**📋 Plano por Secções**

**🔹 Estrutura da aplicação (HTML)**

* Dois ecrãs: screen1 (home) e screen2 (funções)
* Navegação entre os ecrãs com botões
* Quatro secções no screen2 com caixas de texto (inputs), labels (outputs) e botões

**🔹 Funcionalidade (JavaScript)**

* Cálculo do lucro ou valor a investir conforme as fórmulas:
  + Lucro = (TP2 - TP1) × Investimento / TP1
  + Investimento = Lucro × TP1 / (TP2 - TP1)
  + Usar % de crescimento para calcular TP2: TP2 = TP1 × (1 + % / 100)
* Botões: calcular, limpar campos, navegar entre ecrãs

**🔹 Design (CSS)**

* Estilo simples e responsivo
* Ocultar/mostrar ecrãs com classes CSS

**🔹 PWA + Firebase + GitHub**

* Manifest + Service Worker para transformar em PWA
* Firebase Hosting
* Publicação no GitHub e ligação ao Firebase

**Para passar para o GitHub**

**Depois do repositório, do 1º Push etc.. Voltar ao projecto e fazer o push de novas alterações**

**1 –** git init

**2 -** git remote add origin <https://github.com/antoniorappleton/FinancePWA.git>

3 - git add .

git commit -m "Primeiro commit da PWA"

4 - git branch -M main

git push -u origin main

git add .

git commit -m "Descrição do que alteraste"

git push

Códigos

HTML

<!DOCTYPE html>

<html lang="pt">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <link rel="icon" href="assets/icons/icon-192.png" />

    <meta name="theme-color" content="#007BFF" />

    <title>Calculadora Financeira</title>

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

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

    <link rel="manifest" href="manifest.json" />

    <!-- Enviar email via EmailJS-->

    <script>

      (function () {

        emailjs.init("1Ro3I9ZseyDAc6RTM"); // substitui pelo teu public key

      })();

    </script>

    <!--Firebase-->

    <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-app.js"></script>

    <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-firestore.js"></script>

  </head>

  <body>

    <!-- SCREEN 1 -->

    <div id="screen1" class="screen">

      <h1>Bem-vindo à Calculadora de Lucros</h1>

      <h2>feito por António Appleton</h2>

      <button onclick="goToScreen2()">Começar</button>

    </div>

    <!-- SCREEN 2 -->

    <div id="screen2" class="screen hidden">

      <div

        class="explanation"

        style="

          margin-bottom: 4px;

          padding: 10px;

          border: 1px solid #090909;

          border-radius: 5px;

          background: #fffcfc;

        "

      >

        <h3 style="margin: 10px 0 2px 0">TP1 (Target Price 1):</h3>

        <h6 style="margin: 0 0 8px 0; font-weight: normal">

          Preço atual ou inicial da ação. É o ponto de partida para os cálculos

          de valorização e investimento.

        </h6>

        <h3 style="margin: 10px 0 2px 0">TP2 (Target Price 2):</h3>

        <h6 style="margin: 0 0 8px 0; font-weight: normal">

          Preço futuro da ação. É o ponto de chegada para os cálculos de

          valorização e investimento.

        </h6>

        <h3 style="margin: 10px 0 2px 0">Dividendos:</h3>

        <h6 style="margin: 0 0 8px 0; font-weight: normal">

          Valor que a Empresa paga por cada Stock que é detida, que será somado

          ao lucro total.

        </h6>

      </div>

      <h2>Escolhe uma opção:</h2>

      <div class="buttons-grid">

        <button onclick="abrirSecao(1)">📈 Lucro Esperado</button>

        <button onclick="abrirSecao(2)">📊 Quanto tens de Investir</button>

        <button onclick="abrirSecao(3)">📈 Lucro % de crescimento</button>

        <button onclick="abrirSecao(4)">

          📊 Tens de Investir a % de Crescimento

        </button>

        <button onclick="abrirSecao(5)">📊 Simulador</button>

        <button onclick="abrirSecao(6)">📅 Base Dados das Tuas Stocks</button>

      </div>

      <div class="buttons">

        <button onclick="goToScreen1()">Voltar</button>

      </div>

    </div>

    <!-- SECÇÃO 1 -->

    <div id="sec1Screen" class="screen hidden">

      <div class="section">

        <div class="section-header">

          <h4>1. Calcular Lucro com TP1, TP2, Investimento e Dividendos.</h4>

          <button class="info-toggle" onclick="toggleInfo('info1')">

            ℹ️ +info

          </button>

        </div>

        <div class="info-text" id="info1">

          <h5>

            Introduz o preço inicial da ação (TP1), o preço esperado no futuro

            (TP2), o valor que vais investir e os dividendos esperados. O

            resultado será o lucro total esperado.

          </h5>

        </div>

        <label

          >Preço Inicial (TP1): <input type="number" id="tp1\_1" step="0.01"

        /></label>

        <label

          >Preço Final (TP2): <input type="number" id="tp2\_1" step="0.01"

        /></label>

        <label

          >Valor Investido: <input type="number" id="invest\_1" step="0.01"

        /></label>

        <label

          >Dividendo por ação:

          <input type="number" id="dividendo\_1" step="0.01"

        /></label>

        <p>Lucro valorização: <span id="lucro\_valorizacao1">-</span></p>

        <p>

          Lucro total (incluindo dividendos): <span id="lucro\_total1">-</span>

        </p>

        <button onclick="calcularLucro1()">Calcular</button>

      </div>

      <div class="buttons">

        <button onclick="limparCampos()">Apagar Tudo</button>

        <button onclick="voltarMenu()">Voltar ao Menu</button>

      </div>

    </div>

    <!-- SECÇÃO 2 -->

    <div id="sec2Screen" class="screen hidden">

      <div class="section">

        <div class="section-header">

          <h4>2. Calcular Valor a Investir com TP1, TP2, Lucro Desejado.</h4>

          <button class="info-toggle" onclick="toggleInfo('info2')">

            ℹ️ +info

          </button>

        </div>

        <div class="info-text" id="info2">

          <h5>

            Introduz o preço inicial (TP1), o preço futuro esperado (TP2) e o

            lucro desejado. O resultado será o valor que precisas investir para

            atingir esse lucro.

          </h5>

        </div>

        <label

          >Preço Inicial (TP1): <input type="number" id="tp1\_2" step="0.01"

        /></label>

        <!--teste <label>Preço Final (TP2): <input type="number" id="tp2\_2" step="0.01" /></label> -->

        <label

          >Preço Final (TP2):

          <input

            type="number"

            id="tp2\_2"

            step="0.01"

            oninput="atualizarTP2\_2()"

          />

          % Crescimento: <span id="percent\_2">-</span></label

        >

        <label

          >Lucro Desejado: <input type="number" id="lucro\_2" step="0.01"

        /></label>

        <p>Investimento necessário: <span id="invest2">-</span></p>

        <button onclick="calcularInvestimento2()">Calcular</button>

      </div>

      <div class="buttons">

        <button onclick="limparCampos()">Apagar Tudo</button>

        <button onclick="voltarMenu()">Voltar ao Menu</button>

      </div>

    </div>

    <!-- SECÇÃO 3 -->

    <div id="sec3Screen" class="screen hidden">

      <div class="section">

        <div class="section-header">

          <h4>

            3. Calcular Lucro com TP1, % Crescimento, Investimento e Dividendos.

          </h4>

          <button class="info-toggle" onclick="toggleInfo('info3')">

            ℹ️ +info

          </button>

        </div>

        <div class="info-text" id="info3">

          <h5>

            Introduz o preço inicial (TP1), a percentagem de crescimento

            esperada, o valor que vais investir e os dividendos esperados. O

            resultado será o lucro total esperado.

          </h5>

        </div>

        <label

          >Preço Inicial (TP1): <input type="number" id="tp1\_3" step="0.01"

        /></label>

        <label

          >% Crescimento: <input type="number" id="percent\_3" step="0.01"

        /></label>

        <label

          >Valor Investido: <input type="number" id="invest\_3" step="0.01"

        /></label>

        <label

          >Dividendo por ação:

          <input type="number" id="dividendo\_3" step="0.01"

        /></label>

        <p>Lucro valorização: <span id="lucro\_valorizacao3">-</span></p>

        <p>

          Lucro total (incluindo dividendos): <span id="lucro\_total3">-</span>

        </p>

        <button onclick="calcularLucro3()">Calcular</button>

      </div>

      <div class="buttons">

        <button onclick="limparCampos()">Apagar Tudo</button>

        <button onclick="voltarMenu()">Voltar ao Menu</button>

      </div>

    </div>

    <!-- SECÇÃO 4 -->

    <div id="sec4Screen" class="screen hidden">

      <div class="section">

        <div class="section-header">

          <h4>

            4. Calcular Valor a Investir com TP1, % Crescimento, Lucro Desejado.

          </h4>

          <button class="info-toggle" onclick="toggleInfo('info4')">

            ℹ️ +info

          </button>

        </div>

        <div class="info-text" id="info4">

          <h5>

            Introduz o preço inicial (TP1), a percentagem de crescimento

            esperada e o lucro desejado. O resultado será o valor que precisas

            investir.

          </h5>

        </div>

        <label

          >Preço Inicial (TP1):

          <input

            type="number"

            id="tp1\_4"

            step="0.01"

            oninput="atualizarTP2\_4()"

        /></label>

        <label

          >% Crescimento:

          <input

            type="number"

            id="percent\_4"

            step="0.01"

            oninput="atualizarTP2\_4()"

          />

          TP2: <span id="tp2\_4">-</span></label

        >

        <label

          >Lucro Desejado: <input type="number" id="lucro\_4" step="0.01"

        /></label>

        <p>Investimento necessário: <span id="invest4">-</span></p>

        <button onclick="calcularInvestimento4()">Calcular</button>

      </div>

      <div class="buttons">

        <button onclick="limparCampos()">Apagar Tudo</button>

        <button onclick="voltarMenu()">Voltar ao Menu</button>

      </div>

    </div>

    <!-- SECÇÃO 5 - Simulação -->

    <div id="sec5Screen" class="screen hidden">

      <div class="section">

        <div class="section-header">

          <h4>5. Screen Simulação Lucros.</h4>

        </div>

        <div class="info-text" id="info5">

          <h5>

            Simulação de Lucros entre duas ou mais potenciais Acões, com

            gráficos ilustrativos

          </h5>

        </div>

        <div class="formulario">

          <input type="text" id="nomeAcao" placeholder="Nome da Ação" />

          <input type="number" id="tp1" placeholder="TP1" />

          <input type="number" id="tp2" placeholder="TP2" />

          <input

            type="number"

            id="investimento"

            placeholder="Valor Investido (€)"

          />

          <input

            type="number"

            id="dividendo"

            placeholder="Dividendo por ação (€)"

          />

          <div

            id="botoesSimulacaoRapida"

            class="hidden"

            style="margin-top: 10px; text-align: center"

          >

            <button onclick="simularValorRapido(500)">💶 500</button>

            <button onclick="simularValorRapido(1000)">💶 1000</button>

            <button onclick="simularValorRapido(1500)">💶 1500</button>

            <button onclick="simularValorRapido(2000)">💶 2000</button>

            <button onclick="simularValorRapido(2500)">💶 2500</button>

            <button onclick="simularValorRapido(3000)">💶 3000</button>

          </div>

          <button onclick="simularEGUardar()">Simular e Guardar</button>

        </div>

        <h2>Tabela de Simulações</h2>

        <div class="tabela-wrapper">

          <table id="tabelaSimulacoes">

            <thead>

              <tr>

                <th>Ação</th>

                <th>TP1</th>

                <th>TP2</th>

                <th>Investido</th>

                <th>Lucro</th>

                <th>% Crescimento</th>

              </tr>

            </thead>

            <tbody></tbody>

          </table>

        </div>

        <!-- Botão Limpar-->

        <div class="grafico-wrapper">

          <canvas id="graficoLucro" width="400" height="200"></canvas>

          <button onclick="limparGrafico()">Limpar Gráfico🗑️</button>

        </div>

        <!-- Fim do botão Limpar-->

      </div>

      <div class="buttons">

        <button onclick="limparCampos()">Apagar Tudo</button>

        <button onclick="voltarMenu()">Voltar ao Menu</button>

        <!-- Botão para abrir o popup -->

        <button onclick="abrirPopupEmail()">Enviar Simulação por Email</button>

        <!-- Popup -->

        <div id="popupEmail" class="hidden">

          <h3>Enviar Simulação</h3>

          <div id="dadosSimulacao"></div>

          <div class="email-form">

            <input

              type="email"

              id="emailDestino"

              placeholder="Email de destino"

              required

            />

          </div>

          <div class="botoes-popup">

            <button onclick="enviarEmail()">Enviar</button>

            <button onclick="fecharPopupEmail()">Cancelar</button>

          </div>

        </div>

      </div>

    </div>

<!-- SECÇÃO 6: Registo e Filtro de Ações -->

<div id="sec6Screen" class="screen hidden">

  <!-- REGISTO -->

  <div class="section">

    <h3>📋 6. Registar Ação com Dividendos</h3>

    <div class="form-grid">

      <input type="text" id="nomeAcaoReg" placeholder="Nome da Ação" />

      <input type="text" id="tickerAcaoReg" placeholder="Ticker" />

      <select id="Setor" placeholder="Setor">

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

        <option value="Alimentacao">Alimentação</option>

        <option value="Automovel">Automóvel</option>

        <option value="BensConsumidor">Bens de Consumo</option>

        <option value="ConsumoCiclico">Consumo Cíclico</option>

        <option value="ConsumoDefensivo">Consumo Defensivo</option>

        <option value="Defesa">Defesa</option>

        <option value="Energia">Energia</option>

        <option value="Financas">Finanças</option>

        <option value="Imobiliario">Imobiliário</option>

        <option value="Industria">Indústria</option>

        <option value="InfraestruturasEnergia">Infraestruturas / Energia</option>

        <option value="Materiais">Materiais</option>

        <option value="Restauracao">Restauração</option>

        <option value="Saude">Saúde</option>

        <option value="Tecnologia">Tecnologia</option>

        <option value="Telecomunicacoes">Telecomunicações</option>

      </select>

      <select id="Mercado" placeholder="Mercado">

        <option value="">Seleciona o Mercado</option>

        <option value="Portugal">Portugal</option>

        <option value="Europeu">Europeu</option>

        <option value="Americano">Americano</option>

        <option value="Americano SP500">Americano SP500</option>

      </select>

      <input type="number" id="valorDividendoReg" step="0.01" placeholder="Valor do Dividendo (€)" />

      <select id="mesDividendoReg">

        <option value="">Seleciona o mês</option>

        <option value="Janeiro">Janeiro</option>

        <option value="Fevereiro">Fevereiro</option>

        <option value="Março">Março</option>

        <option value="Abril">Abril</option>

        <option value="Maio">Maio</option>

        <option value="Junho">Junho</option>

        <option value="Julho">Julho</option>

        <option value="Agosto">Agosto</option>

        <option value="Setembro">Setembro</option>

        <option value="Outubro">Outubro</option>

        <option value="Novembro">Novembro</option>

        <option value="Dezembro">Dezembro</option>

      </select>

      <select id="periodicidade" placeholder="Periodicidade">

        <option value="">Seleciona a Frequência</option>

        <option value="Anual">Anual</option>

        <option value="Semestral">Semestral</option>

        <option value="Trimestral">Trimestral</option>

        <option value="Mensal">Mensal</option>

      </select>

    </div>

    <div class="buttons">

      <button onclick="guardarOuAtualizarAcaoFirebase()">💾 Guardar / Atualizar</button>

    </div>

  </div>

  <!-- BOTÃO PARA ABRIR POPUP -->

  <div class="section">

    <div class="section-header">

      <h4>🔍 Filtrar Ações</h4>

      <button onclick="abrirPopupFiltro()">🔍 Abrir Filtros</button>

    </div>

  </div>

  <div class="buttons">

    <button onclick="voltarMenu()">Voltar ao Menu</button>

  </div>

</div>

<!-- POPUP FILTRO AÇÕES -->

<div id="popupFiltro" class="popup hidden">

  <div class="popup-content">

    <h4>🔍 Filtros</h4>

    <select id="filtroSetor">

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

      <option value="Alimentacao">Alimentação</option>

      <option value="Automovel">Automóvel</option>

      <option value="BensConsumidor">Bens de Consumo</option>

      <option value="ConsumoCiclico">Consumo Cíclico</option>

      <option value="ConsumoDefensivo">Consumo Defensivo</option>

      <option value="Defesa">Defesa</option>

      <option value="Energia">Energia</option>

      <option value="Financas">Finanças</option>

      <option value="Imobiliario">Imobiliário</option>

      <option value="Industria">Indústria</option>

      <option value="InfraestruturasEnergia">Infraestruturas / Energia</option>

      <option value="Materiais">Materiais</option>

      <option value="Restauracao">Restauração</option>

      <option value="Saude">Saúde</option>

      <option value="Tecnologia">Tecnologia</option>

      <option value="Telecomunicacoes">Telecomunicações</option>

    </select>

    <select id="filtroPeriodicidade">

      <option value="">Selecione a Periodicidade</option>

      <option value="Mensal">Mensal</option>

      <option value="Trimestral">Trimestral</option>

      <option value="Semestral">Semestral</option>

      <option value="Anual">Anual</option>

    </select>

    <select id="filtroMercado">

      <option value="">Seleciona o Mercado</option>

      <option value="Portugal">Portugal</option>

      <option value="Europeu">Europeu</option>

      <option value="Americano">Americano</option>

      <option value="Americano SP500">Americano SP500</option>

    </select>

    <select id="filtroMes">

      <option value="">Selecione um Mês</option>

      <option value="Janeiro">Janeiro</option>

      <option value="Fevereiro">Fevereiro</option>

      <option value="Março">Março</option>

      <option value="Abril">Abril</option>

      <option value="Maio">Maio</option>

      <option value="Junho">Junho</option>

      <option value="Julho">Julho</option>

      <option value="Agosto">Agosto</option>

      <option value="Setembro">Setembro</option>

      <option value="Outubro">Outubro</option>

      <option value="Novembro">Novembro</option>

      <option value="Dezembro">Dezembro</option>

    </select>

    <button onclick="filtrarAcoes()">🔎 Aplicar Filtros</button>

    <button onclick="fecharPopupFiltro()">❌ Fechar</button>

    <div id="resultadoFiltroMes" class="resultado-popup"></div>

  </div>

</div>

    <script src="script.js"></script>

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

  </body>

</html>

CSS

.buttons-grid {

  display: flex;

  gap: 10px;

  flex-wrap: wrap;

  justify-content: space-between;

  margin-bottom: 20px;

  text-align: center;

  width: 100%;

  /\* <- antes estava 250px, estava a limitar os botões \*/

}

.buttons-grid button {

  flex: 1 1 100%;

  max-width: 100%;

  padding: 15px;

  font-size: 1rem;

  border-radius: 60px;

  background: #626466;

  border: none;

  color: white;

  cursor: pointer;

  text-align: center;

  white-space: normal;

  /\* permite quebra de linha se o texto for longo \*/

}

.buttons-grid button:hover {

  background: #323234;

}

body {

  font-family: Arial, sans-serif;

  background: #ffffff;

  color: #fff;

  margin: 0;

  padding: 20px;

  display: flex;

  justify-content: center;

  align-items: flex-start;

  min-height: 100vh;

}

.screen {

  background: #fcfcfc;

  border-radius: 10px;

  box-sizing: border-box;

  box-shadow: 0 0 12px rgba(0, 0, 0, 0.3);

  width: 100%;

  max-width: 480px;/\* ou 600px para tablets \*/

  margin: 0 auto;

  padding: 20px;

}

.hidden {

  display: none;

}

h1,

h2 {

  color: #141515;

  margin-bottom: 4px;

}

h3,

h5,

h6,

h4 {

  color: #0e0f0f;

  margin-bottom: 0px;

}

label {

  display: block;

  margin-bottom: 8px;

  color: #131313;

}

input[type="number"] {

  width: 100%;

  padding: 6px 8px;

  border-radius: 6px;

  border: none;

  margin-top: 3px;

  margin-bottom: 12px;

  box-sizing: border-box;

}

button {

  background-color: #1c1c1d;

  border: none;

  color: white;

  padding: 10px 20px;

  margin: 8px 6px 0 0;

  border-radius: 60px;

  cursor: pointer;

  font-size: 16px;

  transition: background-color 0.25s ease;

  width: 250px;

}

button:hover {

  background-color: #383a3b;

}

p {

  margin: 6px 0 10px;

  font-weight: bold;

  color: #313334;

}

.buttons {

  margin-top: 15px;

  width: 100%;

  display: flex;

  justify-content: center;/\* ou 'flex-end' se quiseres à direita \*/

}

.buttons button {

  width: 250px;

  padding: 20px;

  font-size: 1rem;

  border-radius: 10px;

  background: #212122;

  border: none;

  color: white;

  cursor: pointer;

  text-align: center;

}

.buttons button:hover {

  background: #505152;

}

label input {

  margin-right: 10px;

  color: #121212;

}

#tp2\_4 {

  font-weight: bold;

  margin-left: 5px;

}

.info-text {

  display: none;

  background-color: #f3f4f5;

  padding: 0.1rem;

  margin: 0.5rem 0;

  border-left: 4px solid rgb(34, 34, 34);

  border-radius: 4px;

  font-size: 0.9rem;

  color: #343636;

}

.section-header {

  display: flex;

  justify-content: space-between;

  align-items: center;

}

.section-header h3 {

  margin: 0;

  font-size: 1.1rem;

  font-weight: bold;

  color: rgb(25, 24, 24);

}

.info-toggle {

  background: none;

  border: none;

  color: #3e4447;

  font-size: 0.9rem;

  cursor: pointer;

  display: flex;

  align-items: center;

  gap: 4px;

}

.section {

  background-color: #ffffff;

  padding: 20px;

  border-radius: 10px;

  margin-bottom: 30px;

  /\* 👈 Espaçamento entre as secções \*/

}

/\* Container da Tabela \*/

.tabela-wrapper {

  overflow-x: auto;

  width: 100%;

  margin-top: 10px;

  border-radius: 8px;

}

/\* Tabela Estilizada \*/

#tabelaSimulacoes {

  width: 100%;

  border-collapse: collapse;

  background-color: #ecf0f1;

  font-size: 0.95em;

  border-radius: 8px;

  overflow: hidden;

  min-width: 600px;

  /\* Garante boa legibilidade \*/

}

#tabelaSimulacoes thead {

  background-color: #dadee1;

  color: #121212;

  text-align: left;

}

#tabelaSimulacoes th,

#tabelaSimulacoes td {

  padding: 12px 16px;

  border-bottom: 1px solid #bdc3c7;

  white-space: nowrap;

}

#tabelaSimulacoes tbody tr {

  background-color: #ffffff;

  color: #2c3e50;

}

#tabelaSimulacoes tbody tr:hover {

  background-color: #d6eaf8;

}

/\*Botão Limpar Gráfico\*/

.grafico-wrapper {

  position: relative;

  width: 100%;

  max-width: 400px;

  /\* Ajusta conforme o teu layout \*/

  margin: 0 auto;

}

#limparGraficoBtn {

  position: absolute;

  bottom: 10px;

  right: 10px;

  background: none;

  border: none;

  font-size: 20px;

  cursor: pointer;

  color: #ffffff;

  /\* Branco para sobressair \*/

}

#limparGraficoBtn:hover {

  color: #ff5555;

  /\* Fica vermelho ao passar o rato \*/

}

/\* Estilo geral do popup \*/

#popupEmail {

  position: fixed;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  background-color: #ffffff;

  border: 2px solid #424243;

  border-radius: 10px;

  padding: 20px;

  width: 90%;

  max-width: 400px;

  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);

  z-index: 1000;

}

/\*ENVIAR EMAIL\*/

/\* Estilo geral do popup \*/

#popupEmail {

  position: fixed;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  background-color: #ffffff;

  border: 2px solid #242424;

  border-radius: 10px;

  padding: 20px;

  width: 90%;

  max-width: 400px;

  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);

  z-index: 1000;

}

/\* Título do popup \*/

#popupEmail h3 {

  margin-top: 0;

  font-size: 1.4em;

  color: #151515;

  text-align: center;

}

/\* Campo de email \*/

#popupEmail input[type="email"] {

  width: 100%;

  padding: 10px;

  margin-top: 10px;

  margin-bottom: 15px;

  border: 1px solid #3b3939;

  border-radius: 5px;

  font-size: 1em;

}

/\* Botões \*/

#popupEmail button {

  padding: 10px 15px;

  margin: 5px;

  border: none;

  border-radius: 5px;

  font-size: 1em;

  cursor: pointer;

}

#popupEmail button:first-of-type {

  background-color: #2d2e30;

  color: white;

}

#popupEmail button:last-of-type {

  background-color: #ccc;

  color: #333;

}

/\* Conteúdo das simulações \*/

#dadosSimulacao {

  background-color: #f0f5f6;

  color: #383636;

  padding: 10px;

  border-radius: 5px;

  max-height: 200px;

  overflow-y: auto;

  font-size: 0.95em;

  line-height: 1.5em;

  border: 1px solid #ddd;

  margin-bottom: 10px;

}

#dadosSimulacao p {

  margin: 6px 0;

}

#dadosSimulacao p strong {

  color: #007BFF;

}

#dadosSimulacao hr {

  border: none;

  border-top: 1px solid #ccc;

  margin: 10px 0;

}

/\* Ocultar popup \*/

.hidden {

  display: none;

}

/\* Caixa de entrada de email \*/

#popupEmail .email-form {

  margin-bottom: 15px;

  text-align: center;

}

#popupEmail input[type="email"] {

  width: 95%;

  max-width: 100%;

  padding: 10px;

  border: 1px solid #ccc;

  border-radius: 5px;

  font-size: 1em;

}

/\* Alinhamento central dos botões \*/

#popupEmail .botoes-popup {

  display: flex;

  justify-content: center;

  gap: 10px;

}

/\* Secção 6 - Registo de Ações \*/

#sec6Screen .section {

  display: flex;

  flex-direction: column;

  gap: 12px;

}

#sec6Screen label {

  display: flex;

  flex-direction: column;

  font-weight: bold;

  color: #ecf0f1;

}

#sec6Screen input,

#sec6Screen select,

#sec6Screen textarea {

  padding: 8px;

  border-radius: 6px;

  border: none;

  font-size: 1em;

  margin-top: 4px;

  box-sizing: border-box;

}

#sec6Screen textarea {

  resize: vertical;

}

/\* Botão para mostrar filtros \*/

#btnMostrarFiltros {

  background-color: #2c2e2e;

  color: white;

  padding: 10px;

  border: none;

  border-radius: 6px;

  cursor: pointer;

  margin-top: 10px;

  transition: background-color 0.3s ease;

}

#btnMostrarFiltros:hover {

  background-color: #3d3f3f;

}

/\* Container dos botões de mês \*/

#filtrosMesContainer {

  display: flex;

  flex-wrap: wrap;

  gap: 8px;

  margin-top: 15px;

  justify-content: center;

  animation: fadeIn 0.4s ease-in-out;

}

#filtrosMesContainer.hidden {

  display: none;

}

#filtrosMesContainer button {

  flex: 1 1 30%;

  padding: 10px;

  background-color: #2c3e50;

  color: white;

  border: none;

  border-radius: 6px;

  cursor: pointer;

  font-size: 0.9em;

  transition: background-color 0.3s ease;

}

#filtrosMesContainer button:hover {

  background-color: #1a252f;

}

/\* Resultado do filtro \*/

#resultadoFiltroMes {

  margin-top: 20px;

  background-color: #1c2833;

  color: #ecf0f1;

  padding: 12px;

  border-radius: 8px;

  max-height: 250px;

  overflow-y: auto;

  font-size: 0.95em;

  line-height: 1.5em;

}

/\* Animação suave \*/

/\* Secção 6 - Registo e Filtro \*/

#sec6Screen .form-grid {

  display: grid;

  grid-template-columns: 1fr;

  gap: 12px;

}

#sec6Screen .filtros-container {

  display: flex;

  flex-direction: column;

  gap: 10px;

  margin-bottom: 10px;

}

#sec6Screen label {

  display: flex;

  flex-direction: column;

  font-weight: bold;

  color: #ecf0f1;

}

#sec6Screen input,

#sec6Screen select {

  padding: 8px;

  border-radius: 6px;

  border: none;

  font-size: 1em;

  margin-top: 4px;

  box-sizing: border-box;

}

#resultadoFiltroMes.resultado-acoes {

  margin-top: 15px;

  background-color: #f2f6f9;

  color: #141414;

  padding: 12px;

  border-radius: 8px;

  max-height: 300px;

  overflow-y: auto;

  font-size: 0.95em;

  line-height: 1.5em;

}

#resultadoFiltroMes ul {

  padding-left: 20px;

}

#resultadoFiltroMes li {

  margin-bottom: 10px;

  border-bottom: 1px solid #444;

  padding-bottom: 6px;

}

@keyframes fadeIn {

  from {

    opacity: 0;

    transform: translateY(-10px);

  }

  to {

    opacity: 1;

    transform: translateY(0);

  }

}

.form-grid {

  display: grid;

  grid-template-columns: 1fr 1fr;

  gap: 10px;

}

.section-header {

  display: flex;

  justify-content: space-between;

  align-items: center;

}

\* {

  box-sizing: border-box;

}

/\*adicionado periência\*/

/\*Tornar tudo responsivo e centrado na largura total\*/

body, html {

  margin: 0;

  padding: 0;

  font-family: 'Segoe UI', sans-serif;

  background-color: #f7f9fa;

  color: #212121;

  width: 100%;

  min-height: 100vh;

  box-sizing: border-box;

}

/\* Garante que todas as "screens" ocupam a largura total \*/

.screen {

  padding: 16px;

  width: 100%;

  max-width: 768px;

  margin: 0 auto;

}

/\*Melhorar o espaçamento e separadores entre secções e títulos\*/

.section {

  margin-bottom: 32px;

  padding: 16px;

  background-color: #ffffff;

  border-radius: 12px;

  box-shadow: 0 2px 6px rgba(0,0,0,0.05);

}

/\* Títulos \*/

.section h3, .section h4 {

  margin-bottom: 16px;

  font-size: 1.25rem;

  color: #2c3e50;

  border-bottom: 1px solid #dcdcdc;

  padding-bottom: 6px;

}

/\*Melhorar campos de input e etiquetas\*/

.section label {

  font-weight: 600;

  font-size: 0.95rem;

  color: #333;

  margin-bottom: 6px;

}

.section input[type="text"],

.section input[type="number"],

.section select {

  width: 100%;

  padding: 10px;

  margin-bottom: 12px;

  border: 1px solid #ccc;

  border-radius: 6px;

  font-size: 1rem;

}

/\*Organizar botões e formulários responsivamente\*/

button {

  padding: 10px 16px;

  font-size: 0.95rem;

  border: none;

  border-radius: 6px;

  margin: 6px 4px 0 0;

  cursor: pointer;

  transition: background-color 0.3s;

}

button:hover {

  opacity: 0.85;

}

.buttons {

  display: flex;

  flex-wrap: wrap;

  gap: 10px;

}

.buttons button {

  flex: 1 1 45%;

}

.form-grid {

  display: grid;

  grid-template-columns: 1fr;

  gap: 12px;

}

/\*Melhorar visual de listagens e resultados\*/

.resultado-acoes {

  background-color: #f0f4f8;

  padding: 12px;

  border-radius: 8px;

  margin-top: 12px;

  font-size: 0.9rem;

  color: #1c2833;

}

.resultado-acoes ul {

  padding-left: 20px;

}

.resultado-acoes li {

  margin-bottom: 12px;

  padding: 10px;

  background: #ffffff;

  border: 1px solid #ddd;

  border-radius: 6px;

  box-shadow: 0 1px 2px rgba(0,0,0,0.03);

}

/\*Mobile-first ajustes e boas práticas\*/

@media (min-width: 600px) {

  .form-grid {

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

  }

  .buttons {

    justify-content: space-between;

  }

}

/\* 💡 Cor do texto das labels na secção 6 \*/

#sec6Screen label {

  color: #333; /\* Podes alterar aqui a cor do texto das etiquetas \*/

}

/\* 💡 Cor dos botões principais \*/

button {

  background-color: #007BFF; /\* Azul principal, muda aqui para verde ou outro \*/

}

JAVASCRIPT

// Firebase Configurações (substituir pelos teus dados reais)

const firebaseConfig = {

  apiKey: "AIzaSyDbSYjVwsOOnBjZe\_X8y7gS-W4DhYqHEnE",

  authDomain: "appfinance-812b2.firebaseapp.com",

  projectId: "appfinance-812b2",

  storageBucket: "appfinance-812b2.firebasestorage.app",

  messagingSenderId: "383837988480",

  appId: "1:383837988480:web:dd114574838c6a9dbb2865",

};

// Registo do Service Worker (PWA)

if ("serviceWorker" in navigator) {

  navigator.serviceWorker

    .register("sw.js")

    .then((reg) => console.log("SW registado!", reg))

    .catch((err) => console.error("SW falhou:", err));

}

// Inicializar Firebase

firebase.initializeApp(firebaseConfig);

const db = firebase.firestore();

// Navegação entre screens

function goToScreen2() {

  document.getElementById("screen1").classList.add("hidden");

  document.getElementById("screen2").classList.remove("hidden");

  document.getElementById("botoesSimulacaoRapida").classList.add("hidden buttons");

}

function goToScreen1() {

  document.getElementById("screen2").classList.add("hidden");

  document.getElementById("screen1").classList.remove("hidden");

}

// Secção 1 - Lucro e lucro total com dividendos

function calcularLucro1() {

  let tp1 = parseFloat(document.getElementById("tp1\_1").value);

  let tp2 = parseFloat(document.getElementById("tp2\_1").value);

  let invest = parseFloat(document.getElementById("invest\_1").value);

  let dividendo = parseFloat(document.getElementById("dividendo\_1").value);

  if (!isNaN(tp1) && !isNaN(tp2) && !isNaN(invest) && !isNaN(dividendo)) {

    let lucro\_valorizacao = (invest \* (tp2 - tp1)) / tp1;

    let numero\_acoes = invest / tp1;

    let lucro\_dividendos = numero\_acoes \* dividendo;

    let lucro\_total = lucro\_valorizacao + lucro\_dividendos;

    document.getElementById("lucro\_valorizacao1").textContent =

      lucro\_valorizacao.toFixed(2);

    document.getElementById("lucro\_total1").textContent =

      lucro\_total.toFixed(2);

  } else {

    document.getElementById("lucro\_valorizacao1").textContent =

      "Preencha todos os campos.";

    document.getElementById("lucro\_total1").textContent = "-";

  }

}

// Secção 2 - Investimento para lucro desejado (sem dividendos)

function calcularInvestimento2() {

  let tp1 = parseFloat(document.getElementById("tp1\_2").value);

  let tp2 = parseFloat(document.getElementById("tp2\_2").value);

  let lucro = parseFloat(document.getElementById("lucro\_2").value);

  if (!isNaN(tp1) && !isNaN(tp2) && !isNaN(lucro)) {

    let invest = (lucro \* tp1) / (tp2 - tp1);

    if (invest < 0) invest = 0;

    document.getElementById("invest2").textContent = invest.toFixed(2);

  } else {

    document.getElementById("invest2").textContent =

      "Preencha todos os campos.";

  }

}

// Atualiza % crescimento dinamicamente com base em TP1 e TP2 Crescimento

function atualizarTP2\_2() {

  const tp1 = parseFloat(document.getElementById("tp1\_2").value);

  const tp2 = parseFloat(document.getElementById("tp2\_2").value);

  if (!isNaN(tp1) && !isNaN(tp2) && tp1 !== 0) {

    const percent = ((tp2 - tp1) / tp1) \* 100;

    document.getElementById("percent\_2").textContent = percent.toFixed(2) + "%";

  } else {

    document.getElementById("percent\_2").textContent = "-";

  }

}

// Secção 3 - Lucro e lucro total com dividendos (percentagem crescimento)

function calcularLucro3() {

  let tp1 = parseFloat(document.getElementById("tp1\_3").value);

  let percent = parseFloat(document.getElementById("percent\_3").value);

  let invest = parseFloat(document.getElementById("invest\_3").value);

  let dividendo = parseFloat(document.getElementById("dividendo\_3").value);

  if (!isNaN(tp1) && !isNaN(percent) && !isNaN(invest) && !isNaN(dividendo)) {

    let tp2 = tp1 \* (1 + percent / 100);

    let lucro\_valorizacao = (invest \* (tp2 - tp1)) / tp1;

    let numero\_acoes = invest / tp1;

    let lucro\_dividendos = numero\_acoes \* dividendo;

    let lucro\_total = lucro\_valorizacao + lucro\_dividendos;

    document.getElementById("lucro\_valorizacao3").textContent =

      lucro\_valorizacao.toFixed(2);

    document.getElementById("lucro\_total3").textContent =

      lucro\_total.toFixed(2);

  } else {

    document.getElementById("lucro\_valorizacao3").textContent =

      "Preencha todos os campos.";

    document.getElementById("lucro\_total3").textContent = "-";

  }

}

// Secção 4 - Lucro e lucro total (percentagem crescimento)

// Atualiza TP2 dinamicamente com base em TP1 e % Crescimento

function atualizarTP2\_4() {

  const tp1 = parseFloat(document.getElementById("tp1\_4").value);

  const percent = parseFloat(document.getElementById("percent\_4").value);

  if (!isNaN(tp1) && !isNaN(percent)) {

    const tp2 = tp1 \* (1 + percent / 100);

    document.getElementById("tp2\_4").textContent = tp2.toFixed(2);

  } else {

    document.getElementById("tp2\_4").textContent = "-";

  }

}

// Calcula o investimento necessário para obter o lucro desejado

function calcularInvestimento4() {

  const tp1 = parseFloat(document.getElementById("tp1\_4").value);

  const percent = parseFloat(document.getElementById("percent\_4").value);

  const lucro = parseFloat(document.getElementById("lucro\_4").value);

  if (!isNaN(tp1) && !isNaN(percent) && !isNaN(lucro)) {

    const tp2 = tp1 \* (1 + percent / 100);

    const invest = (lucro \* tp1) / (tp2 - tp1);

    document.getElementById("tp2\_4").textContent = tp2.toFixed(2);

    document.getElementById("invest4").textContent =

      invest < 0 ? "0.00" : invest.toFixed(2);

  } else {

    document.getElementById("invest4").textContent =

      "Preencha todos os campos.";

  }

}

//Botão +Info

function toggleInfo(id) {

  const info = document.getElementById(id);

  if (info.style.display === "none" || info.style.display === "") {

    info.style.display = "block";

  } else {

    info.style.display = "none";

  }

}

function mostrarSecao(secaoId) {

  document.getElementById("menuSecoes").classList.add("hidden");

  document.querySelectorAll("#screen2 > div").forEach((div) => {

    if (div.id.startsWith("secao")) {

      div.classList.add("hidden");

    }

  });

  document.getElementById(secaoId).classList.remove("hidden");

}

function voltarMenuSecoes() {

  document.querySelectorAll("#screen2 > div").forEach((div) => {

    if (div.id.startsWith("secao")) {

      div.classList.add("hidden");

    }

  });

  document.getElementById("menuSecoes").classList.remove("hidden");

}

function abrirSecao(num) {

  document.getElementById("screen2").classList.add("hidden");

  for (let i = 1; i <= 4; i++) {

    document.getElementById(`sec${i}Screen`).classList.add("hidden");

  }

  document.getElementById(`sec${num}Screen`).classList.remove("hidden");

}

function voltarMenu() {

  for (let i = 1; i <= 4; i++) {

    document.getElementById(`sec${i}Screen`).classList.add("hidden");

  }

  document.getElementById("screen2").classList.remove("hidden");

}

//Simulações

const simulacoes = [];

//Fórmulas de Lucros

function guardarSimulacao(nomeAcao, tp1, tp2, valorInvestido, dividendo = 0) {

  const crescimento = ((tp2 - tp1) / tp1) \* 100;

  const numeroAcoes = valorInvestido / tp1;

  const lucroValorizacao = (tp2 - tp1) \* numeroAcoes;

  const lucroDividendos = numeroAcoes \* dividendo;

  const lucroTotal = lucroValorizacao + lucroDividendos;

  const novaSimulacao = {

    nomeAcao,

    tp1,

    tp2,

    valorInvestido,

    lucro: parseFloat(lucroTotal.toFixed(2)),

    crescimentoPercentual: parseFloat(crescimento.toFixed(2)),

  };

  simulacoes.push(novaSimulacao);

  atualizarTabela();

  atualizarGrafico();

}

//Atualizar a Tabela

function atualizarTabela() {

  const corpoTabela = document.querySelector("#tabelaSimulacoes tbody");

  corpoTabela.innerHTML = "";

  simulacoes.forEach((sim) => {

    const linha = document.createElement("tr");

    linha.innerHTML = `

      <td>${sim.nomeAcao}</td>

      <td>${sim.tp1}</td>

      <td>${sim.tp2}</td>

      <td>${sim.valorInvestido}</td>

      <td>${sim.lucro}</td>

      <td>${sim.crescimentoPercentual}%</td>

    `;

    corpoTabela.appendChild(linha);

  });

}

let grafico;

function atualizarGrafico() {

  const labels = simulacoes.map((s) => s.nomeAcao);

  const dadosLucro = simulacoes.map((s) => s.lucro);

  if (grafico) grafico.destroy();

  const ctx = document.getElementById("graficoLucro").getContext("2d");

  grafico = new Chart(ctx, {

    type: "bar",

    data: {

      labels: labels,

      datasets: [

        {

          label: "Lucro (€)",

          data: dadosLucro,

          backgroundColor: "rgba(54, 162, 235, 0.5)",

          borderColor: "rgba(54, 162, 235, 1)",

          borderWidth: 1,

        },

      ],

    },

    options: {

      scales: {

        y: { beginAtZero: true },

      },

    },

  });

}

//Botão Simular e Guardar para passar para o gráfico

function simularEGUardar() {

  const nome = document.getElementById("nomeAcao").value;

  const tp1 = parseFloat(document.getElementById("tp1").value);

  const tp2 = parseFloat(document.getElementById("tp2").value);

  const investimento = parseFloat(document.getElementById("investimento").value);

  const dividendo = parseFloat(document.getElementById("dividendo").value || 0);

  if (!nome || isNaN(tp1) || isNaN(tp2) || isNaN(investimento)) {

    alert("Preenche todos os campos!");

    return;

  }

  guardarSimulacao(nome, tp1, tp2, investimento, dividendo);

}

function voltarMenu() {

  // Esconde todas as secções

  const secoes = document.querySelectorAll(".screen");

  secoes.forEach((secao) => secao.classList.add("hidden"));

  // Mostra apenas o menu principal

  document.getElementById("screen2").classList.remove("hidden");

  secoes.forEach((secao) => secao.classList.add("hidden"));

  document.getElementById("screen2").classList.remove("hidden");

  // 🔒 Esconde os botões de simulação rápida se estiverem visíveis

  const botoes = document.getElementById("botoesSimulacaoRapida");

  if (botoes) botoes.classList.add("hidden");

}

function limparCampos() {

  // Encontra a secção visível (sem a classe "hidden")

  const secaoVisivel = document.querySelector(".screen:not(.hidden)");

  if (secaoVisivel) {

    // Limpa todos os inputs do tipo number e text dentro da secção visível

    const inputs = secaoVisivel.querySelectorAll(

      "input[type=number], input[type=text]"

    );

    inputs.forEach((input) => (input.value = ""));

    // Limpa todos os spans com resultados dentro da secção visível

    const spans = secaoVisivel.querySelectorAll("span");

    spans.forEach((span) => {

      if (span.id) {

        span.textContent = "-";

      }

    });

  }

}

//secção 5 - Simulador

function abrirSecao(num) {

  document.getElementById("screen2").classList.add("hidden");

  for (let i = 1; i <= 6; i++) {

    document.getElementById(`sec${i}Screen`).classList.add("hidden");

  }

  document.getElementById(`sec${num}Screen`).classList.remove("hidden");

  // 🔄 Garante que os botões de simulação rápida estão sempre escondidos ao entrar no screen 5

  if (num === 5) {

    const botoes = document.getElementById("botoesSimulacaoRapida");

    if (botoes && !botoes.classList.contains("hidden")) {

      botoes.classList.add("hidden");

    }

  }

}

function prepararSimulacao(nome,valorStock,dividendo) {

  console.log("Preparar simulação para:", nome,"a TP1",valorStock,"com dividendo:", dividendo);

  abrirSecao(5); // Vai para screen simulação

  document.getElementById("nomeAcao").value = nome;

  // Mostra os botões de simulação rápida

  const botoes = document.getElementById("botoesSimulacaoRapida");

  if (botoes) {

    botoes.classList.remove("hidden");

  }

}

// 🚨 NOVO: Limpar Gráfico

function limparGrafico() {

  simulacoes.length = 0; // Apaga todas as simulações

  atualizarTabela(); // Atualiza a tabela (fica vazia)

  if (grafico) {

    grafico.destroy(); // Destrói o gráfico existente

    grafico = null;

  }

}

//Enviar Email

function abrirPopupEmail() {

  const popup = document.getElementById("popupEmail");

  popup.classList.remove("hidden");

  const dadosDiv = document.getElementById("dadosSimulacao");

  if (simulacoes.length > 0) {

    let html = "<h4>Simulações Guardadas:</h4>";

    simulacoes.forEach((sim, index) => {

      html += `

        <p><strong>Simulação ${index + 1}</strong></p>

        <p>Ação: ${sim.nomeAcao}</p>

        <p>Preço Inicial: €${sim.tp1}</p>

        <p>Preço Final: €${sim.tp2}</p>

        <p>Investimento: €${sim.valorInvestido}</p>

        <p>Lucro: €${sim.lucro}</p>

        <p>Crescimento: ${sim.crescimentoPercentual}%</p>

        <hr>

      `;

    });

    dadosDiv.innerHTML = html;

  } else {

    dadosDiv.innerHTML = "<p>Sem simulações disponíveis.</p>";

  }

}

function fecharPopupEmail() {

  document.getElementById("popupEmail").classList.add("hidden");

}

//Enviar Email

function enviarEmail() {

  const emailDestino = document.getElementById("emailDestino").value;

  if (!emailDestino || simulacoes.length === 0) {

    alert("Preenche o email e faz pelo menos uma simulação.");

    return;

  }

  const assunto = encodeURIComponent("Resumo de Simulações Financeiras");

  let corpoTexto = "Resumo das Simulações:\n\n";

  simulacoes.forEach((sim, index) => {

    corpoTexto += `Simulação ${index + 1}:\n`;

    corpoTexto += `Ação: ${sim.nomeAcao}\n`;

    corpoTexto += `Preço Inicial: €${sim.tp1}\n`;

    corpoTexto += `Preço Final: €${sim.tp2}\n`;

    corpoTexto += `Investimento: €${sim.valorInvestido}\n`;

    corpoTexto += `Lucro: €${sim.lucro}\n`;

    corpoTexto += `Crescimento: ${sim.crescimentoPercentual}%\n\n`;

  });

  const corpo = encodeURIComponent(corpoTexto);

  const mailtoLink = `mailto:${emailDestino}?subject=${assunto}&body=${corpo}`;

  window.location.href = mailtoLink;

}

//Menu 6 - Registar/Editar e Filtrar Empresas com Dividendos

//Botão secção dos filtros

function toggleFiltrosMes() {

  const filtrosDiv = document.getElementById("filtrosMesContainer");

  const botao = document.getElementById("btnMostrarFiltros");

  if (filtrosDiv.classList.contains("hidden")) {

    filtrosDiv.classList.remove("hidden");

    botao.textContent = "▲";

  } else {

    filtrosDiv.classList.add("hidden");

    botao.textContent = "▼";

  }

}

// Filtrar Base Dados Firebase por múltiplos critérios combinados

function filtrarAcoes() {

  const setor = document.getElementById("filtroSetor").value;

  const mercado = document.getElementById("filtroMercado").value;

  const mes = document.getElementById("filtroMes").value;

  const periodicidade = document.getElementById("filtroPeriodicidade").value;

  const resultadoDiv = document.getElementById("resultadoFiltroMes");

  resultadoDiv.innerHTML = "A carregar...";

  db.collection("acoesDividendos")

    .get()

    .then((querySnapshot) => {

      let html = "<ul>";

      let count = 0;

      querySnapshot.forEach((doc) => {

        const dados = doc.data();

        const matchSetor =

          !setor || dados.setor?.trim().toLowerCase() === setor.trim().toLowerCase();

        const matchMercado =

          !mercado || dados.mercado?.trim().toLowerCase() === mercado.trim().toLowerCase();

        const matchMes =

          !mes || dados.mes?.trim().toLowerCase() === mes.trim().toLowerCase();

        const matchPeriodicidade =

          !periodicidade || dados.periodicidade?.trim().toLowerCase() === periodicidade.trim().toLowerCase();

        if (matchSetor && matchMercado && matchMes && matchPeriodicidade) {

          html += `<li>

            <strong>${dados.nome}</strong> (${dados.ticker})<br>

            Setor: ${dados.setor} | Mercado: ${dados.mercado} | valorStock: €${

            dados.valorStock

          } |Dividendo: €${dados.dividendo} |

            Mês: ${dados.mes} | Periodicidade: ${dados.periodicidade}<br>

            <div style="display: flex; justify-content: center; gap: 8px; flex-wrap: wrap; margin-top: 5px;">

              <button onclick="editarAcao('${doc.id}', ${JSON.stringify(

            dados

          ).replace(/"/g, "&quot;")})">✏️ Editar</button>

              <button onclick="prepararSimulacao('${dados.nome}','${

            dados.valorStock

          }', ${

            dados.dividendo

          })">📊 Simular</button><button onclick="eliminarAcao('${

            doc.id

          }')">🗑️ Eliminar</button>

            </div>

          </li>`;

          count++;

        }

      });

      html += "</ul>";

      resultadoDiv.innerHTML =

        count > 0

          ? html

          : "<p>Nenhuma ação encontrada com os filtros aplicados.</p>";

    })

    .catch((error) => {

      console.error("Erro ao filtrar:", error);

      resultadoDiv.innerHTML =

        "<p style='color:red;'>Erro ao carregar dados.</p>";

    });

}

//Atualizar a Firebase

function atualizarAcaoFirebase() {

  if (!idAcaoEmEdicao) {

    alert("Nenhuma ação selecionada para edição.");

    return;

  }

  const nome = document.getElementById("nomeAcaoReg").value.trim();

  const ticker = document.getElementById("tickerAcaoReg").value.trim();

  const setor = document.getElementById("Setor").value;

  const mercado = document.getElementById("Mercado").value;

  const periodicidade = document.getElementById("Periodicidade").value;

  const dividendo = parseFloat(document.getElementById("valorDividendoReg").value);

  const mes = document.getElementById("mesDividendoReg").value;

  if (!nome || !ticker || !setor || !mercado || isNaN(dividendo) || !mes || !periodicidade) {

    alert("Preenche todos os campos corretamente.");

    return;

  }

  db.collection("acoesDividendos")

    .doc(idAcaoEmEdicao)

    .update({

      nome,

      ticker,

      setor,

      mercado,

      dividendo,

      periodicidade,

      mes,

      timestamp: new Date(),

    })

    .then(() => {

      alert("✅ Ação atualizada com sucesso!");

      limparCamposSec6();

      idAcaoEmEdicao = null;

    })

    .catch((error) => {

      console.error("Erro ao atualizar:", error);

      alert("❌ Erro ao atualizar. Tenta novamente.");

    });

}

// Abrir popup e mostrar todas as empresas por defeito

function abrirPopupFiltro() {

  filtrarAcoes(); // mostra todas as empresas

  document.getElementById("popupFiltro").classList.add("show");

}

// Fechar popup

function fecharPopupFiltro() {

  document.getElementById("popupFiltro").classList.remove("show");

}

// preparar simulação do screen 6 para o 5

function prepararSimulacao(nome, valorStock, dividendo) {

  fecharPopupFiltro(); // fecha o popup ao simular

  console.log("Preparar simulação para:", nome, "com dividendo:", dividendo, "e TP1:", valorStock);

  abrirSecao(5); // Vai para screen simulação

  document.getElementById("nomeAcao").value = nome;

  document.getElementById("dividendo").value = dividendo || 0;

  document.getElementById("tp1").value = valorStock || "";

  document.getElementById("tp2").value = "";

  document.getElementById("investimento").value = "";

  document.getElementById("botoesSimulacaoRapida").classList.remove("hidden");

}

// editar ação

function editarAcao(id, dados) {

  fecharPopupFiltro(); // fecha o popup ao editar

  idAcaoEmEdicao = id;

  document.getElementById("nomeAcaoReg").value = dados.nome || "";

  document.getElementById("tickerAcaoReg").value = dados.ticker || "";

  document.getElementById("Setor").value = dados.setor || "";

  document.getElementById("Mercado").value = dados.mercado || "";

  document.getElementById("valorDividendoReg").value = dados.dividendo || "";

  document.getElementById("mesDividendoReg").value = dados.mes || "";

  document.getElementById("periodicidade").value = dados.periodicidade || "";

}

//Função simularValorRapido(valor)

function simularValorRapido(valor) {

  const tp1 = parseFloat(document.getElementById("tp1").value);

  const tp2 = parseFloat(document.getElementById("tp2").value);

  const dividendo = parseFloat(document.getElementById("dividendo").value || 0);

  if (isNaN(tp1) || isNaN(tp2)) {

    alert("⚠️ Introduz TP1 e TP2 antes de simular.");

    return;

  }

  document.getElementById("investimento").value = valor;

  guardarSimulacao(

    document.getElementById("nomeAcao").value,

    tp1,

    tp2,

    valor,

    dividendo

  );

  // Esconde os botões após simular

  //document.getElementById("botoesSimulacaoRapida").classList.add("hidden");

}

//Quando clico no Botão Voltar no menu Simular

function voltarMenu() {

  // Esconde todas as secções

  const secoes = document.querySelectorAll(".screen");

  secoes.forEach((secao) => secao.classList.add("hidden"));

  // Mostra apenas o menu principal

  document.getElementById("screen2").classList.remove("hidden");

  // Esconde os botões de simulação rápida

  const botoes = document.getElementById("botoesSimulacaoRapida");

  if (botoes) botoes.classList.add("hidden");

}

//Guardar na Firebase

let idAcaoEmEdicao = null;

function guardarOuAtualizarAcaoFirebase() {

  const nome = document.getElementById("nomeAcaoReg").value.trim();

  const ticker = document.getElementById("tickerAcaoReg").value.trim();

  const setor = document.getElementById("Setor").value.trim();

  const mercado = document.getElementById("Mercado").value.trim();

  const dividendo = parseFloat(document.getElementById("valorDividendoReg").value);

  const mes = document.getElementById("mesDividendoReg").value;

  const periodicidade = document.getElementById("periodicidade").value;

  if (!nome || !ticker || !setor || !mercado || isNaN(dividendo) || !mes || !periodicidade) {

    alert("⚠️ Preenche todos os campos corretamente.");

    return;

  }

  const dadosAcao = {

    nome,

    ticker,

    setor,

    mercado,

    dividendo,

    mes,

    periodicidade,

    timestamp: new Date()

  };

  if (idAcaoEmEdicao) {

    // Atualizar ação existente

    db.collection("acoesDividendos")

      .doc(idAcaoEmEdicao)

      .update(dadosAcao)

      .then(() => {

        alert("✅ Ação atualizada com sucesso!");

        limparCamposSec6();

        idAcaoEmEdicao = null;

      })

      .catch((error) => {

        console.error("Erro ao atualizar:", error);

        alert("❌ Erro ao atualizar. Tenta novamente.");

      });

  } else {

    // Guardar nova ação

    db.collection("acoesDividendos")

      .add(dadosAcao)

      .then(() => {

        alert("✅ Ação guardada com sucesso na Firebase!");

        limparCamposSec6();

      })

      .catch((error) => {

        console.error("Erro ao guardar ação:", error);

        alert("❌ Ocorreu um erro ao guardar. Verifica a ligação com a Firebase.");

      });

  }

}

//Eliminar um Registo

function eliminarAcao(id) {

  if (confirm("⚠️ Tens a certeza que queres eliminar esta ação?")) {

    db.collection("acoesDividendos")

      .doc(id)

      .delete()

      .then(() => {

        alert("✅ Ação eliminada com sucesso.");

        filtrarAcoes(); // Recarrega os resultados após eliminar

      })

      .catch((error) => {

        console.error("Erro ao eliminar ação:", error);

        alert("❌ Erro ao eliminar. Tenta novamente.");

      });

  }

}