Projeto — Guia Profissional de Desenvolvimento Web (HTML, CSS, JS)

Obs. É permitido o uso de inteligência artificial nessa etapa, contudo, é necessário que você crie um arquivo .MD chamado ChatIA e coloque todo a conversa que você teve com chats até o site ficar pronto.

Trabalho individual

Não serão aceitos commmits fora do horário de aula.

Trabalhos copiados e muito parecidos receberão nota zero.

É obrigatório personalizar o seu site com suas informações.

Objetivo

Construir um site educativo que explique:

1. **O que são HTML, CSS e JavaScript** e o papel de cada um.
2. **Boas práticas profissionais** (acessibilidade, semântica, responsividade, organização, versionamento).
3. **Panorama de tecnologias web** (front-end, back-end, bancos de dados, devops, testes) e **como escolher**.
4. **Fluxograma** de um projeto web completo (da ideia ao deploy).

Tecnologias permitidas: apenas **HTML5**, **CSS3** e **JavaScript ES6+** (sem frameworks). **Sem back-end**. Dados persistidos apenas com **localStorage** quando necessário.

Entregáveis

- **Site** com as páginas/seções pedidas (abaixo).
- **Fluxograma** (feito com **SVG no próprio site** ou imagem criada no diagrams.net e inserida no projeto).
- **README.md** com como abrir o projeto e créditos de conteúdo.
- **Repositório GitHub** público: guia-web-1ano-<seunome>, com commits claros.

Identidade Visual (obrigatória)

- **Cores**
    - Primária: #2563EB (azul)
    - Secundária: #10B981 (verde)
    - Neutros: #111827 (texto), #F3F4F6 (fundo claro), #E5E7EB (bordas)

- **Tipografia**
    - Títulos: “Poppins”, fallback sans-serif
    - Texto: “Inter”, fallback sans-serif
- **Tema claro/escuro** com toggle (salvar preferência em localStorage).

Estrutura de Pastas

/assets (imagens, ícones, svg)

/css

styles.css

/js

app.js

index.html (Home + navegação)

tecnologias.html (Panorama de tecnologias)

boas-praticas.html

fluxo.html (Fluxograma + etapas)

quiz.html (Quiz de 8-10 questões)

README.md

Requisitos de Conteúdo e Funcionalidade

1) Home (index.html)

- **Header fixo** com logo/título “Guia Profissional de Desenvolvimento Web” e **menu responsivo** (hambúrguer no mobile).
- **Hero** com frase-resumo e botão “Começar”.
- **Seção:** “O que é HTML, CSS, JS?”
    - Três **cards** com: definição curta (2–3 linhas), **exemplo mínimo** (um code block por card) e **quando usar**.
- **Footer** com links para páginas, créditos e ano.

Boas práticas exigidas

- HTML semântico (header, nav, main, section, article, footer).
- Imagens com alt descritivo.
- Navegação por teclado (foco visível).

---

2) Tecnologias (tecnologias.html)

Objetivo: panorama do ecossistema (explicativo, não usar essas techs no código).

- **Filtros (JS)** por categoria: *Front-end frameworks*, *Back-end*, *Bancos de dados*, *DevOps/Deploy*, *Testes*.
- **Grid de cards** com, no mínimo, 10 tecnologias (ex.: React, Vue, Angular; Node, Python/Django, Java/Spring; MySQL, PostgreSQL, MongoDB; Docker, CI/CD; Jest, Cypress).
- Cada card: **nome**, **para que serve**, **prós e contras** (2 de cada), **quando evitar** e **link oficial**.
- **Busca textual (JS)** filtra os cards em tempo real.
- **Modal de detalhes** ao clicar no card, com resumo "Quando escolher X?" (em 3 bullets).
- **Badge "Nível"** (básico/intermediário/avançado).

Regras

- Filtros e busca funcionam **sem recarregar** a página.
- Salvar no localStorage o **último filtro** aplicado.
- Acessibilidade: aria-controls, aria-expanded nos filtros e no modal.

---

3) Boas Práticas (boas-praticas.html)

Seções obrigatórias (cada uma com 3–5 bullets e **1 exemplo** em código):

- **Semântica & Acessibilidade** (landmarks, alt, labels, contraste).
- **Responsividade** (mobile-first, Flex/Grid, unidades relativas).
- **Organização de CSS** (BEM ou outra convenção), comentários e variáveis CSS.
- **Performance leve** (imagens otimizadas, lazy loading simples, evitar JS desnecessário).
- **Versionamento (Git)**: mensagens de commit, branches, frequência.
- **Boas práticas de JS**: funções puras quando possível, nomes claros, tratar erros.

Interatividade

- **Accordion (JS)**: cada tópico abre/fecha suavemente (transição CSS).
- **Checklist pessoal**: checkbox por item; progresso (%) salvo em localStorage.

---

4) Fluxo (fluxo.html)

Fluxograma de um projeto web completo:

- Etapas mínimas: **Descoberta → Requisitos → Protótipo → Design → Implementação → Testes → Preparação de deploy → Publicação → Monitoramento/Manutenção**.
- **Feito com SVG** (linhas, setas, caixas) **ou** imagem .png/.svg criada no diagrams.net e inserida na página.
- Ao clicar em uma etapa, abrir **tooltip** (JS) com o que entregar e **riscos comuns**.
- **Linha do tempo** horizontal (CSS) com as mesmas etapas (visão alternativa).

---

5) Quiz (quiz.html)

- **8–10 questões** de múltipla escolha (HTML/CSS/JS, boas práticas, escolhas de tecnologia).
- Mostrar **pontuação ao final** e **explicações** das respostas.
- Salvar melhor pontuação em localStorage.

---

6) Requisitos de CSS

- **Responsivo** (mobile primeiro), sem quebra horizontal.
- **Grid/Flex** para layouts.
- **Variáveis CSS** para cores e tipografia.
- **Focus styles** visíveis (não remover outline).
- **Sem frameworks** (Bootstrap/Tailwind não).

---

7) Requisitos de JavaScript

- JS **modular** (funções organizadas em app.js).
- **Sem dependências externas**.
- Usar addEventListener, manipulação DOM, classList, dataset.
- Salvar preferências (tema, filtros, quiz) no **localStorage**.
- Tratar erros de interação (inputs vazios, busca sem resultado).

Peculiaridades (escolher 3 para implementar)

1. **Tema claro/escuro** (toggle com persistência).
2. **Exportar CSV** dos cards filtrados de “Tecnologias” (nome, categoria, prós/contras).
3. **Teclas de atalho**: abrir busca ("/"), focar menu (“Alt+M”), subir (“Home”).

4. **Impressão**: CSS para imprimir “Boas Práticas” em 1 página.
5. **Animações suaves** (prefira transform/opacity, respeitando prefers-reduced-motion).
6. **Glossário** (mini buscas a termos como SSR, SPA, REST, CI/CD).
7. **Validação de formulários** (se fizer um “fale conosco” fake, validar campos).
8. **Favoritos**: marcar cards de tecnologia como favoritos (ícone), salvar no localStorage.
9. **Roteiro de estudos**: gerar uma lista ordenada (array) com base nas escolhas do aluno (iniciante → avançado).
10. **Medidor de contraste**: campo para inserir 2 cores e calcular contraste AAA/AA (básico).

Acessibilidade (checagem mínima)

- Contraste texto/fundo ≥ 4.5:1 em texto padrão.
- Labels conectadas a inputs (for/id).
- Ordem de tabulação coerente.
- aria-* em componentes interativos (accordion, modal, toggles).
- Título de página (<title>) significativo.

Git/GitHub (obrigatório)

- Repositório **público**: guia-web-1ano-<seunome>.
- Commits **pequenos e frequentes** (mensagens: feat: filtro por categoria, fix: foco em modal ...).
- README com: objetivo, estrutura, como rodar (abrir index.html), recursos implementados e créditos de imagens/ícones.

Rubrica (100 pts)

- **Conteúdo técnico correto e claro** — 25
- **HTML semântico + acessibilidade** — 15
- **CSS responsivo e organizado** — 15
- **JS funcional (filtros, modais, quiz, localStorage)** — 20
- **Peculiaridades (3 de 10)** — 10
- **Git/README/estrutura** — 10

- **Apresentação/clareza** — 5

Penalidades: copiar conteúdo sem entender (será verificado no quiz/apresentação), ausência de commits granulares, site não responsivo.

Questionário de Defesa (para prova/entrevista)

1. Em 3–5 linhas, **diferencie** HTML, CSS e JS com exemplos do seu projeto.
2. Cite **5 boas práticas** que você aplicou e onde.
3. Mostre **como** seu filtro por categoria funciona (DOM + eventos).
4. Explique seu **fluxograma**: por que essas etapas, riscos e entregas.
5. Quando **escolher** React/Angular/Vue, ou permanecer com JS puro? Dê **2 prós e 2 contras** de framework.
6. O que é **semântica** e como ela ajuda na **acessibilidade** e **SEO**?
7. Como você **persistiu** dados no front (localStorage)? Vantagens e limites.
8. Qual seria um **próximo passo** profissional (build, testes, deploy)? Por quê?