

# Lucas Queiroz Vieira

Parque Novo Mundo - SP • +55 (11)-94574-5870 • [lqvieira7@gmail.com](mailto:lqvieira7@gmail.com)

**Desenvolvedor Front-End | HTML • CSS (BEM, Grid/Flex, Design Tokens, Media Queries/Responsividade) • JavaScript (ES6+) | Git (fluxo GitHub) | Acessibilidade (WCAG 2.2/ARIA 1.2) • Performance (Lighthouse/Core Web Vitals) • SEO Técnico**

## Stack central

HTML, CSS, JavaScript, Git.

## Diferenciais

Acessibilidade (WCAG 2.2/ARIA 1.2), Performance (Lighthouse/Core Web Vitals), SEO técnico, arquitetura CSS (BEM/design tokens), responsividade avançada (media queries, container queries, Grid/Flex), JS modular, Git (branches/PRs), automação simples (ESLint/Prettier, npm scripts), CI/CD básico (GitHub Actions), deploy (Vercel/Netlify/GitHub Pages)..

## Sobre

Desenvolvedor Front-End focado em interfaces estáticas e interativas com HTML semântico, CSS modular (BEM + design tokens) e JavaScript ES6+ em módulos. Entrego acessibilidade aplicável (WCAG 2.2, ARIA 1.2, navegação por teclado, foco visível, mensagens via aria-live), performance mensurada (Lighthouse/Core Web Vitals: LCP, CLS, INP; PageSpeed Insights/WebPageTest) e SEO técnico (metas, canonical, hreflang, schema, sitemaps, robots). Padronizo DX com ESLint/Prettier, npm scripts, estrutura de pastas consistente, CI leve (GitHub Actions) e deploy contínuo (Vercel/Netlify/GitHub Pages). Comunicação 100% técnica, estritamente em contexto de Front-End.

Em versionamento, opero Git com rigor de fluxo: convenções de branch (feat/, fix/, chore/), Conventional Commits, PRs com checklist técnico, squash/merge para histórico legível, tags semânticas de release. Rebase interativo para linearização, resolução de conflitos por hunk com preservação de intenção, cherry-pick em hotfixes, bisect para diagnóstico. DAG consistente; reescrita de histórico apenas em branches de trabalho.

Em layout responsivo, aplico mobile-first, breakpoints derivados de content-width, container queries quando aplicável, Grid para macro-layout e Flex para microfluxo, tipografia fluida via clamp() e tokens responsivos. Imagens com srcset/sizes e dimensões intrínsecas para reduzir CLS; prefers-reduced-motion e prefers-color-scheme para respeito a preferências do usuário.

## Diferenciais Técnicos

- HTML semântico: landmarks, headings hierárquicos, microcopy orientada à navegação, accessibility tree coerente
- CSS profissional: BEM, design tokens, Grid/Flex em camadas (macro/micro), media queries avançadas e container queries, tipografia fluida (clamp), dark mode via CSS variables, responsividade mobile-first
- JavaScript ES6+ modular: organização por módulos, fetch/JSON, localStorage, padrões de interação acessíveis; controle de foco e rolagem
- Acessibilidade aplicada: roles/ARIA 1.2, foco visível, navegação por teclado full, aria-live em atualizações; validações com feedback acessível
- Performance: otimização de fontes/imagens, lazy loading, preload/preconnect, srcset/sizes, dimensões intrínsecas; HTTP caching (Cache-Control/ETag), compressão (Brotli/Gzip)
- Métricas: relatórios Lighthouse e Core Web Vitals com metas e evidências (LCP, CLS, INP); PageSpeed Insights/WebPageTest
- SEO técnico: title/description coerentes, OG/Twitter, headings, canonical, hreflang, sitemaps, robots, schema básico
- Dev Experience: Git/GitHub (PRs e code review objetivos), ESLint/Prettier, npm scripts, GitHub Actions (lint/build), deploy em Vercel/Netlify/GH Pages
- Git: branching model limpo (main protegida, develop opcional), feature branches curtas, PRs pequenos; rebase interativo; cherry-pick/bisect; tags semânticas; commit signing (GPG) opcional
- Documentação e qualidade: READMEs com arquitetura, padrões, checklists e roadmap; Issues/Projects no GitHub

## Portfólio

Portal Corporativo Static (HTML, CSS, JS)

- Stack: HTML semântico, CSS (BEM, Grid/Flex, design tokens, dark mode), JS modular (menu acessível, carrossel sem dependências)
- Layout: Grid em regiões estruturais; Flex em componentes; breakpoints por conteúdo; tipografia fluida; prefers-color-scheme
- A11y: landmarks, aria-\* em componentes interativos, foco visível, navegação por teclado 100%
- Performance: LCP < 2.5s (4G simulado), CLS 0.02; preload de fontes/imagens; lazy loading; caching
- SEO: headings coerentes, metas, Open Graph, sitemap, robots, canonical
- Dev: Git (feature branches, PRs curtos, Conventional Commits), ESLint/Prettier, npm scripts, GitHub Actions (lint), deploy em Vercel
- Links: DEMO | Código | README (métricas, arquitetura e checklist WCAG)

## Design System Lite (Tokens + Componentes Acessíveis)

- Tokens: cores, tipografia, espaçamento, sombras, motion; CSS variables com temas (light/dark)
- Componentes sem framework: Button, Alert, Modal, Tabs, Accordion, Dropdown, Tooltip
- Responsividade: tokens responsivos; utilitários de Grid/Flex; media/container queries utilitárias; comportamento adaptativo
- A11y: roles/ARIA corretos; keyboard traps evitados; focus management; testes manuais
- Documentação: catálogo com exemplos e código copiável
- Dev: Git (branches/PRs), ESLint/Prettier, npm scripts, GitHub Actions, deploy
- Links: DEMO | Código | README (guia de contribuição e convenções)

## **Observação**

Todos os projetos com README que inclui: objetivos, arquitetura, estrutura de pastas, padrões (BEM/naming), critérios de acessibilidade, métricas de performance (prints Lighthouse), rotas, decisões técnicas, limitações e próximos passos, estratégia de Git (naming de branches, Conventional Commits, checklist de PR), estratégia responsiva (breakpoints por conteúdo, container queries, tipografia fluida, Grid/Flex em camadas) e critérios de ausência de regressão de CLS.

## **Princípios de Qualidade e Processo**

- Consistência visual e semântica: tokens, BEM, headings documentados
- A11y como requisito: teclado primeiro, ARIA 1.2, testes manuais
- Performance orientada a objetivos: metas de LCP/CLS/INP com antes/depois
- SEO técnico embutido: indexação e preview social; canonical e hreflang
- DX sustentável: lint/format/CI, estrutura clara, scripts simples, deploy automatizado
- Documentação como parte do produto: READMEs, prints Lighthouse, checklists WCAG, decisões e trade-offs
- Git explícito e legível: histórico linear via rebase interativo; PRs pequenos; tags semânticas; proteção de main; política de revisão objetiva
- Responsividade orientada a conteúdo: breakpoints semânticos; tipografia/spacing fluidos; redução de CLS em mudanças de layout

## **Competências**

- HTML5 semântico
- CSS3: BEM; Grid (template areas, auto-fit/auto-fill, minmax()); Flexbox (gap, wrap, align/justify avançado); Design Tokens; Media Queries (width/height/orientation/reduced-motion/color-scheme); Container Queries; Responsividade mobile-first; tipografia fluida (clamp); Dark Mode
- JavaScript (ES6+): módulos, fetch, localStorage, DOM, padrões de interação e foco
- Acessibilidade: WCAG 2.2, ARIA 1.2, navegação por teclado, foco
- Performance: Lighthouse; Core Web Vitals (LCP, CLS, INP); lazy loading; preconnect/preload; HTTP caching; compressão
- SEO Técnico: meta tags, OG/Twitter, canonical, hreflang, sitemaps, robots, schema
- Ferramentas: Git/GitHub (branches/PRs, rebase interativo, cherry-pick, bisect, Conventional Commits, tagging), ESLint, Prettier, npm scripts, GitHub Actions, Vercel/Netlify/GitHub Pages
- UI/UX: Figma (handoff, protótipos low-fi), microcopy

## **Links Gerais**

GitHub/Portfólio: [github.com/lcsqueiroz](https://github.com/lcsqueiroz)

LinkedIn: [linkedin.com/in/lcsqueiroz](https://linkedin.com/in/lcsqueiroz)

E-mail: [lqvieira7@gmail.com](mailto:lqvieira7@gmail.com)

## **Formação acadêmica**

Ciência da Computação (2025 - 2029) - Carlos Drummond de Andrade

## **Licenças e Certificados**

- HTML e CSS para Iniciantes
- UI Design para Iniciantes
- JavaScript Completo ES6

## **Idiomas**

- Português — Nativo
- Inglês — Intermediário (leitura avançada; conversação intermediária)