tipo |
| C:\Users\slima\Desktop\fit\_plus\_v1 | fit\_plus\_v1 | Pasta raiz |
| C:\Users\slima\Desktop\fit\_plus\_v1\css | css | Pasta |
| C:\Users\slima\Desktop\fit\_plus\_v1\css\style.css | style.css | CSS |
| C:\Users\slima\Desktop\fit\_plus\_v1\images | images | Pasta |
| C:\Users\slima\Desktop\fit\_plus\_v1\images\logo\_fit\_plus.png | logo\_fit\_plus.png | Imagem (PNG) |
| C:\Users\slima\Desktop\fit\_plus\_v1\images\user-pd.png | user-pd.png | Imagem (PNG) |
| C:\Users\slima\Desktop\fit\_plus\_v1\js | js | Pasta |
| C:\Users\slima\Desktop\fit\_plus\_v1\js\register.js | register.js | JavaScript |
| C:\Users\slima\Desktop\fit\_plus\_v1\js\login.js | login.js | JavaScript |
| C:\Users\slima\Desktop\fit\_plus\_v1\js\recovery.js | recovery.js | JavaScript |
| C:\Users\slima\Desktop\fit\_plus\_v1\js\home.js | home.js | JavaScript |
| C:\Users\slima\Desktop\fit\_plus\_v1\node\_modules | node\_modules | Pasta (Node.js) |
| C:\Users\slima\Desktop\fit\_plus\_v1.env | .env | Configuração |
| C:\Users\slima\Desktop\fit\_plus\_v1.gitigonore | .gitigonore | Git |
| C:\Users\slima\Desktop\fit\_plus\_v1\clean-server.bat | clean-server.bat | Script BAT |
| C:\Users\slima\Desktop\fit\_plus\_v1\db.js | db.js | JavaScript |
| C:\Users\slima\Desktop\fit\_plus\_v1\package.json | package.json | Node Package |
| C:\Users\slima\Desktop\fit\_plus\_v1\register.html | register.html | HTML |
| C:\Users\slima\Desktop\fit\_plus\_v1\login.html | login.html | HTML |
| C:\Users\slima\Desktop\fit\_plus\_v1\home.html | home.html | HTML |
| C:\Users\slima\Desktop\fit\_plus\_v1\recovery.html | recovery.html | HTML |
| C:\Users\slima\Desktop\fit\_plus\_v1\index.html | index.html | HTML |
| C:\Users\slima\Desktop\fit\_plus\_v1\server.js | server.js | JavaScript (Node) |
| "C:\Users\slima\Desktop\fit\_plus\_v1\food\_mod.html" | food\_mod.html | HTML |
| "C:\Users\slima\Desktop\fit\_plus\_v1\js\food\_mod.js" | food\_mod.js | JavaScript(Node) |
| "C:\Users\slima\Desktop\fit\_plus\_v1\images\icn\_alerta.png" | icn\_alerta.png | Imagem (PNG) |
| "C:\Users\slima\Desktop\fit\_plus\_v1\images\icn\_verificado.png" | icn\_verificado.png | Imagem (PNG) |

O site está publicado, sendo:

|  |  |  |  |
| --- | --- | --- | --- |
| Componente | Tecnologia | Plataforma | Detalhes |
| Frontend | HTML5, CSS3 e JS | Git | Vanilla JavaScript sem frameworks |
| Backend | Node.js + Express | Render | Versão 4.17.3 |
| Banco de Dados | PostgreSQL | Neon.tech | Hospedado na nuvem com Neon.tech |
| E-mails | Brevo API | Brevo | Plano gratuito (300 e-mails/dia) |

# Arquivos e Seus Códigos

## HTML

### index.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>Fit+ | Seu parceiro de fitness</title>      <link rel="stylesheet" href="css/style.css">      <link rel="preconnect" href="https://cdnjs.cloudflare.com">      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">  </head>  <body>      <div class="container index-container">          <header class="header">              <img src="images/logo\_fit\_plus.png" alt="Fit+ Logo" class="logo">              <h1>Bem-vindo ao Fit+</h1>              <p class="welcome-text">Sua jornada para uma vida mais saudável começa aqui</p>          </header>          <div class="index-actions">              <a href="login.html" class="btn-index btn-login">                  <i class="fas fa-sign-in-alt"></i> Entrar              </a>                <a href="register.html" class="btn-index btn-register">                  <i class="fas fa-user-plus"></i> Criar Conta              </a>                <button class="btn-index btn-support">                  <i class="fas fa-headset"></i> Falar com Suporte              </button>          </div>          <div class="index-features">              <div class="feature">                  <i class="fas fa-chart-line"></i>                  <h3>Acompanhe seu progresso</h3>                  <p>Monitore suas métricas e veja sua evolução</p>              </div>                <div class="feature">                  <i class="fas fa-dumbbell"></i>                  <h3>Personalize os seus treinos</h3>                  <p>Planos adaptados ao seu nível e objetivos</p>              </div>                <div class="feature">                  <i class="fas fa-apple-alt"></i>                  <h3>Organização da sua dieta</h3>                  <p>Acompanhamento da sua alimentação diária </p>              </div>               <div class="feature">                  <i class="fas fa-tasks"></i>                  <h3>Crie e monitore seus projetos</h3>                  <p>Foco no seu bem-estar físico e mental</p>              </div>            </div>      </div>      <div id="support-modal" class="support-modal support-hidden">          <div class="support-modal-content">              <span class="support-close-btn">&times;</span>              <h2>Suporte Fit+</h2>              <p>Em breve você poderá falar com nosso suporte por aqui!</p>          </div>      </div>      <script>          const supportModal = document.getElementById('support-modal');          const supportCloseBtn = document.querySelector('.support-close-btn');          const supportBtn = document.querySelector('.btn-support');          supportBtn.addEventListener('click', () => {              supportModal.classList.remove('support-hidden');          });          supportCloseBtn.addEventListener('click', () => {              supportModal.classList.add('support-hidden');          });          window.addEventListener('click', (event) => {              if (event.target === supportModal) {                  supportModal.classList.add('support-hidden');              }          });      </script>  </body>  </html> |

### login.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>Fit+ | Login</title>  <link rel="stylesheet" href="css/style.css">  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">  </head>  <body>  <div class="container">  <header class="header">  <img src="images/logo\_fit\_plus.png" alt="Fit+ Logo" class="logo">  <h1>Acesse sua conta</h1>  </header>  <form id="loginForm" class="form">  <div class="form-group">  <label for="username">Usuário</label>  <input type="text" id="username" name="username" placeholder="Digite seu e-mail" required>  </div>  <div class="form-group">  <label for="loginPassword">Senha</label>  <div class="password-input">  <input type="password" id="loginPassword" name="password" placeholder="Digite sua senha" required>  <i class="fas fa-eye-slash toggle-password"></i>  </div>  </div>  <button type="submit" class="btn-login">Entrar</button>  <div class="login-actions">  <a href="recovery.html" class="forgot-password">Esqueci minha senha</a>  </div>  <div class="register-option">  <p>Não tem uma conta? <a href="register.html" class="btn-register-link">Cadastre-se</a></p>  </div>  </form>  </div>  <script src="js/login.js"></script>  </body>  </html> |

### register.html

|  |
| --- |
| register.html |
| <!DOCTYPE html>  <html lang="pt-BR">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Fit+ | Cadastro</title>  <!-- Link para o arquivo CSS -->  <link rel="stylesheet" href="css/style.css">  <!-- Link para ícones (usando Font Awesome gratuito) -->  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">  </head>  <body>  <div class="container">  <!-- Cabeçalho com logo -->  <header class="header">  <img src="images/logo\_fit\_plus.png" alt="Fit+ Logo" class="logo">  <h1>Crie sua conta</h1>  </header>  <!-- Formulário de cadastro -->  <form id="registerForm" class="form" action="javascript:void(0);">  <!-- Grupo de campo: E-mail -->  <div class="form-group">  <label for="email">E-mail</label>  <input type="email" id="email" name="email" placeholder="Digite seu e-mail" required>  <span class="error-message" id="email-error"></span>  </div>  <!-- Grupo de campo: Nome e Sobrenome -->  <div class="name-group">  <div class="form-group half-width">  <label for="firstName">Nome</label>  <input type="text" id="firstName" name="firstName" placeholder="Digite seu nome" required>  <span class="error-message" id="firstName-error"></span>  </div>  <div class="form-group half-width">  <label for="lastName">Sobrenome</label>  <input type="text" id="lastName" name="lastName" placeholder="Digite seu sobrenome" required>  <span class="error-message" id="lastName-error"></span>  </div>  </div>  <!-- Grupo de campo: Data de Nascimento -->  <div class="form-group">  <label for="birthDate">Data de Nascimento</label>  <input type="date" id="birthDate" name="birthDate" required>  <span class="error-message" id="birthDate-error"></span>  </div>  <!-- Grupo de campo: Sexo -->  <div class="form-group">  <label for="gender">Sexo</label>  <select id="gender" name="gender" required>  <option value="" disabled selected>Selecione seu sexo</option>  <option value="feminino">Feminino</option>  <option value="masculino">Masculino</option>  <option value="outros">Outros</option>  </select>  <span class="error-message" id="gender-error"></span>  </div>  <!-- Grupo de campo: Telefone -->  <div class="form-group">  <label for="phone">Telefone</label>  <input type="tel" id="phone" name="phone" placeholder="(00) 00000-0000" required>  <span class="error-message" id="phone-error"></span>  </div>  <!-- Grupo de campo: Senha -->  <div class="form-group">  <label for="password">Senha</label>  <div class="password-input">  <input type="password" id="password" name="password" placeholder="Digite sua senha" required>  <i class="fas fa-eye toggle-password"></i>  </div>  <!-- Requisitos da senha -->  <div class="password-requirements">  <p id="length" class="requirement">Mínimo 8 caracteres</p>  <p id="uppercase" class="requirement">Pelo menos 1 letra maiúscula</p>  <p id="number" class="requirement">Pelo menos 1 número</p>  <p id="special" class="requirement">Pelo menos 1 caractere especial</p>  </div>  <span class="error-message" id="password-error"></span>  </div>  <!-- Grupo de campo: Confirmação de Senha -->  <div class="form-group">  <label for="confirmPassword">Confirme sua senha</label>  <div class="password-input">  <input type="password" id="confirmPassword" name="confirmPassword" placeholder="Confirme sua senha" required>  <i class="fas fa-eye toggle-password"></i>  </div>  <span class="error-message" id="confirmPassword-error"></span>  </div>  <!-- Botão de Registro -->  <button type="submit" class="btn-register">Registrar</button>  </form>  <div class="back-container">  <a href="login.html" id="backToLogin" class="btn-back">  <i class="fas fa-chevron-left"></i> Voltar ao Login  </a>  </div>  <!-- Modal de validação por e-mail -->  <div id="validationModal" class="modal">  <div class="modal-content">  <span class="close-modal">&times;</span>  <h2>Validação de E-mail</h2>  <p>Enviamos um código de 6 dígitos para <span id="userEmail"></span></p>    <div class="form-group">  <label for="validationCode">Código de Validação</label>  <input type="text" id="validationCode" placeholder="Digite o código recebido" maxlength="6">  <span class="error-message" id="code-error"></span>  </div>    <button id="validateBtn" class="btn-register">Validar</button>  <button id="resendBtn" class="btn-resend" disabled>Reenviar Código (<span id="countdown">60</span>s)</button>  </div>  </div>  </div>  <!-- Link para o arquivo JavaScript -->  <script src="js/register.js"></script>  </body>  </html> |

### home.html

|  |
| --- |
| home.html |
| <!DOCTYPE html>  <html lang="pt-BR">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Fit+ | Home</title>  <link rel="stylesheet" href="css/style.css">  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">  </head>  <body>  <!-- Barra superior -->  <header class="top-bar">  <img src="images/logo\_fit\_plus.png" alt="Fit+ Logo" class="logo">  <button id="menuButton" class="menu-button">  <i class="fas fa-bars"></i>  </button>  </header>  <!-- Painel do menu (inicialmente oculto) -->  <div id="menuPanel" class="menu-panel">  <div class="menu-content">  <!-- Foto de perfil -->  <img id="profilePic" src="images/user-pd.png" alt="Foto do usuário" class="profile-pic">    <!-- Botão para alterar foto -->  <button id="changePicBtn" class="btn-change-photo">Alterar Foto</button>    <input type="file" id="profilePicInput" accept="image/\*" style="display:none;">  <!-- Mensagem de boas-vindas -->  <h3 id="welcomeMessage">Bem-vindo, <span id="userName">Usuário</span></h3>    <!-- Botão Sair -->  <button id="logoutBtn" class="btn-logout">Sair</button>  </div>  </div>  <!-- Conteúdo principal (em desenvolvimento) -->  <div class="container">  <div class="content">  <p>Área logada - Conteúdo em desenvolvimento</p>  </div>  </div>  <!-- Scripts -->  <script src="js/home.js"></script>  </body>  </html> |

### recovery.html

|  |
| --- |
| recovery.html |
| <!DOCTYPE html>  <html lang="pt-BR">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Fit+ | Recuperação de Senha</title>  <link rel="stylesheet" href="css/style.css">  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">  </head>  <body>  <div class="container">  <header class="header">  <img src="images/logo\_fit\_plus.png" alt="Fit+ Logo" class="logo">  <h1>Redefinir Senha</h1>  </header>  <form id="recoveryForm" class="form">  <!-- Campo E-mail -->  <div class="form-group">  <label for="recoveryEmail">E-mail</label>  <input type="email" id="recoveryEmail" name="email" placeholder="Digite seu e-mail cadastrado" required>  <span class="error-message" id="email-error"></span>  </div>  <!-- Botão para enviar código -->  <button type="button" id="sendCodeBtn" class="btn-login">Enviar Código de Validação</button>  <!-- Campos de senha (inicialmente ocultos) -->  <div id="passwordFields" style="display: none;">  <div class="form-group">  <label for="newPassword">Nova Senha</label>  <div class="password-input">  <input type="password" id="newPassword" name="newPassword" placeholder="Digite sua nova senha" disabled>  <i class="fas fa-eye-slash toggle-password"></i>  </div>  <div class="password-requirements">  <p id="length" class="requirement">Mínimo 8 caracteres</p>  <p id="uppercase" class="requirement">Pelo menos 1 letra maiúscula</p>  <p id="number" class="requirement">Pelo menos 1 número</p>  <p id="special" class="requirement">Pelo menos 1 caractere especial</p>  </div>  <span class="error-message" id="password-error"></span>  </div>  <div class="form-group">  <label for="confirmNewPassword">Confirme a Nova Senha</label>  <div class="password-input">  <input type="password" id="confirmNewPassword" name="confirmNewPassword" placeholder="Repita sua nova senha" disabled>  <i class="fas fa-eye-slash toggle-password"></i>  </div>  <span class="error-message" id="confirmPassword-error"></span>  </div>  <button type="submit" id="resetBtn" class="btn-login" disabled>Redefinir Senha</button>  </div>  <!-- Botão Voltar -->  <div class="back-container">  <a href="login.html" class="btn-back">  <i class="fas fa-chevron-left"></i> Voltar ao Login  </a>  </div>  </form>  </div>  <!-- Modal de Validação (igual ao de cadastro) -->  <div id="validationModal" class="modal">  <div class="modal-content">  <span class="close-modal">&times;</span>  <h2>Validação de E-mail</h2>  <p>Enviamos um código de 6 dígitos para <span id="userEmail"></span></p>    <div class="form-group">  <label for="validationCode">Código de Validação</label>  <input type="text" id="validationCode" placeholder="Digite o código recebido" maxlength="6">  <span class="error-message" id="code-error"></span>  </div>    <button id="validateBtn" class="btn-register">Validar</button>  <button id="resendBtn" class="btn-resend" disabled>Reenviar Código (<span id="countdown">60</span>s)</button>  </div>  </div>  <!-- Scripts -->  <script src="js/recovery.js"></script>  </body>  </html> |

### food\_mod.html

|  |
| --- |
| food\_mod.html |
| <!DOCTYPE html>  <html lang="pt-BR">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Fit+ | Alimentos</title>  <link rel="stylesheet" href="css/style.css">  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">  </head>  <body>  <!-- Barra superior -->  <header class="top-bar">  <img src="images/logo\_fit\_plus.png" alt="Fit+ Logo" class="logo">  <button id="menuButton" class="menu-button">  <i class="fas fa-bars"></i>  </button>  </header>  <!-- Container principal -->  <div class="food-mod-container">  <!-- Cabeçalho com busca -->  <div class="food-mod-header">  <div class="food-search-container">  <input type="text" class="food-search-input" placeholder="Pesquise o alimento desejado">  <button class="food-add-btn" title="Cadastre um novo alimento">  <i class="fas fa-plus-circle"></i>  </button>  </div>  </div>  <!-- Tabela de alimentos -->  <div class="food-table-container">  <table class="food-table">  <thead>  <tr>  <th class="text-left">Item Alimentício</th>  <th class="text-left">Marca</th>  <th class="text-left">Modo de Preparo</th>  <th class="text-left">Grupo Alimentar</th>  <th class="text-center">Porção</th>  <th class="text-center">Kcal.</th>  <th></th>  </tr>  </thead>  <tbody>  <!-- Linhas serão preenchidas dinamicamente -->  </tbody>  </table>  <!-- Paginação -->  <div class="food-pagination">  <button class="pagination-btn" disabled>Anterior</button>  <span class="pagination-info">Página 1 de 0</span>  <button class="pagination-btn" disabled>Próximo</button>  </div>  </div>  </div>  <!-- Modal de Detalhes -->  <div id="foodDetailModal" class="food-modal">  <div class="food-modal-content">  <span class="food-modal-close">&times;</span>  <div class="food-modal-body"></div>  </div>  </div>  <!-- Modal de Cadastro -->  <div id="foodAddModal" class="food-modal">  <div class="food-modal-content">  <span class="food-modal-close">&times;</span>  <div class="food-modal-body"></div>  </div>  </div>  <!-- Scripts -->  <script src="js/home.js"></script>  <script src="js/food\_mod.js"></script>  </body>  </html> |

## Java Script

### server.js

|  |
| --- |
| server.js |
| // server.js adaptado para PostgreSQL (Neon)  process.on('SIGTERM', () => process.exit(0));  process.on('SIGINT', () => process.exit(0));  require('dotenv').config();  const express = require('express');  const bodyParser = require('body-parser');  const cors = require('cors');  const { pool } = require('./db');  const SibApiV3Sdk = require('sib-api-v3-sdk');  const multer = require('multer');  const upload = multer();  const defaultClient = SibApiV3Sdk.ApiClient.instance;  const apiKey = defaultClient.authentications['api-key'];  apiKey.apiKey = process.env.BREVO\_API\_KEY;  if (!apiKey.apiKey || apiKey.apiKey.length < 50) {  console.error('ERRO CRÍTICO: Chave API do Brevo inválida ou não configurada!');  process.exit(1);  }  const app = express();  const PORT = 3002;  app.use(cors());  app.use(bodyParser.json());  app.use(bodyParser.urlencoded({ extended: true }));  app.use(express.static(\_\_dirname));  function generateRandomCode() {  const min = 100000;  const max = 999999;  return Math.floor(Math.random() \* (max - min + 1)) + min;  }  async function sendValidationEmail(email, code) {  try {  const apiInstance = new SibApiV3Sdk.TransactionalEmailsApi();  const sendSmtpEmail = new SibApiV3Sdk.SendSmtpEmail();  sendSmtpEmail.subject = "🔑 Seu Código de Validação Fit+";  sendSmtpEmail.htmlContent = `<div><strong>${code}</strong></div>`;  sendSmtpEmail.sender = { email: "bex.diegolima@gmail.com", name: "Fit+ - Confirmação de Cadastro" };  sendSmtpEmail.to = [{ email }];  sendSmtpEmail.replyTo = { email: "bex.diegolima@gmail.com", name: "Suporte Fit+" };  await apiInstance.sendTransacEmail(sendSmtpEmail);  return true;  } catch (error) {  console.error('Erro no envio:', error);  return false;  }  }  app.post('/api/register', async (req, res) => {  try {  const { email, firstName, lastName, birthDate, gender, phone, password } = req.body;  if (!email || !/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)) {  return res.status(400).json({ success: false, message: 'E-mail inválido' });  }  const validationCode = generateRandomCode();  const client = await pool.connect();  try {  await client.query('BEGIN');  const result = await client.query(`  INSERT INTO tbl01\_users  (email\_user, name\_user, last\_name\_user, birth\_date\_user, gender\_user, phone\_user, password\_user, creator, reg\_code, account\_status)  VALUES ($1, $2, $3, $4, $5, $6, $7, 'system', $8, 'pendente') RETURNING user\_id`,  [email, firstName, lastName, birthDate, gender, phone, password, validationCode]  );  const userId = result.rows[0].user\_id;  const emailSent = await sendValidationEmail(email, validationCode);  if (!emailSent) throw new Error('Falha no envio do e-mail');  await client.query('COMMIT');  return res.status(201).json({ success: true, message: 'Registro realizado com sucesso.', userId, email });  } catch (error) {  await client.query('ROLLBACK');  throw error;  } finally {  client.release();  }  } catch (error) {  return res.status(500).json({ success: false, message: 'Erro durante o registro', error: error.message });  }  });  app.post('/api/validate', async (req, res) => {  try {  const { email, code } = req.body;  if (!email || !code || code.length !== 6) {  return res.status(400).json({ success: false, message: 'Dados de validação inválidos' });  }  const result = await pool.query(`  SELECT user\_id, reg\_code FROM tbl01\_users  WHERE email\_user = $1 AND account\_status = 'pendente'`, [email]  );  if (result.rows.length === 0) {  return res.status(404).json({ success: false, message: 'Usuário não encontrado ou código já validado' });  }  const user = result.rows[0];  if (code === user.reg\_code.toString()) {  await pool.query(`UPDATE tbl01\_users SET account\_status = 'active' WHERE user\_id = $1`, [user.user\_id]);  return res.json({ success: true, message: 'Cadastro validado com sucesso!' });  } else {  return res.status(400).json({ success: false, message: 'Código de validação incorreto' });  }  } catch (error) {  return res.status(500).json({ success: false, message: 'Erro durante a validação' });  }  });  app.post('/api/resend-code', async (req, res) => {  try {  const { email } = req.body;  if (!email || !/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)) {  return res.status(400).json({ success: false, message: 'E-mail inválido' });  }  const newCode = generateRandomCode();  const result = await pool.query(`  UPDATE tbl01\_users SET reg\_code = $1  WHERE email\_user = $2 AND account\_status = 'pendente'`, [newCode, email]);  if (result.rowCount === 0) {  return res.status(404).json({ success: false, message: 'E-mail não encontrado ou conta já ativada' });  }  const emailSent = await sendValidationEmail(email, newCode);  if (!emailSent) throw new Error('Falha no envio do e-mail');  return res.json({ success: true, message: 'Novo código enviado com sucesso' });  } catch (error) {  return res.status(500).json({ success: false, message: 'Erro durante o reenvio do código' });  }  });  app.post('/api/login', async (req, res) => {  try {  const { email, password } = req.body;  const result = await pool.query(`  SELECT user\_id, password\_user, account\_status FROM tbl01\_users WHERE email\_user = $1`, [email]  );  if (result.rows.length === 0) {  return res.status(401).json({ success: false, message: 'Usuário não encontrado' });  }  const user = result.rows[0];  if (password !== user.password\_user) {  return res.status(401).json({ success: false, message: 'Senha incorreta' });  }  if (user.account\_status !== 'active') {  return res.status(403).json({ success: false, message: 'Conta inativa. Por favor, contate o suporte.' });  }  return res.json({ success: true, message: 'Login realizado com sucesso' });  } catch (error) {  return res.status(500).json({ success: false, message: 'Erro durante o login' });  }  });  app.post('/api/send-recovery-code', async (req, res) => {  try {  const { email } = req.body;  const result = await pool.query(`SELECT user\_id FROM tbl01\_users WHERE email\_user = $1`, [email]);  if (result.rows.length === 0) {  return res.status(404).json({ success: false, message: 'E-mail não encontrado' });  }  const code = generateRandomCode();  await pool.query(`UPDATE tbl01\_users SET pass\_code = $1 WHERE email\_user = $2`, [code, email]);  const emailSent = await sendValidationEmail(email, `Seu código de recuperação: ${code}`);  if (!emailSent) throw new Error('Falha no envio do e-mail');  res.json({ success: true, message: 'Código enviado' });  } catch (error) {  res.status(500).json({ success: false, message: 'Erro no servidor' });  }  });  app.post('/api/validate-recovery-code', async (req, res) => {  try {  const { email, code } = req.body;  const result = await pool.query(  'SELECT user\_id FROM tbl01\_users WHERE email\_user = $1 AND pass\_code = $2',  [email, code]  );  if (result.rows.length === 0) {  return res.status(400).json({ success: false, message: 'Código inválido' });  }  res.json({ success: true });  } catch (error) {  res.status(500).json({ success: false, message: 'Erro na validação' });  }  });  app.post('/api/reset-password', async (req, res) => {  try {  const { email, newPassword } = req.body;  const result = await pool.query(  'SELECT password\_user FROM tbl01\_users WHERE email\_user = $1',  [email]  );  if (result.rows.length === 0) {  return res.status(404).json({ success: false, message: 'Usuário não encontrado' });  }  if (result.rows[0].password\_user === newPassword) {  return res.status(400).json({ success: false, message: 'A nova senha não pode ser igual à anterior' });  }  await pool.query(  'UPDATE tbl01\_users SET password\_user = $1, pass\_code = NULL WHERE email\_user = $2',  [newPassword, email]  );  res.json({ success: true, message: 'Senha atualizada' });  } catch (error) {  res.status(500).json({ success: false, message: 'Erro ao redefinir senha' });  }  });  app.post('/api/resend-recovery-code', async (req, res) => {  try {  const { email } = req.body;  const code = generateRandomCode();  await pool.query('UPDATE tbl01\_users SET pass\_code = $1 WHERE email\_user = $2', [code, email]);  const emailSent = await sendValidationEmail(email, `Seu novo código: ${code}`);  if (!emailSent) throw new Error('Falha no envio');  res.json({ success: true });  } catch (error) {  res.status(500).json({ success: false, message: 'Erro ao reenviar código' });  }  });  app.get('/api/get-user-data', async (req, res) => {  try {  const userEmail = req.query.email;  if (!userEmail) {  return res.status(400).json({ success: false, message: 'E-mail não fornecido' });  }  const result = await pool.query(  'SELECT name\_user FROM tbl01\_users WHERE email\_user = $1',  [userEmail]  );  if (result.rows.length === 0) {  return res.status(404).json({ success: false, message: 'Usuário não encontrado' });  }  res.json({ success: true, userName: result.rows[0].name\_user });  } catch (error) {  res.status(500).json({ success: false, message: 'Erro no servidor' });  }  });  app.post('/api/upload-profile-pic', upload.single('profilePic'), async (req, res) => {  try {  const userEmail = req.body.email;  if (!userEmail) {  return res.status(400).json({ success: false, message: 'E-mail do usuário é obrigatório' });  }  if (!req.file) {  return res.status(400).json({ success: false, message: 'Arquivo de imagem não enviado' });  }  const imgBuffer = req.file.buffer;  const result = await pool.query(  'UPDATE tbl01\_users SET profile\_pic = $1 WHERE email\_user = $2',  [imgBuffer, userEmail]  );  if (result.rowCount === 0) {  return res.status(404).json({ success: false, message: 'Usuário não encontrado' });  }  res.json({ success: true, message: 'Foto de perfil atualizada com sucesso' });  } catch (error) {  res.status(500).json({ success: false, message: 'Erro no servidor' });  }  });  app.get('/api/profile-pic', async (req, res) => {  try {  const userEmail = req.query.email;  if (!userEmail) {  return res.status(400).json({ success: false, message: 'E-mail do usuário não fornecido' });  }  const result = await pool.query(  'SELECT profile\_pic FROM tbl01\_users WHERE email\_user = $1',  [userEmail]  );  if (result.rows.length === 0 || !result.rows[0].profile\_pic) {  return res.status(404).json({ success: false, message: 'Foto de perfil não encontrada' });  }  const imgBuffer = result.rows[0].profile\_pic;  res.writeHead(200, {  'Content-Type': 'image/jpeg',  'Content-Length': imgBuffer.length  });  res.end(imgBuffer);  } catch (error) {  res.status(500).json({ success: false, message: 'Erro no servidor' });  }  });  app.listen(PORT, async () => {  console.log(`🚀 Servidor Fit+ rodando na porta ${PORT}`);  console.log(`🔑 Chave Brevo: ${apiKey.apiKey.substring(0, 6)}...${apiKey.apiKey.substring(apiKey.apiKey.length - 4)}`);  try {  const result = await pool.query('SELECT 1+1 AS result');  console.log('✅ Conexão com o banco de dados OK:', result.rows[0].result === 2 ? 'Sucesso' : 'Erro');  } catch (error) {  console.error('❌ Falha na conexão com o banco:', error);  }  }); |

### db.js

|  |
| --- |
| db.js |
| // Configuração do banco de dados PostgreSQL  const { Pool } = require('pg');  // Configurações de conexão  const pool = new Pool({  connectionString: 'postgresql://neondb\_owner:npg\_d1meWpC4sLvR@ep-divine-wave-achmpwge-pooler.sa-east-1.aws.neon.tech/neondb?sslmode=require&channel\_binding=require',  ssl: {  rejectUnauthorized: false // Necessário para conectar ao Neon  }  });  // Função para testar a conexão  async function testConnection() {  try {  const result = await pool.query('SELECT NOW()');  console.log('Conexão com o banco de dados estabelecida com sucesso! Hora atual:', result.rows[0].now);  return true;  } catch (error) {  console.error('Erro ao conectar ao banco de dados:', error);  return false;  }  }  // Exporta o pool de conexões e a função de teste  module.exports = {  pool,  testConnection  }; |

### home.js

|  |
| --- |
| home.js |
| document.addEventListener('DOMContentLoaded', function() {  const menuButton = document.getElementById('menuButton');  const menuPanel = document.getElementById('menuPanel');  const logoutBtn = document.getElementById('logoutBtn');  // 1. Alternar visibilidade do menu  menuButton.addEventListener('click', function(e) {  e.stopPropagation(); // Evita fechar ao clicar no botão  menuPanel.classList.toggle('active');  });  // 2. Fechar menu ao clicar fora  document.addEventListener('click', function(e) {  if (menuPanel.classList.contains('active') && !menuPanel.contains(e.target)) {  menuPanel.classList.remove('active');  }  });  // 3. Botão Sair  logoutBtn.addEventListener('click', function() {  const confirmLogout = confirm('Tem certeza que deseja sair?');  if (confirmLogout) {  localStorage.removeItem('userEmail'); // Limpa o e-mail salvo  window.location.href = 'login.html';  }  // Se cancelar, não faz nada e o usuário permanece logado  });  // 4. Carregar nome do usuário (simulação - substituir por chamada real na próxima etapa)  fetchUserData();  // Função para carregar a foto de perfil  async function loadProfilePic() {  const userEmail = localStorage.getItem('userEmail');  if (!userEmail) return;  try {  const response = await fetch(`https://fit-plus-backend.onrender.com/api/profile-pic?email=${encodeURIComponent(userEmail)}`);  if (response.ok) {  const blob = await response.blob();  const url = URL.createObjectURL(blob);  document.getElementById('profilePic').src = url;  } else {  // Mantém imagem padrão caso não tenha foto  console.log('Foto de perfil não encontrada, usando padrão');  }  } catch (error) {  console.error('Erro ao carregar foto de perfil:', error);  }  }  loadProfilePic();  // Abrir seletor de arquivo ao clicar no botão  const changePicBtn = document.getElementById('changePicBtn');  const profilePicInput = document.getElementById('profilePicInput');  changePicBtn.addEventListener('click', () => {  profilePicInput.click();  });  // Ao selecionar arquivo, enviar para o servidor  profilePicInput.addEventListener('change', async () => {  const file = profilePicInput.files[0];  if (!file) return;  const userEmail = localStorage.getItem('userEmail');  if (!userEmail) {  alert('Usuário não logado');  return;  }  const formData = new FormData();  formData.append('profilePic', file);  formData.append('email', userEmail);  try {  const response = await fetch('https://fit-plus-backend.onrender.com/api/upload-profile-pic', {  method: 'POST',  body: formData  });  const result = await response.json();  if (result.success) {  alert('Foto de perfil atualizada!');  loadProfilePic(); // Recarrega a foto no painel  } else {  alert('Erro: ' + (result.message || 'Não foi possível atualizar a foto'));  }  } catch (error) {  console.error('Erro ao enviar foto:', error);  alert('Erro ao enviar foto');  }  });  });  // Função simulada (será substituída por chamada real ao backend)  async function fetchUserData() {  try {  const userEmail = localStorage.getItem('userEmail'); // Obtém o e-mail salvo no login  if (!userEmail) {  throw new Error('E-mail do usuário não encontrado');  }  const response = await fetch(`https://fit-plus-backend.onrender.com/api/get-user-data?email=${encodeURIComponent(userEmail)}`);  const result = await response.json();  if (result.success) {  document.getElementById('userName').textContent = result.userName;  } else {  console.error('Erro ao carregar nome:', result.message);  document.getElementById('userName').textContent = 'Usuário';  }  } catch (error) {  console.error('Erro na requisição:', error);  document.getElementById('userName').textContent = 'Usuário';  }  } |

### login.js

|  |
| --- |
| login.js |
| document.addEventListener('DOMContentLoaded', function() {  // Elementos do formulário  const loginForm = document.getElementById('loginForm');  const togglePassword = document.querySelector('.toggle-password');  const passwordInput = document.getElementById('loginPassword');  const errorContainer = document.createElement('div');  errorContainer.className = 'error-message';  loginForm.appendChild(errorContainer);  // Toggle para mostrar/esconder senha  togglePassword.addEventListener('click', function() {  const type = passwordInput.getAttribute('type') === 'password' ? 'text' : 'password';  passwordInput.setAttribute('type', type);  this.classList.toggle('fa-eye');  this.classList.toggle('fa-eye-slash');  });  // Validação do login  loginForm.addEventListener('submit', async function(e) {  e.preventDefault();  errorContainer.textContent = '';    const username = document.getElementById('username').value.trim();  const password = passwordInput.value;  try {  const response = await fetch('https://fit-plus-backend.onrender.com/api/login', {  method: 'POST',  headers: { 'Content-Type': 'application/json' },  body: JSON.stringify({ email: username, password })  });  const result = await response.json();  if (result.success) {  localStorage.setItem('userEmail', document.getElementById('username').value.trim());  window.location.href = 'home.html';  } else {  errorContainer.textContent = result.message;  }  } catch (error) {  errorContainer.textContent = 'Erro ao conectar com o servidor';  console.error('Erro no login:', error);  }  });  }); |

### recovery.js

|  |
| --- |
| recovery.js |
| document.addEventListener('DOMContentLoaded', function() {  // Elementos do DOM  const recoveryForm = document.getElementById('recoveryForm');  const sendCodeBtn = document.getElementById('sendCodeBtn');  const passwordFields = document.getElementById('passwordFields');  const newPasswordInput = document.getElementById('newPassword');  const confirmPasswordInput = document.getElementById('confirmNewPassword');  const resetBtn = document.getElementById('resetBtn');  const validationModal = document.getElementById('validationModal');  const userEmailSpan = document.getElementById('userEmail');  const validationCodeInput = document.getElementById('validationCode');  const validateBtn = document.getElementById('validateBtn');  const resendBtn = document.getElementById('resendBtn');  const countdownSpan = document.getElementById('countdown');  const closeModal = document.querySelector('.close-modal');  const errorMessages = {  email: document.getElementById('email-error'),  password: document.getElementById('password-error'),  confirmPassword: document.getElementById('confirmPassword-error'),  code: document.getElementById('code-error')  };  // Variáveis de estado  let currentUserEmail = '';  let countdownInterval;  let canResend = false;  // Toggle para mostrar/esconder senha  document.querySelectorAll('.toggle-password').forEach(icon => {  icon.addEventListener('click', function() {  const input = this.previousElementSibling;  const type = input.getAttribute('type') === 'password' ? 'text' : 'password';  input.setAttribute('type', type);  this.classList.toggle('fa-eye-slash');  this.classList.toggle('fa-eye');  });  });  // Validação da força da senha (em tempo real)  newPasswordInput.addEventListener('input', function() {  const value = this.value;  document.getElementById('length').classList.toggle('met', value.length >= 8);  document.getElementById('uppercase').classList.toggle('met', /[A-Z]/.test(value));  document.getElementById('number').classList.toggle('met', /[0-9]/.test(value));  document.getElementById('special').classList.toggle('met', /[^A-Za-z0-9]/.test(value));  });  // 1. Enviar código de validação  sendCodeBtn.addEventListener('click', async function() {  const email = document.getElementById('recoveryEmail').value.trim();  errorMessages.email.textContent = '';  if (!email || !/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)) {  errorMessages.email.textContent = 'E-mail inválido';  return;  }  try {  const response = await fetch('https://fit-plus-backend.onrender.com/api/send-recovery-code', {  method: 'POST',  headers: { 'Content-Type': 'application/json' },  body: JSON.stringify({ email })  });  const result = await response.json();  if (result.success) {  currentUserEmail = email;  userEmailSpan.textContent = email;  validationModal.style.display = 'block';  startCountdown();  } else {  errorMessages.email.textContent = result.message || 'E-mail não encontrado';  }  } catch (error) {  console.error('Erro:', error);  errorMessages.email.textContent = 'Erro ao enviar código';  }  });  // 2. Validar código recebido  validateBtn.addEventListener('click', async function() {  const code = validationCodeInput.value.trim();  errorMessages.code.textContent = '';  if (!code || code.length !== 6 || !/^\d+$/.test(code)) {  errorMessages.code.textContent = 'Código inválido';  return;  }  try {  const response = await fetch('https://fit-plus-backend.onrender.com/api/validate-recovery-code', {  method: 'POST',  headers: { 'Content-Type': 'application/json' },  body: JSON.stringify({ email: currentUserEmail, code })  });  const result = await response.json();  if (result.success) {  validationModal.style.display = 'none';  passwordFields.style.display = 'block';  newPasswordInput.disabled = false;  confirmPasswordInput.disabled = false;  resetBtn.disabled = false;  } else {  errorMessages.code.textContent = result.message || 'Código incorreto';  }  } catch (error) {  console.error('Erro:', error);  errorMessages.code.textContent = 'Erro na validação';  }  });  // 3. Redefinir senha  recoveryForm.addEventListener('submit', async function(e) {  e.preventDefault();  const newPassword = newPasswordInput.value;  const confirmPassword = confirmPasswordInput.value;  // Validações  if (newPassword !== confirmPassword) {  errorMessages.confirmPassword.textContent = 'As senhas não coincidem';  return;  }  if (newPassword.length < 8 || !/[A-Z]/.test(newPassword) || !/[0-9]/.test(newPassword) || !/[^A-Za-z0-9]/.test(newPassword)) {  errorMessages.password.textContent = 'Senha não atende aos requisitos';  return;  }  try {  const response = await fetch('https://fit-plus-backend.onrender.com/api/reset-password', {  method: 'POST',  headers: { 'Content-Type': 'application/json' },  body: JSON.stringify({ email: currentUserEmail, newPassword })  });  const result = await response.json();  if (result.success) {  alert('Senha redefinida com sucesso!');  window.location.href = 'login.html';  } else {  errorMessages.password.textContent = result.message || 'Erro ao redefinir senha';  }  } catch (error) {  console.error('Erro:', error);  errorMessages.password.textContent = 'Erro no servidor';  }  });  // Funções auxiliares (countdown e reenvio)  function startCountdown() {  let seconds = 60;  resendBtn.disabled = true;  canResend = false;    clearInterval(countdownInterval);  countdownInterval = setInterval(() => {  seconds--;  countdownSpan.textContent = seconds;    if (seconds <= 0) {  clearInterval(countdownInterval);  resendBtn.disabled = false;  canResend = true;  countdownSpan.textContent = '0';  }  }, 1000);  }  resendBtn.addEventListener('click', async function() {  if (!canResend) return;  try {  resendBtn.disabled = true;  const response = await fetch('https://fit-plus-backend.onrender.com/api/resend-recovery-code', {  method: 'POST',  headers: { 'Content-Type': 'application/json' },  body: JSON.stringify({ email: currentUserEmail })  });  const result = await response.json();  if (result.success) {  startCountdown();  alert('Novo código enviado!');  } else {  alert(result.message);  resendBtn.disabled = false;  }  } catch (error) {  console.error('Erro:', error);  alert('Erro ao reenviar código');  resendBtn.disabled = false;  }  });  closeModal.addEventListener('click', () => {  validationModal.style.display = 'none';  clearInterval(countdownInterval);  });  }); |

### register.js

|  |
| --- |
| register.js |
| document.addEventListener('DOMContentLoaded', function() {  // Elementos do formulário principal  const registerForm = document.getElementById('registerForm');  const passwordInput = document.getElementById('password');  const confirmPasswordInput = document.getElementById('confirmPassword');  const togglePasswordButtons = document.querySelectorAll('.toggle-password');  const passwordRequirements = {  length: document.getElementById('length'),  uppercase: document.getElementById('uppercase'),  number: document.getElementById('number'),  special: document.getElementById('special')  };  // Limpeza segura do formulário  document.getElementById('backToLogin').addEventListener('click', (e) => {  if (registerForm.elements.length > 0 && Array.from(registerForm.elements).some(el => el.value)) {  if (!confirm('Deseja descartar todos os dados inseridos?')) {  e.preventDefault();  }  }  // O redirecionamento será feito naturalmente pelo href do link  });  // Elementos do modal de validação  const validationModal = document.getElementById('validationModal');  const userEmailSpan = document.getElementById('userEmail');  const validationCodeInput = document.getElementById('validationCode');  const validateBtn = document.getElementById('validateBtn');  const resendBtn = document.getElementById('resendBtn');  const countdownSpan = document.getElementById('countdown');  const closeModal = document.querySelector('.close-modal');  // Variáveis de estado  let currentUserEmail = '';  let countdownInterval;  let canResend = false;  // Máscara para telefone  const phoneInput = document.getElementById('phone');  phoneInput.addEventListener('input', formatPhone);  // Toggle para mostrar/esconder senha  togglePasswordButtons.forEach(button => {  button.addEventListener('click', togglePasswordVisibility);  });  // Validação em tempo real da senha  passwordInput.addEventListener('input', validatePasswordStrength);  // Fechar modal  closeModal.addEventListener('click', () => {  validationModal.style.display = 'none';  clearInterval(countdownInterval);  });  // Validação do formulário principal  registerForm.addEventListener('submit', handleRegisterSubmit);  // Validação do código  validateBtn.addEventListener('click', handleValidation);  // Reenviar código  resendBtn.addEventListener('click', handleResendCode);  // Funções auxiliares  function formatPhone(e) {  let value = e.target.value.replace(/\D/g, '');  value = value.replace(/^(\d{2})(\d)/g, '($1) $2');  value = value.replace(/(\d)(\d{4})$/, '$1-$2');  e.target.value = value;  }  function togglePasswordVisibility() {  const input = this.previousElementSibling;  const type = input.getAttribute('type') === 'password' ? 'text' : 'password';  input.setAttribute('type', type);  this.classList.toggle('fa-eye-slash');  }  function validatePasswordStrength() {  const value = this.value;  passwordRequirements.length.classList.toggle('met', value.length >= 8);  passwordRequirements.uppercase.classList.toggle('met', /[A-Z]/.test(value));  passwordRequirements.number.classList.toggle('met', /[0-9]/.test(value));  passwordRequirements.special.classList.toggle('met', /[^A-Za-z0-9]/.test(value));  }  function clearErrors() {  document.querySelectorAll('.error-message').forEach(el => el.textContent = '');  }  function validateForm() {  let isValid = true;  clearErrors();  // Validação do e-mail  const email = document.getElementById('email').value;  if (!email) {  document.getElementById('email-error').textContent = 'E-mail é obrigatório';  isValid = false;  } else if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)) {  document.getElementById('email-error').textContent = 'E-mail inválido';  isValid = false;  }  // Validação do nome  const firstName = document.getElementById('firstName').value;  if (!firstName) {  document.getElementById('firstName-error').textContent = 'Nome é obrigatório';  isValid = false;  }  // Validação do sobrenome  const lastName = document.getElementById('lastName').value;  if (!lastName) {  document.getElementById('lastName-error').textContent = 'Sobrenome é obrigatório';  isValid = false;  }  // Validação da data de nascimento  const birthDate = document.getElementById('birthDate').value;  if (!birthDate) {  document.getElementById('birthDate-error').textContent = 'Data de nascimento é obrigatória';  isValid = false;  } else {  const birthDateObj = new Date(birthDate);  const today = new Date();  let age = today.getFullYear() - birthDateObj.getFullYear();  const monthDiff = today.getMonth() - birthDateObj.getMonth();    if (monthDiff < 0 || (monthDiff === 0 && today.getDate() < birthDateObj.getDate())) {  age--;  }    if (age < 18) {  document.getElementById('birthDate-error').textContent = 'Você deve ter pelo menos 18 anos';  isValid = false;  }  }  // Validação do sexo  const gender = document.getElementById('gender').value;  if (!gender) {  document.getElementById('gender-error').textContent = 'Sexo é obrigatório';  isValid = false;  }  // Validação do telefone  const phone = document.getElementById('phone').value;  if (!phone) {  document.getElementById('phone-error').textContent = 'Telefone é obrigatório';  isValid = false;  } else if (!/^\(\d{2}\) \d{5}-\d{4}$/.test(phone)) {  document.getElementById('phone-error').textContent = 'Telefone inválido';  isValid = false;  }  // Validação da senha  const password = passwordInput.value;  if (!password) {  document.getElementById('password-error').textContent = 'Senha é obrigatória';  isValid = false;  } else if (password.length < 8 || !/[A-Z]/.test(password) || !/[0-9]/.test(password) || !/[^A-Za-z0-9]/.test(password)) {  document.getElementById('password-error').textContent = 'Senha não atende aos requisitos';  isValid = false;  }  // Validação da confirmação de senha  const confirmPassword = confirmPasswordInput.value;  if (!confirmPassword) {  document.getElementById('confirmPassword-error').textContent = 'Confirmação de senha é obrigatória';  isValid = false;  } else if (password !== confirmPassword) {  document.getElementById('confirmPassword-error').textContent = 'Senhas não coincidem';  isValid = false;  }  return isValid;  }  async function handleRegisterSubmit(e) {  e.preventDefault();    if (!validateForm()) return;  const formData = {  email: document.getElementById('email').value,  firstName: document.getElementById('firstName').value,  lastName: document.getElementById('lastName').value,  birthDate: document.getElementById('birthDate').value,  gender: document.getElementById('gender').value,  phone: document.getElementById('phone').value,  password: passwordInput.value  };  try {  const response = await fetch('https://fit-plus-backend.onrender.com/api/register', {  method: 'POST',  headers: { 'Content-Type': 'application/json' },  body: JSON.stringify(formData)  });  const result = await response.json();  if (result.success) {  currentUserEmail = formData.email;  userEmailSpan.textContent = currentUserEmail;  validationModal.style.display = 'block';  startCountdown();  } else {  const errorField = result.message.includes('e-mail') ? 'email-error' : 'password-error';  document.getElementById(errorField).textContent = result.message;  }  } catch (error) {  console.error('Erro ao registrar:', error);  alert('Erro ao conectar com o servidor. Tente novamente.');  }  }  function startCountdown() {  let seconds = 60;  resendBtn.disabled = true;  canResend = false;    clearInterval(countdownInterval);  countdownInterval = setInterval(() => {  seconds--;  countdownSpan.textContent = seconds;    if (seconds <= 0) {  clearInterval(countdownInterval);  resendBtn.disabled = false;  canResend = true;  countdownSpan.textContent = '0';  }  }, 1000);  }  async function handleValidation() {  const code = validationCodeInput.value.trim();  const codeErrorElement = document.getElementById('code-error');    // Limpa erros anteriores  codeErrorElement.textContent = '';    // Validação básica do formato  if (!code || code.length !== 6 || !/^\d+$/.test(code)) {  codeErrorElement.textContent = 'Digite um código válido de 6 dígitos';  return;  }  try {  console.log('Validando código...');  const response = await fetch('https://fit-plus-backend.onrender.com/api/validate', {  method: 'POST',  headers: {  'Content-Type': 'application/json',  'Accept': 'application/json'  },  body: JSON.stringify({  email: currentUserEmail,  code  })  });  const result = await response.json();  if (!response.ok) {  throw new Error(result.message || 'Erro na validação');  }  if (result.success) {  alert('✅ Cadastro confirmado com sucesso!');  validationModal.style.display = 'none';  registerForm.reset();  clearInterval(countdownInterval);  // Redireciona para a página de login após 2 segundos  setTimeout(() => {  window.location.href = 'login.html';  }, 2000);  } else {  codeErrorElement.textContent = result.message;  }  } catch (error) {  console.error('Erro na validação:', error);  codeErrorElement.textContent = typeof error === 'string' ? error : 'Erro ao validar código';  }  }  async function handleResendCode() {  if (!canResend) return;  try {  resendBtn.disabled = true;  const response = await fetch('https://fit-plus-backend.onrender.com/api/resend-code', {  method: 'POST',  headers: { 'Content-Type': 'application/json' },  body: JSON.stringify({ email: currentUserEmail })  });  const result = await response.json();  if (result.success) {  startCountdown();  alert('Novo código enviado para seu e-mail!');  } else {  alert(result.message);  resendBtn.disabled = false;  }  } catch (error) {  console.error('Erro ao reenviar código:', error);  alert('Erro ao reenviar código');  resendBtn.disabled = false;  }  }  }); |

### food\_mod.js

|  |
| --- |
| food\_mod.js |
| document.addEventListener('DOMContentLoaded', function() {  // Elementos da página  const addBtn = document.querySelector('.food-add-btn');  const detailModal = document.getElementById('foodDetailModal');  const addModal = document.getElementById('foodAddModal');  const closeButtons = document.querySelectorAll('.food-modal-close');  const foodTableBody = document.querySelector('.food-table tbody');  // Inicialização da tabela vazia  function initializeEmptyTable() {  foodTableBody.innerHTML = '<tr class="empty-row"><td colspan="7">Nenhum alimento cadastrado</td></tr>';  }  // Abrir modal de cadastro  addBtn.addEventListener('click', () => {  addModal.style.display = 'block';  });  // Fechar modais  closeButtons.forEach(btn => {  btn.addEventListener('click', function() {  this.closest('.food-modal').style.display = 'none';  });  });  // Fechar ao clicar fora do modal  window.addEventListener('click', (event) => {  if (event.target.classList.contains('food-modal')) {  event.target.style.display = 'none';  }  });  // Inicializa a tabela vazia  initializeEmptyTable();  }); |

## CSS

### style.css

|  |
| --- |
| style.css |
| /\* Reset básico e estilos globais \*/  \* {  margin: 0;  padding: 0;  box-sizing: border-box;  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  }  body {  background-color: #f5f5f5;  color: #333;  line-height: 1.6;  }  /\* Container principal \*/  .container {  max-width: 500px;  margin: 0 auto;  padding: 25px;  background-color: white;  border-radius: 10px;  box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);  margin-top: 70px; /\* Ajustado para não sobrepor top-bar \*/  margin-bottom: 30px;  }  /\* Cabeçalho \*/  .header {  text-align: center;  margin-bottom: 20px;  }  .logo {  max-width: 150px;  margin-bottom: 15px;  }  .header h1 {  color: #2c3e50;  font-size: 24px;  }  /\* Estilos do formulário - Ajustado espaçamento \*/  .form {  display: flex;  flex-direction: column;  gap: 8px; /\* Reduzido de 20px para 15px \*/  }  .form-group {  display: flex;  flex-direction: column;  gap: 5px;  }  .form-group label {  font-weight: 600;  color: #2c3e50;  font-size: 14px;  }  .form-group input,  .form-group select {  padding: 12px 15px;  border: 1px solid #ddd;  border-radius: 5px;  font-size: 15px;  transition: border 0.3s;  }  .form-group input:focus,  .form-group select:focus {  border-color: #3498db;  outline: none;  }  /\* Grupo de nome (nome e sobrenome lado a lado) \*/  .name-group {  display: flex;  gap: 10px; /\* Reduzido de 15px para 10px \*/  }  .half-width {  flex: 1;  }  /\* Estilo para campos de senha - Ajustado ícone do olho \*/  .password-input {  position: relative;  }  .password-input input {  width: 100%;  padding-right: 40px;  }  .toggle-password {  position: absolute;  right: 15px;  top: 50%;  transform: translateY(-50%);  cursor: pointer;  color: #7f8c8d;  font-size: 18px;  }  /\* Ícone do olho com barra quando fechado \*/  .toggle-password::after {  content: '';  position: absolute;  left: 50%;  top: 50%;  width: 18px;  height: 2px;  background-color: #7f8c8d;  transform: translate(-50%, -50%) rotate(45deg);  opacity: 0;  transition: opacity 0.3s;  }  .toggle-password:not(.fa-eye-slash)::after {  opacity: 1;  }  /\* Requisitos da senha \*/  .password-requirements {  margin-top: 5px;  }  .requirement {  font-size: 12px;  color: #7f8c8d;  margin-bottom: 2px;  }  .requirement.met {  color: #27ae60;  }  /\* Mensagens de erro \*/  .error-message {  color: #e74c3c;  font-size: 12px;  height: 14px;  margin-top: 2px;  text-align: left; /\* Adicionado para alinhamento melhor \*/  }  /\* Botão de registro \*/  .btn-register {  background-color: #59798f;  color: white;  border: none;  padding: 14px;  border-radius: 5px;  font-size: 16px;  font-weight: 600;  cursor: pointer;  transition: background-color 0.3s;  margin-top: 5px; /\* Reduzido de 10px para 5px \*/  }  .btn-register:hover {  background-color: #2980b9;  }  /\* Botão Voltar - Estilo Moderno \*/  .back-container {  text-align: center;  margin-top: 15px;  }  .btn-back {  background: transparent;  border: 2px solid #3498db;  color: #3498db;  padding: 10px 20px;  border-radius: 25px; /\* Bordas arredondadas \*/  font-weight: 600;  cursor: pointer;  transition: all 0.3s;  display: inline-flex;  align-items: center;  gap: 8px;  }  .btn-back:hover {  background: #3498db10; /\* Fundo semi-transparente \*/  transform: translateY(-2px); /\* Efeito de levitar \*/  }  .btn-back i {  font-size: 14px;  }  /\* Responsividade para dispositivos móveis \*/  @media (max-width: 600px) {  .container {  margin: 15px;  padding: 15px;  }  .name-group {  flex-direction: column;  gap: 15px; /\* Reduzido de 20px para 15px \*/  }  .half-width {  width: 100%;  }  }  /\* Estilos para o modal de validação \*/  .modal {  display: none;  position: fixed;  z-index: 1000;  left: 0;  top: 0;  width: 100%;  height: 100%;  background-color: rgba(0, 0, 0, 0.7);  }  .modal-content {  background-color: #fff;  margin: 10% auto;  padding: 25px;  border-radius: 8px;  max-width: 450px;  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  position: relative;  }  .close-modal {  position: absolute;  right: 15px;  top: 15px;  font-size: 24px;  cursor: pointer;  color: #7f8c8d;  }  .btn-resend {  background-color: #f39c12;  color: white;  border: none;  padding: 12px;  border-radius: 5px;  font-size: 14px;  cursor: pointer;  transition: background-color 0.3s;  width: 100%;  margin-top: 10px;  }  .btn-resend:hover {  background-color: #e67e22;  }  .btn-resend:disabled {  background-color: #95a5a6;  cursor: not-allowed;  }  /\*PASSO 4 COMEÇA AQUI\*/  /\* Estilos específicos para login \*/  .btn-login {  background-color: #3498db;  color: white;  border: none;  padding: 14px;  border-radius: 5px;  font-size: 16px;  font-weight: 600;  cursor: pointer;  transition: background-color 0.3s;  margin-top: 15px;  width: 100%;  }  .btn-login:hover {  background-color: #27ae60;  }  .login-actions {  text-align: center;  margin: 5px 0 15px 0;  }  .forgot-password {  color: #3498db;  text-decoration: none;  font-size: 14px;  }  .forgot-password:hover {  text-decoration: underline;  }  .register-option {  text-align: center;  margin-top: 5px;  font-size: 14px;  }  .btn-register-link {  color: #3498db;  font-weight: 600;  text-decoration: none;  }  .btn-register-link:hover {  text-decoration: underline;  }  .error-message {  /\* Mantido para login e registro com estilo mais legível \*/  color: #e74c3c;  text-align: center;  margin: 1px 0;  font-size: 14px;  min-height: 20px;  }  /\* Estilos específicos para recovery.html \*/  #recoveryForm {  margin-top: 10px;  }  .btn-back {  margin-top: 15px;  }  /\* Estilos da barra superior \*/  .top-bar {  background-color: #f8f9fa; /\* Cor clean (cinza claro) \*/  padding: 20px 25px; /\* Mais espaçamento \*/  display: flex;  justify-content: space-between;  align-items: center;  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05); /\* Sombra mais suave \*/  position: fixed;  width: 100%;  top: 0;  left: 0;  z-index: 1000;  height: 80px; /\* Altura maior \*/  border-bottom: 1px solid #e9ecef; /\* Linha sutil \*/  }  .top-bar .logo {  height: 40px; /\* Tamanho aumentado \*/  }  .menu-button {  background: #ffffff;  border: 1px solid #e9ecef; /\* Borda sutil \*/  width: 45px;  height: 45px;  border-radius: 50%;  display: flex;  align-items: center;  justify-content: center;  cursor: pointer;  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);  transition: all 0.3s ease;  }  .menu-button:hover {  background: #f1f3f5; /\* Efeito hover suave \*/  }  .menu-button i {  color: #495057; /\* Cor mais neutra \*/  font-size: 18px;  }  /\* Painel do menu \*/  .menu-panel {  position: fixed;  top: 0;  right: -300px;  width: 300px;  height: 100vh;  background: white;  box-shadow: -2px 0 5px rgba(0, 0, 0, 0.1);  transition: right 0.3s ease;  z-index: 1001;  }  .menu-panel.active {  right: 0;  }  .menu-content {  padding: 20px;  display: flex;  flex-direction: column;  align-items: center;  gap: 20px;  margin-top: 70px;  }  .profile-pic {  width: 100px;  height: 100px;  border-radius: 50%;  object-fit: cover;  border: 3px solid #3498db;  }  .btn-change-photo {  background: #f8f9fa;  border: 1px dashed #3498db;  color: #3498db;  padding: 8px 15px;  border-radius: 5px;  cursor: pointer;  }  .btn-logout {  background: #e74c3c;  color: white;  border: none;  padding: 10px 20px;  border-radius: 5px;  cursor: pointer;  font-weight: 600;  }  /\* Ajuste para o container principal (evitar sobreposição com a top-bar) \*/  /\* Já ajustado margin-top: 70px na .container acima \*/  /\* Estilos específicos para a página inicial \*/  .index-container {  text-align: center;  max-width: 800px;  }  .welcome-text {  color: #7f8c8d;  margin-bottom: 30px;  font-size: 18px;  }  .index-actions {  display: flex;  flex-direction: column;  gap: 15px;  margin: 30px 0;  }  .btn-index {  padding: 15px;  border-radius: 5px;  font-size: 16px;  font-weight: 600;  cursor: pointer;  transition: all 0.3s;  display: flex;  align-items: center;  justify-content: center;  gap: 10px;  text-decoration: none;  }  .btn-login {  background-color: #3498db;  color: white;  border: none;  }  .btn-login:hover {  background-color: #3498db;  transform: translateY(-2px);  }  .btn-register {  background-color: #3498db;  color: white;  border: none;  }  .btn-register:hover {  background-color: #2980b9;  transform: translateY(-2px);  }  .btn-support {  background-color: #f39c12;  color: white;  border: none;  }  .btn-support:hover {  background-color: #e67e22;  transform: translateY(-2px);  }  .index-features {  display: grid;  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  gap: 20px;  margin-top: 40px;  }  .feature {  background: #f8f9fa;  padding: 20px;  border-radius: 8px;  transition: transform 0.3s;  }  .feature:hover {  transform: translateY(-5px);  box-shadow: 0 5px 15px rgba(0,0,0,0.1);  }  .feature i {  font-size: 2.5rem;  color: #3498db;  margin-bottom: 15px;  }  .feature h3 {  color: #2c3e50;  margin-bottom: 10px;  }  .feature p {  color: #7f8c8d;  font-size: 14px;  }  /\* Responsividade \*/  @media (max-width: 600px) {  .index-features {  grid-template-columns: 1fr;  }  }  /\* Modal suporte \*/  .support-modal {  position: fixed;  top: 0;  left: 0;  width: 100%;  height: 100%;  background-color: rgba(0, 0, 0, 0.6);  display: flex;  align-items: center;  justify-content: center;  z-index: 1000;  }  .support-hidden {  display: none;  }  .support-modal-content {  background-color: #fff;  padding: 2rem;  border-radius: 12px;  text-align: center;  max-width: 400px;  box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);  position: relative;  }  .support-close-btn {  position: absolute;  top: 10px;  right: 16px;  font-size: 24px;  font-weight: bold;  color: #333;  cursor: pointer;  }  /\* ========== ESTILOS DO MÓDULO DE ALIMENTOS ========== \*/  .food-mod-container {  width: 100%;  padding: 20px;  margin-top: 80px; /\* Espaço para a top-bar \*/  box-sizing: border-box;  }  .food-mod-header {  margin-bottom: 20px;  }  .food-search-container {  display: flex;  align-items: center;  gap: 15px;  max-width: 800px;  margin: 0 auto;  }  .food-search-input {  flex: 1;  padding: 12px 15px;  border: 1px solid #ddd;  border-radius: 5px;  font-size: 16px;  }  .food-add-btn {  background: none;  border: none;  color: #3498db;  font-size: 28px;  cursor: pointer;  transition: transform 0.2s;  }  .food-add-btn:hover {  transform: scale(1.1);  }  .food-table-container {  width: 100%;  overflow-x: auto;  }  .food-table {  width: 100%;  border-collapse: collapse;  margin-bottom: 20px;  }  .food-table th {  background-color: #f8f9fa;  padding: 12px 15px;  text-align: left;  border-bottom: 2px solid #ddd;  }  .food-table td {  padding: 12px 15px;  border-bottom: 1px solid #eee;  vertical-align: middle;  }  .food-table tr:hover {  background-color: #f5f5f5;  cursor: pointer;  }  .text-center {  text-align: center;  }  .food-status-icon {  text-align: center;  width: 40px;  }  .status-icon {  width: 20px;  height: 20px;  }  .food-pagination {  display: flex;  justify-content: center;  align-items: center;  gap: 20px;  margin-top: 20px;  }  .pagination-btn {  padding: 8px 16px;  background-color: #3498db;  color: white;  border: none;  border-radius: 4px;  cursor: pointer;  }  .pagination-btn:disabled {  background-color: #95a5a6;  cursor: not-allowed;  }  .pagination-info {  font-size: 14px;  color: #7f8c8d;  }  /\* ========== MELHORIAS NA TABELA ========== \*/  .food-table th.text-center,  .food-table td.text-center {  text-align: center;  }  .food-table th.text-left,  .food-table td.text-left {  text-align: left;  }  /\* ========== ESTILOS DOS MODAIS ========== \*/  .food-modal {  display: none;  position: fixed;  z-index: 1001;  left: 0;  top: 0;  width: 100%;  height: 100%;  background-color: rgba(0,0,0,0.7);  }  .food-modal-content {  background-color: #fefefe;  margin: 5% auto;  padding: 20px;  border-radius: 8px;  width: 60%;  max-width: 700px;  box-shadow: 0 4px 8px rgba(0,0,0,0.1);  position: relative;  }  .food-modal-close {  position: absolute;  right: 20px;  top: 10px;  font-size: 28px;  font-weight: bold;  color: #aaa;  cursor: pointer;  }  .food-modal-close:hover {  color: #333;  }  .food-modal-body {  padding: 20px 0;  }  /\* ========== ESTILOS PARA TABELA VAZIA ========== \*/  .food-table .empty-row td {  text-align: center;  padding: 30px;  color: #7f8c8d;  font-style: italic;  }  /\* ========== MELHORIAS NA PAGINAÇÃO ========== \*/  .food-pagination {  display: flex;  justify-content: center;  align-items: center;  gap: 20px;  margin-top: 20px;  }  .pagination-btn:disabled {  opacity: 0.5;  cursor: not-allowed;  } |

## Outros

### package.json

|  |
| --- |
| package.json |
| {  "name": "fit\_plus\_v1",  "version": "1.0.0",  "description": "Sistema Fit+",  "main": "server.js",  "scripts": {  "start": "node server.js",  "dev": "nodemon server.js"  },  "dependencies": {  "body-parser": "^1.20.2",  "cors": "^2.8.5",  "dotenv": "^17.2.0",  "express": "^4.21.2",  "multer": "^2.0.1",  "mysql2": "^3.6.0",  "pg": "^8.16.3",  "sib-api-v3-sdk": "^8.5.0"  },  "devDependencies": {  "nodemon": "^3.0.2"  }  } |

### clean-server.bat

|  |
| --- |
| Clean-server.bat |
| @echo off  taskkill /F /IM node.exe > nul 2>&1  npx kill-port 5005 5006 5007 5008 5009 > nul 2>&1  timeout 1 > nul  start npm start |

### .env

|  |
| --- |
| .env |
| BREVO\_API\_KEY=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 |

### .gitignore

|  |
| --- |
| .gitignore |
| # Node modules - não subir  node\_modules/  # Variáveis de ambiente  .env  # Logs  logs  \*.log  npm-debug.log\*  # Diretórios de build (caso use)  dist/  build/  # Sistemas operacionais  .DS\_Store  Thumbs.db  # Outros arquivos temporários  \*.tmp  \*.temp  # Coverage - se usar testes  coverage/ |

# Banco de Dados e Tabelas

## Tabela: tbl\_foods

### Estrutura da Tabela:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Tipo de Preenchimento** | **Valor Default** | **Tbl Aux** |
| id | PRIMARY KEY | Obrigatório | null |  |
| user\_registro | VARCHAR(100) | Automático | null | tbl01\_users |
| dt\_registro | DATETIME | Automático | null |  |
| dt\_atualizacao | DATETIME | Automático | null |  |
| item | VARCHAR(255) | Obrigatório | null |  |
| marca | VARCHAR(50) | Obrigatório | null |  |
| porcao\_base | NUMERIC(10,2) | Obrigatório | 100 |  |
| calorias\_kcal | NUMERIC(10,2) | Obrigatório | null |  |
| proteina\_gr | NUMERIC(10,2) | Obrigatório | null |  |
| carbo\_gr | NUMERIC(10,2) | Obrigatório | null |  |
| gorduras\_totais\_gr | NUMERIC(10,2) | Obrigatório | null |  |
| gorduras\_boas\_gr | NUMERIC(10,2) | Opcional | null |  |
| gorduras\_ruins\_gr | NUMERIC(10,2) | Opcional | null |  |
| omega\_tres\_mg | NUMERIC(10,2) | Opcional | null |  |
| fibras\_gr | NUMERIC(10,2) | Obrigatório | null |  |
| sodio\_mg | NUMERIC(10,2) | Opcional | null |  |
| colesterol\_mg | NUMERIC(10,2) | Opcional | null |  |
| acucares\_gr | NUMERIC(10,2) | Opcional | null |  |
| indice\_glicemico | NUMERIC(5,2) | Opcional | null |  |
| carga\_glicemica | NUMERIC(5,2) | Opcional | null |  |
| calcio\_mg | NUMERIC(10,2) | Opcional | null |  |
| ferro\_mg | NUMERIC(10,2) | Opcional | null |  |
| potassio\_mg | NUMERIC(10,2) | Opcional | null |  |
| magnesio\_mg | NUMERIC(10,2) | Opcional | null |  |
| zinco\_mg | NUMERIC(10,2) | Opcional | null |  |
| vitamina\_c\_mg | NUMERIC(10,2) | Opcional | null |  |
| vitamina\_a\_mcg | NUMERIC(10,2) | Opcional | null |  |
| vitamina\_d\_mcg | NUMERIC(10,2) | Opcional | null |  |
| vitamina\_b12\_mcg | NUMERIC(10,2) | Opcional | null |  |
| acido\_folico\_mcg | NUMERIC(10,2) | Opcional | null |  |
| teor\_alcoolico | NUMERIC(5,2) | Opcional | null |  |
| observacoes | TEXT | Opcional | null |  |
| densidade\_calorica | NUMERIC(10,2) | Automático | calorias\_kcal / porcao\_base\_gr |  |
| glutem | Verdadeiro/Falso | Opcional | null |  |
| alergicos\_comuns | VARCHAR(255) | Opcional | null |  |
| carga\_antioxidante | VARCHAR(50) | Opcional | null |  |
| img\_registro | VARCHAR(255) | Opcional | null |  |
| key\_words | VARCHAR(255) | Automático | Palavras do Item + Marca (Separado por virgula) |  |
| status\_registro | VARCHAR(20) | Automático | Ativo | tbl\_aux\_status\_register |
| tipo\_registro\_alimento | VARCHAR(50) | Automático | Alerta | tbl\_aux\_verification |
| tipo\_medida\_alimento | VARCHAR(2) | Automático | null | tbl\_aux\_tipo\_medida |
| modo\_preparo | VARCHAR(50) | Obrigatório | null | tbl\_aux\_modo\_preparo |
| grupo\_alimentar | VARCHAR(100) | Obrigatório | null | tbl\_aux\_grupo\_alimentar |
| categoria\_nutricional | VARCHAR(100) | Obrigatório | null | tbl\_aux\_categoria\_nutri |
| origem | VARCHAR(50) | Obrigatório | null | tbl\_aux\_origem\_alimentar |
| nivel\_processamento | VARCHAR(50) | Obrigatório | null | tbl\_aux\_processamento |
| acucar\_adicionados\_gr | NUMERIC(10,2) | Opcional | Null |  |
| vitamina\_e\_mcg | NUMERIC(10,2) | Opcional | Null |  |

## Tabela Auxiliar: tbl\_aux\_modo\_preparo

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **nome** | **detalhe** | **status** |
| 1 | Cru | Sem qualquer tipo de cocção ou alteração térmica | Ativo |
| 2 | Cozido | Preparado com água quente ou vapor | Ativo |
| 3 | Grelhado | Cozido diretamente sobre grelha ou chapa quente | Ativo |
| 4 | Assado | Preparado em forno convencional ou elétrico | Ativo |
| 5 | Frito Gordura Vegetal | Imerso ou preparado com óleo vegetal | Ativo |
| 6 | Frito Gordura Animal | Preparado com banha ou gordura animal | Ativo |
| 7 | Airfryer | Preparado em fritadeira elétrica sem óleo | Ativo |
| 8 | Refogado | Preparado com pequena quantidade de gordura e calor | Ativo |
| 9 | Fermentado | Sofre processo de fermentação natural ou induzida | Ativo |
| 10 | Desidratado | Água retirada para conservação | Ativo |
| 11 | Congelado | Alimento conservado sob baixa temperatura | Ativo |
| 12 | Pré-cozido | Parcialmente cozido antes do preparo final | Ativo |
| 13 | Instantâneo | Preparação rápida com adição de água quente | Ativo |
| 14 | Injetado/Marinado | Recebeu salmoura, tempero ou conservante | Ativo |
| 15 | Liofilizado | Seco a frio em processo industrial | Ativo |
| 16 | Pronto Pra Consumo | Bebidas enlatadas, engarrafadas ou embaladas | Ativo |

## Tabela Auxiliar: tbl\_aux\_grupo\_alimentar

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |
| --- | --- | --- |
| **id** | **nome** | **detalhe** |
| 1 | Cereais e Derivados | Grãos, massas, farinhas, pães |
| 2 | Leguminosas | Feijão, lentilha, ervilha, grão-de-bico |
| 3 | Carnes e Ovos | Bovinos, suínos, aves, peixes, ovos |
| 4 | Leites e Derivados | Leite, queijos, iogurtes |
| 5 | Frutas | Frutas frescas, secas ou congeladas |
| 6 | Verduras e Legumes | Hortaliças e vegetais |
| 7 | Óleos e Gorduras | Manteiga, margarina, azeite, banha |
| 8 | Açúcares e Doces | Açúcar, mel, sobremesas, doces |
| 9 | Industrializados Diversos | Alimentos ultraprocessados, snacks, congelados |
| 10 | Bebidas | Sucos, refrigerantes, Água, Chás, Café |
| 11 | Bebidas Alcoólicas | Bebidas alcoólicas |

## Tabela Auxiliar: tbl\_aux\_categoria\_nutri

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **nome** | **detalhe** | **status** |
| 1 | Fonte de Carboidratos | Alimento com predominância de carboidratos | Ativo |
| 2 | Fonte de Proteínas | Rico em proteínas | Ativo |
| 3 | Alto em Fibras | Rico em fibras alimentares | Ativo |
| 4 | Baixo Teor de Sódio | Possui pouco sódio natural | Ativo |
| 5 | Rico em Gorduras Boas | Fonte de gorduras insaturadas | Ativo |
| 6 | Baixo em Calorias | Pouco valor energético | Ativo |
| 7 | Alto Valor Energético | Energético, ideal para esportistas | Ativo |

## Tabela Auxiliar: tbl\_aux\_origem\_alimentar

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **nome** | **detalhe** | **status** |
| 1 | Vegetal | Derivado de plantas, frutas, vegetais | Ativo |
| 2 | Animal | Derivado de animais (carne, ovos, leite) | Ativo |
| 3 | Mineral | Extraído da natureza (ex: sal, água) | Ativo |
| 4 | Misto | Combinação de origens (ex: pizza, lasanha) | Ativo |
| 5 | Processado | Alimento modificado por processo industrial | Ativo |

## Tabela Auxiliar: tbl\_aux\_processamento

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **nome** | **detalhe** | **status** |
| 1 | In Natura | Natural, sem modificação após colheita/abate | Ativo |
| 2 | Minimamente Processado | Lavado, cortado, resfriado | Ativo |
| 3 | Processado | Passou por cocção ou adição de sal, açúcar, gordura | Ativo |
| 4 | Ultraprocessado | Formulações industriais com aditivos | Ativo |

## Tabela Auxiliar: tbl\_aux\_status\_register

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **nome** | **detalhe** | **status** |
| 1 | Ativo | Registro válido e utilizável | Ativo |
| 2 | Inativo | Temporariamente não visível | Ativo |
| 4 | Excluído | Removido logicamente | Ativo |

## Tabela Auxiliar: tbl\_aux\_verification

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **nome** | **detalhe** | **status** |
| 1 | Verificado | Validado por fonte confiável ou especialista | Ativo |
| 2 | Alerta | Ainda não verificado | Ativo |

## Tabela Auxiliar: tbl\_aux\_tipo\_medida

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **nome** | **detalhe** | **status** |
| 1 | g | Medida em gramas | Ativo |
| 2 | ml | Medida em mililitros | Ativo |

## Tabela Auxiliar: tbl\_aux\_alergicos

### Estrutura da Tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Campo** | **Tipo do Campo** | **Valor Default** | **Tipo de Preenchimento** |
| id | PRIMARY KEY | null | Obrigatório |
| nome | VARCHAR(255) | null | Obrigatório |
| detalhe | VARCHAR(255) | null | Opcional |
| status | VARCHAR(20) | null | Obrigatório |
| user\_registro | VARCHAR(100) | null | Obrigatório |
| dt\_registro | DATETIME | null | Obrigatório |
| dt\_atualizacao | DATETIME | null | Obrigatório |

### Registros da Tabela:

|  |  |  |
| --- | --- | --- |
| **id** | **nome** | **detalhe** |
| 1 | Leite | Alérgeno – Proteínas do leite (caseína, soro); pode causar alergia severa |
| 2 | Ovos | Alérgeno – Clara e gema; presente em massas, pães e alimentos processados |
| 3 | Glúten | Intolerância – Presente no trigo, cevada e centeio; causa doença celíaca ou sensibilidade |
| 4 | Amendoim | Alérgeno – Forte potencial anafilático; mesmo traços podem causar reação |
| 5 | Frutos do mar | Alérgeno – Inclui peixes, crustáceos e moluscos; alergias comuns e graves |
| 6 | Soja | Alérgeno – Presente em óleos, molhos, produtos processados |
| 7 | Trigo | Alérgeno – Causa alergia ou pode estar ligado à intolerância ao glúten |
| 8 | Castanhas (em geral) | Alérgeno – Ex: noz, pecã, castanha-do-pará; alto risco de anafilaxia |
| 9 | Lactose | Intolerância – Açúcar do leite; causa desconforto digestivo por deficiência de lactase |
| 10 | Corantes artificiais | Intolerância – Reações não imunológicas; afetam crianças sensíveis |
| 11 | Gergelim | Alérgeno – Presente em pães, biscoitos e condimentos |
| 12 | Nozes-macadâmia | Alérgeno – Presente em doces, bolos, cookies |
| 13 | Caju (castanha de caju) | Alérgeno – Alergia frequente no Brasil, especialmente entre crianças |
| 14 | Avelã | Alérgeno – Presente em chocolates e cremes |
| 15 | Pistache | Alérgeno – Mais comum em misturas de nuts e sobremesas |
| 16 | Milho | Intolerância – Pode causar reações digestivas em sensíveis |
| 17 | Amêndoas | Alérgeno – Muito usada em confeitaria, leite vegetal |
| 18 | Sementes de girassol | Alérgeno – Presente em barras, pães e granolas |
| 19 | Sulfitos | Alérgeno – Conservante comum; pode causar crises respiratórias em asmáticos |
| 20 | Mostarda | Alérgeno – Condimento que causa alergia, mesmo em pequenas quantidades |
| 21 | Aditivos alimentares | Intolerância – Reações a glutamato monossódico (MSG), nitratos, entre outros |
| 22 | Café / cafeína | Intolerância – Estimula o sistema nervoso; causa taquicardia em sensíveis |
| 23 | Frutose | Intolerância – Açúcar natural das frutas; causa distúrbios gastrointestinais |
| 24 | Açúcar refinado | Intolerância – Sensibilidade à absorção; pode gerar picos glicêmicos |
| 25 | Chocolate | Intolerância – Contém cafeína, açúcar e traços de leite |
| 26 | Tomate | Intolerância – Possui ácido e histaminas; pode causar azia e irritações |
| 27 | Canela | Intolerância – Em sensíveis pode causar urticária ou ardência bucal |
| 28 | Banana | Intolerância – Pode causar fermentação intestinal ou reação cruzada alérgica |
| 29 | Morango | Intolerância – Rico em histaminas; causa coceira ou formigamento |
| 30 | Alho | Intolerância – Pode causar inchaço, gases, refluxo |
| 31 | Cebola | Intolerância – Semelhante ao alho, por compostos sulfurados |
| 32 | Pimenta | Intolerância – Pode causar irritação gástrica e refluxo |
| 33 | Bebidas alcoólicas | Intolerância – Reações a álcool ou seus componentes (como histaminas, sulfitos) |
| 34 | Glutamato monossódico | Intolerância – Realçador de sabor; ligado à “síndrome do restaurante chinês” |
| 35 | Bebidas com gás | Intolerância – Podem causar distensão e gases excessivos |
| 36 | Produtos fermentados | Intolerância – Ex: kefir, kombucha, vinho; podem causar fermentação intestinal |
| 37 | Oleaginosas | Alérgeno – Grupo geral: amendoim, nozes, castanhas, etc. |
| 38 | Lupino | Alérgeno – Farinha de leguminosa; comum na Europa, crescente no Brasil |
| 39 | Proteína da carne vermelha | Intolerância – Pode causar digestão lenta ou reações a alfa-gal (rara) |
| 40 | Histaminas | Intolerância – Presente em queijos, vinho, peixe curado; causa rubor, coceira |