

FazIA — Pacote do projeto

Este documento contém os arquivos prontos para copiar/colar no seu projeto local: `index.html`, `style.css`, `app.js`, `README.md` e `logo-fazia.svg`.

Instrução: Copie cada bloco para um arquivo com o mesmo nome dentro de uma pasta do projeto, abra `index.html` no navegador e tudo deverá funcionar (uso offline via armazenamento local).

`index.html`

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <title>FazIA – Tarefas inteligentes e DRE</title>
  <link rel="stylesheet" href="style.css">
  <!-- Chart.js CDN -->
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
  <header class="top">
    <div class="brand">
      
      <h1>FazIA</h1>
    </div>
    <div class="controls">
      <button id="themeToggle">Modo escuro</button>
      <button id="exportCsv">Exportar DRE (CSV)</button>
      <button id="clearAll">Limpar Tudo</button>
    </div>
  </header>

  <main class="container">
    <section class="form">
      <h2>Nova tarefa</h2>
      <input id="newTodo" placeholder="Descrição da tarefa" />
      <input id="newDueDate" type="date" />
      <input id="newCategory" placeholder="Categoria (ex: Vendas, Marketing)" />
      <select id="newTipo">
        <option value="Neutro">Neutro</option>
        <option value="Receita">Receita</option>
        <option value="Despesa">Despesa</option>
      </select>
      <input id="newValue" type="number" step="0.01" placeholder="Valor (ex: 10.50)" />
    </section>
  </main>
</body>
```

```

1250.00)" />
    <div class="form-actions">
        <button id="addBtn">Adicionar</button>
        <button id="importSample">Importar Exemplo</button>
    </div>
</section>

<section class="list">
    <div class="search-filter">
        <input id="search" placeholder="Pesquisar..." />
        <select id="filterTipo">
            <option value="Todas">Todas</option>
            <option value="Receita">Receita</option>
            <option value="Despesa">Despesa</option>
            <option value="Neutro">Neutro</option>
        </select>
        <select id="filterCategoria">
            <option value="Todas">Todas</option>
        </select>
    </div>

    <div id="todoList" class="todos"></div>

    <div class="summary">
        <div>Receitas: <span id="sumReceita">0</span></div>
        <div>Despesas: <span id="sumDespesa">0</span></div>
        <div>Resultado: <span id="resultado">0</span></div>
    </div>

    <canvas id="dreChart" width="400" height="120"></canvas>

    <div id="aiSuggestion" class="ai-box"></div>
</section>
</main>

<script src="app.js"></script>
</body>
</html>

```

style.css

```

:root{
    --primary:#2176ff;
    --bg:#f4f8fb;
    --card:#fff;
    --muted:#666;
}
*{box-sizing:border-box}

```

```

body{font-family:Inter, Arial, sans-serif;margin:0;background:var(--bg);color:#111}
.top{display:flex;justify-content:space-between;align-items:center;padding:12px 18px;background:linear-gradient(90deg,#fff,#f0f6ff);border-bottom:1px solid #e6eefc}
.brand{display:flex;align-items:center;gap:12px}
.logo{height:40px}
.container{display:flex;gap:20px;padding:20px;max-width:1100px;margin:18px auto}
.form{flex:0 0 320px;background:var(--card);padding:16px;border-radius:12px;box-shadow:0 6px 18px rgba(33,118,255,0.06)}
.form input, .form select{width:100%;padding:8px;margin:8px 0;border-radius:8px;border:1px solid #dfe8ff}
.form-actions{display:flex;gap:8px}
.form button{flex:1;padding:8px;border-radius:8px;border:none;background:var(--primary);color:white}
.list{flex:1;background:var(--card);padding:16px;border-radius:12px;box-shadow:0 6px 18px rgba(0,0,0,0.04)}
.search-filter{display:flex;gap:8px;margin-bottom:8px}
.search-filter input{flex:1;padding:8px;border-radius:8px;border:1px solid #e3eefc}
.todos{display:flex;flex-direction:column;gap:8px;max-height:360px;overflow:auto;padding-right:8px}
.todo{display:flex;justify-content:space-between;align-items:center;padding:10px;border-radius:8px;border:1px solid #eef6ff}
.todo .left{display:flex;flex-direction:column}
.todo .meta{font-size:12px;color:var(--muted)}
.todo .actions button{margin-left:6px}
.summary{display:flex;gap:16px;margin-top:12px;font-weight:600}
.ai-box{margin-top:12px;padding:10px;border-radius:8px;background:#ffffbe6;border:1px solid #fff1b8}
.dark{background:#0b1020;color:#ddd}
.dark .form, .dark .list{background:#0f1724}
@media (max-width:900px){.container{flex-direction:column}.form{width:100%}}

```

app.js

```

// FazIA - app.js (versão funcional com localStorage, DRE e sugestão básica de IA)
const KEY = 'fazia_todos_v1';

// DOM
const todoList = document.getElementById('todoList');
const newTodo = document.getElementById('newTodo');
const newDueDate = document.getElementById('newDueDate');
const newCategory = document.getElementById('newCategory');
const newTipo = document.getElementById('newTipo');
const newValue = document.getElementById('newValue');

```

```

const addBtn = document.getElementById('addBtn');
const clearAll = document.getElementById('clearAll');
const search = document.getElementById('search');
const filterTipo = document.getElementById('filterTipo');
const filterCategoria = document.getElementById('filterCategoria');
const sumReceita = document.getElementById('sumReceita');
const sumDespesa = document.getElementById('sumDespesa');
const resultado = document.getElementById('resultado');
const dreChart = document.getElementById('dreChart');
const aiSuggestion = document.getElementById('aiSuggestion');
const exportCsv = document.getElementById('exportCsv');
const themeToggle = document.getElementById('themeToggle');
const importSample = document.getElementById('importSample');

let chart = null;

function getTodos(){
    try{ return JSON.parse(localStorage.getItem(KEY) || '[]'); }catch(e)
{return []}
}
function saveTodos(t){localStorage.setItem(KEY, JSON.stringify(t));}

function uid(){return Date.now().toString(36)
+Math.random().toString(36).slice(2,6)}

function renderTodos(){
    const todos = getTodos();
    // update category filter
    const cats = ['Todas', ...Array.from(new Set(todos.map(t=>t.category||'Sem
categoria')))].slice(0,50);
    filterCategoria.innerHTML = '';
    cats.forEach(c=>{
        const opt = document.createElement('option'); opt.value = c;
        opt.textContent = c; filterCategoria.appendChild(opt);
    });

    // apply filters
    const q = (search.value||'').toLowerCase();
    const tipo = filterTipo.value;
    const cat = filterCategoria.value;
    const shown = todos.filter(t=>{
        if(tipo!=='Todas' && t.type!==tipo) return false;
        if(cat!=='Todas' && (t.category||'Sem categoria')!==cat) return false;
        if(q && !((t.text||'').toLowerCase().includes(q) ||
(t.category||'').toLowerCase().includes(q))) return false;
        return true;
    });

    todoList.innerHTML = '';
    shown.forEach(t=>{
        const div = document.createElement('div'); div.className='todo';

```

```

        const left = document.createElement('div'); left.className='left';
        const title = document.createElement('div'); title.textContent = t.text
+ (t.type==='Receita'? '+' : '');
        const meta = document.createElement('div'); meta.className='meta';
        meta.textContent = `${t.type} • ${t.category||'Sem categoria'} • $
{t.dueDate || 'sem data'} • R$ ${Number(t.value||0).toFixed(2)}`;
        left.appendChild(title); left.appendChild(meta);

        const actions = document.createElement('div');
        actions.className='actions';
        const done = document.createElement('button'); done.textContent =
t.completed? '✓' : '☐';
        done.onclick = ()=>{ toggleComplete(t.id); };
        const edit = document.createElement('button'); edit.textContent='✎';
        edit.onclick = ()=>{ startEdit(t.id); };
        const del = document.createElement('button'); del.textContent='✖';
        del.onclick = ()=>{ removeTodo(t.id); };
        actions.appendChild(done); actions.appendChild(edit);
        actions.appendChild(del);

        div.appendChild(left); div.appendChild(actions);
        todoList.appendChild(div);
    });

    updateSummary();
    updateChart();
    aiSuggest();
}

function addTodo(){
    const text = (newTodo.value||'').trim();
    if(!text) return alert('Digite a descrição da tarefa');
    const t = { id: uid(), text, dueDate: newDueDate.value||null, category:
newCategory.value||'Sem categoria', type: newTipo.value||'Neutro', value:
Number(newValor.value||0), completed:false, createdAt: new
Date().toISOString() };
    const todos = getTodos(); todos.push(t); saveTodos(todos);
    newTodo.value=''; newValor.value=''; newCategory.value='';
    newDueDate.value=''; newTipo.value='Neutro';
    renderTodos();
}

function toggleComplete(id){ const todos=getTodos(); const
i=todos.findIndex(x=>x.id==id); if(i>-1){ todos[i].completed!=
todos[i].completed; saveTodos(todos); renderTodos(); }}
function removeTodo(id){ if(!confirm('Remover esta tarefa?')) return; const
todos=getTodos().filter(t=>t.id!=id); saveTodos(todos); renderTodos(); }

function startEdit(id){ const todos=getTodos(); const
t=todos.find(x=>x.id==id); if(!t) return; const newText =
prompt('Descrição', t.text); if(newText==null) return; t.text = newText;
}

```

```

t.type = prompt('Tipo (Receita/Despesa/Neutro)', t.type) || t.type;
t.category = prompt('Categoria', t.category) || t.category; t.value =
Number(prompt('Valor', t.value)||t.value); t.dueDate = prompt('Data (YYYY-MM-
DD)', t.dueDate)||t.dueDate; saveTodos(todos); renderTodos(); }

function updateSummary(){ const todos=getTodos(); const receitas =
todos.filter(t=>t.type==='Receita').reduce((s,t)=>s+Number(t.value||0),0);
const despesas =
todos.filter(t=>t.type==='Despesa').reduce((s,t)=>s+Number(t.value||0),0);
sumReceita.textContent = receitas.toFixed(2); sumDespesa.textContent =
despesas.toFixed(2); resultado.textContent = (receitas-
despesas).toFixed(2); }

function updateChart(){ const todos=getTodos(); const byCat = {};
todos.forEach(t=>{ const c = t.category|| 'Sem categoria'; byCat[c] =
byCat[c] || { receita:0, despesa:0 }; if(t.type==='Receita')
byCat[c].receita += Number(t.value||0); if(t.type==='Despesa')
byCat[c].despesa += Number(t.value||0); });
const labels = Object.keys(byCat).slice(0,10);
const receitas = labels.map(l=>byCat[l].receita||0);
const despesas = labels.map(l=>byCat[l].despesa||0);
if(chart) chart.destroy();
chart = new Chart(dreChart.getContext('2d'),{ type:'bar', data:{ labels,
datasets:[{ label:'Receita', data:receitas },{ label:'Despesa',
data:despesas }] }, options:{ responsive:true,
maintainAspectRatio:false } });
}

function aiSuggest(){ const todos=getTodos(); if(todos.length==0){
aiSuggestion.textContent='Sem dados para análise.'; return; }
// Sugestão simples: categoria com maior despesa
const despByCat = {};
todos.forEach(t=>{ if(t.type==='Despesa'){ const c=t.category|| 'Sem
categoria'; despByCat[c]=(despByCat[c]||0)+Number(t.value||0); }});
const topDespCat = Object.keys(despByCat).sort((a,b)=>despByCat[b]-
despByCat[a])[0];
if(topDespCat){ aiSuggestion.innerHTML =
`Sugestão IA: Sua maior despesa está em <em>${topDespCat}</
em> (R$ ${despByCat[topDespCat].toFixed(2)}). Considere reduzir 10% nas
despesas dessa categoria e analise substituições.`; }
else { aiSuggestion.textContent = 'Nenhuma despesa encontrada para
análise.'; }
}

function exportCSV(){ const todos = getTodos(); if(todos.length==0) return
alert('Sem dados para exportar'); const header =
['id,text,category,type,value,dueDate,completed,createdAt'];
const rows = todos.map(t=> [t.id, `"${(t.text||'').replace(/\
g,'""')}` ,t.category,t.type,t.value,t.dueDate,t.completed,t.createdAt].join(','));
const csv = header.concat(rows).join('\n');
const blob = new Blob([csv],{type:'text/csv;charset=utf-8;'});

```

```

    const url = URL.createObjectURL(blob); const
a=document.createElement('a'); a.href=url;
a.download='fazia_dre_export.csv'; a.click(); URL.revokeObjectURL(url);
}

function removeAll(){ if(!confirm('Apagar todas as tarefas?')) return;
localStorage.removeItem(KEY); renderTodos(); }

function importSampleData(){ const sample = [ { id:uid(),
text:'Venda produto Y', category:'Vendas', type:'Receita', value:1250,
dueDate:'2025-11-25', completed:false, createdAt:new Date().toISOString()},
{ id:uid(), text:'Pagamento fornecedor XPTO', category:'Operacional',
type:'Despesa', value:500, dueDate:'2025-11-25', completed:false,
createdAt:new Date().toISOString() } ]; const todos =
getTodos().concat(sample); saveTodos(todos); renderTodos(); }

// Theme
function toggleTheme(){ document.body.classList.toggle('dark');
themeToggle.textContent = document.body.classList.contains('dark')? 'Modo
claro' : 'Modo escuro'; }

// Events
addBtn.onclick = addTodo;
clearAll.onclick = removeAll;
search.oninput = renderTodos;
filterTipo.onchange = renderTodos;
filterCategoria.onchange = renderTodos;
exportCsv.onclick = exportCSV;
themeToggle.onclick = toggleTheme;
importSample.onclick = importSampleData;

// keyboard: Enter to add
newTodo.addEventListener('keydown', e=>{ if(e.key==='Enter') addTodo(); });

// init
renderTodos();

```

README.md

```

# FazIA

FazIA – aplicativo de tarefas inteligentes com painel DRE (Receitas &
Despesas) e sugestões básicas de IA.

## Funcionalidades
- Cadastro de tarefas com: descrição, data, categoria, tipo (Receita/Despesa/
Neutro) e valor.
- Edição/remoção/ conclusão das tarefas.

```

- Dashboard DRE com resumo e gráfico por categoria (Chart.js).
- Exportação CSV do DRE.
- Sugestão básica de IA: identifica categoria com maior despesa.
- Armazenamento local (localStorage).
- Modo escuro/claro.

```
## Como usar
1. Copie os arquivos `index.html`, `style.css`, `app.js` e `logo-fazia.svg` para a mesma pasta.
2. Abra `index.html` no navegador.
3. Use "Importar Exemplo" para popular com dados de exemplo.
```

```
## Como subir para o GitHub (passo-a-passo rápido)
1. Crie um repositório no GitHub (por ex.: `FazIA`).
2. No seu computador, dentro da pasta do projeto:
```bash
git init
git add .
git commit -m "Initial commit - FazIA"
git branch -M main
substitua com sua URL do repositório
git remote add origin https://github.com/SEU_USUARIO/FazIA.git
git push -u origin main
```

3. Se preferir, crie um `README.md` (já incluído) e um `.gitignore` com `node_modules` (se aplicar).

## Próximos passos sugeridos

- Integração com backend (Firebase/Supabase) para multiusuário e sincronização.
- IA mais avançada (classificação automática por texto, previsões de fluxo de caixa).
- Exportar PDF e gerar relatórios mensais/ano a ano.

---

## Licença

Projeto de exemplo — sinta-se livre para copiar e adaptar.

```

`logo-fazia.svg`
```svg
<svg width="220" height="60" viewBox="0 0 220 60" fill="none" xmlns="http://www.w3.org/2000/svg">
  <rect x="2" y="2" width="56" height="56" rx="14" fill="#2176ff" stroke="#174ea6" stroke-width="2"/>
  <text x="14" y="42" font-family="Arial, sans-serif" font-size="34" font-weight="bold" fill="#fff">Fz</text>
  <rect x="67" y="11" width="147" height="38" rx="7" fill="#fff" stroke="#2176ff" stroke-width="2"/>
```

```
<text x="83" y="39" font-family="Arial, sans-serif" font-size="30"  
fill="#2176ff" font-weight="bold">Faz<tspan fill="#174ea6">IA</tspan></text>  
<circle cx="35" cy="27" r="5" fill="#fff"/>  
</svg>
```

Observações finais

- Este pacote foi pensado para ser um ponto de partida: o código é simples, comentado e fácil de estender.
- Se quiser, eu posso: criar o repositório no GitHub para você (preciso de autorização: token/username — **não** envie tokens por chat; eu posso gerar instruções para CLI ou GitHub web), ou transformar o projeto em PWA, ou adicionar autenticação.

Boa sorte! Copie os arquivos e me diga se quer que eu gere o `package.zip` ou um passo-a-passo visual para o GitHub.