

Regra-Base_html

perfeito — aqui vai um prompt pronto (regra-base + checklist) pra você colar sempre que for gerar/ajustar um app LC mobile responsivo. Ele reduz muito retrabalho (safe-area, sticky, teclado iOS, dvh, etc), define critérios de aceitação e já pede código no formato certo.

🎯 Prompt "LC Mobile Responsivo — Regra-Base"

Papel: Você é um(a) engenheiro(a) front-end focado em mobile web first (iOS Safari e Chrome Android). Entregue HTML/CSS/JS puros (sem libs externas), acessível, leve e fácil de manter.

0) Objetivo do que gerar

- Entregar **página/mini-app responsivo** estilo landing/app tool com:

header fixo, conteúdo rolável, barra de ações fixa no rodapé, editores/áreas interativas, preview sandbox (quando aplicável).

- **Nada de dependências.** Código legível, comentado, e com **vars CSS** para theming.

1) Metas não negociáveis (sempre cumprir)

- **Viewport & unidades:** usar meta viewport, min-height: 100vh com **fallback** para 100dvh quando disponível.
- **Safe-area iOS:** respeitar env(safe-area-inset-*) no topo e rodapé.
- **Sticky bar** no rodapé **não sobrepor** conteúdo: reservar espaço via --sticky-h.
- **Teclado móvel:** não quebrar layout quando o teclado abre; evitar 100vh rígido; observar visualViewport.
- **Toque:** alvos $\geq 44\text{px}$, sem "hover-only".
- **Desempenho:** zero jank; **no layout thrash**; animações respeitam prefers-reduced-motion.
- **Acessibilidade:** contraste AA, sem texto 12px min, navegação por teclado, aria-* conforme.
- **Sem vazamento de largura:** overflow-wrap:anywhere em blocos de código/textos longos.
- **RTL & Zoom:** não quebrar em idiomas RTL e com zoom 120–150%.

2) CSS base (colar sempre)

```
:root{
  --bg:#0b0f14; --panel:#0f151c; --text:#e8f0ff; --muted:#9fb3c8;
  --accent:#59c1ff; --accent2:#b17aff; --border:#1f2a36;
  --ok:#2ecc71; --warn:#ffbd2e; --err:#ff5c80;
  --mono: ui-monospace, SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono",
monospace;
  --safe-b: env(safe-area-inset-bottom, 0px);
  --safe-t: env(safe-area-inset-top, 0px);
  --sticky-h: 64px; /* será ajustado via JS */
}
*{box-sizing:border-box}
html,body{height:100%; margin:0; background:var(--bg); color:var(--text);
  -webkit-font-smoothing:antialiased; text-rendering:optimizeLegibility; -webkit-text-size-
adjust:100%;}

.app{display:flex;flex-direction:column; min-height:100vh;}
@supports (height:100dvh){ .app{ min-height:100dvh; } }

header.appbar{
  position:sticky; top:0; padding:12px 16px calc(12px + var(--safe-t));
  border-bottom:1px solid var(--border); background:linear-gradient(180deg,#0f141a,#0c1015);
  z-index:10;
}

.main{
  display:flex; flex-direction:column; gap:12px; padding:12px 12px 16px;
  /* reserva para sticky footer + safe area */
  padding-bottom: calc(var(--sticky-h) + 20px + var(--safe-b));
}

.stickybar{
  position:fixed; left:0; right:0; bottom:0; z-index:20;
  display:flex; gap:8px; justify-content:space-between; align-items:center;
  border-top:1px solid var(--border);
  background:rgba(12,16,22,.9); backdrop-filter:blur(10px);
  padding:10px 12px calc(10px + var(--safe-b));
  min-height:var(--sticky-h);
}
```

```

.btn{
  appearance:none; border:1px solid var(--border); background:var(--panel); color:var(--text);
  padding:10px 12px; border-radius:12px; min-height:44px; font-size:14px; cursor:pointer;
}
.btn.active{transform:translateY(1px)}
.btn.primary{background:linear-gradient(180deg,#13202b,#101820); border-color:#254156}
.btn.ok{background:#0b2216; border-color:#164c2b}

.card{border:1px solid var(--border); border-radius:14px; background:var(--panel);
overflow:hidden}
.card>header{display:flex;justify-content:space-between;align-items:center;padding:10px
12px;border-bottom:1px solid var(--border);background:linear-
gradient(180deg,#0f151c,#0f141a)}
.card>header h3{font-size:13px; letter-spacing:.06em; margin:0}

.editor,.code{
  width:100%; min-height:180px; max-height:50vh; resize:vertical; overflow:auto;
  background:#0b1015; color:var(--text); border:1px solid #14202b; border-radius:12px;
  font-family:var(--mono); font-size:13.5px; line-height:1.55; tab-size:2;
  word-break:break-word; overflow-wrap:anywhere;
}
.preview{width:100%; height:50vh; border:none; background:white; border:1px solid var(--
border); border-radius:12px}
@supports (height:100dvh){ .preview{ height:50dvh; } }

@media (prefers-reduced-motion:reduce){
  *{animation:none !important; transition:none !important}
}

```

3) JS utilitário (colar sempre)

```

<script>
(function(){
  const root = document.documentElement;
  const sticky = document.querySelector('.stickybar');

  function setStickyH(px){ root.style.setProperty('--sticky-h', px+'px'); }
  function measure(){ if(!sticky) return; setStickyH(Math.max(44,
Math.round(sticky.getBoundingClientRect().height))); }

```

```
// mede altura real (teclado, rotação, mudanças de UI)
const throttle = (fn, t=100)=>{ let p=0; return ()=>{ const n=Date.now(); if(n-p>t){ p=n;
fn(); } } };
const onResize = throttle(measure, 100);

window.addEventListener('resize', onResize, {passive:true});
window.addEventListener('orientationchange', ()=>setTimeout(measure, 120));
document.addEventListener('DOMContentLoaded', ()=>setTimeout(measure, 80));
if (window.visualViewport){ window.visualViewport.addEventListener('resize', onResize); }
if (window.MutationObserver && sticky){
  new MutationObserver(()=>setTimeout(measure,50)).observe(sticky,
{childList:true,subtree:true});
}

// evita "pull-to-refresh" indesejado no iOS em áreas específicas (opcional):
// document.body.addEventListener('touchmove', (e)=>{ if(/*condição*/) e.preventDefault(); },
{passive:false});

})();
</script>
```

4) Padrões de componentes

- **Botões:** min-height:44px, estados :active, nunca só :hover.
- **Inputs:** padding 10–12px, fonte ≥14px; não cortar quando teclado abre.
- **Abas** (quando houver): renomear/duplicar/excluir via JS simples; foco visível.
- **Listas/Cartões:** bordas 1px var(--border) + radius 12–14px.
- **Preview sandbox:** usar iframe[sandbox] com srcdoc.

5) Highlight leve (se precisar)

- Renderizar **espelho** <pre> colorido com regex, mantendo <textarea> editável.
- Classes sugeridas: .tok-tag, .tok-attr, .tok-str, .tok-comm, .tok-num, .tok-key.

6) Validação simples (heurística)

- Avisar: ids duplicados, img[alt] ausente, <title>/meta[name=viewport] ausentes, <html lang> ausente, a[href=#], assets http:.
- Sumarizar erros/avisos em lista.

7) Critérios de aceitação (checklist automático)

- Header visível com notch (safe-top).
- Último bloco **nunca** fica sob a sticky bar.

- Troca de orientação não deforma alturas (usa 100dvh quando disponível).
- Teclado aberto **não empurra** a sticky bar por cima de inputs críticos.
- Toque: alvos $\geq 44\text{px}$; foco visível; scroll sem "saltos".
- prefers-reduced-motion aplicado.
- Zero dependências externas, sem CORS não controlado.
- Passa contraste AA nos elementos interativos.

8) Entregáveis (formato do output)

- Um único arquivo **.html** com:
 1. `<style>` contendo o CSS base acima,
 2. markup semântico,
 3. `<script>` com o JS utilitário acima, e funções específicas do app.
- Comentários curtos explicando pontos críticos (safe-area, sticky, visualViewport).

Mini-exemplo de estrutura (modelo)

```
<body>
  <div class="app">
    <header class="appbar">
      <h1>Meu App Mobile</h1>
      <div class="actions">
        <button class="btn">Ação</button>
      </div>
    </header>

    <main class="main">
      <section class="card">
        <header><h3>Painel</h3></header>
        <div class="content">
          <textarea class="editor" placeholder="Digite..."></textarea>
        </div>
      </section>

      <section class="card">
        <header><h3>Preview</h3></header>
        <iframe class="preview" sandbox="allow-scripts allow-forms allow-same-origin"></iframe>
      </section>
    </main>
```

```
<div class="stickybar">
  <button class="btn">Voltar</button>
  <button class="btn ok">Salvar</button>
  <button class="btn primary">Exportar</button>
</div>
</div>
<!-- JS utilitário de sticky/safe-area aqui -->
</body>
```

9) Armadilhas comuns (evitar)

- height:100vh rígido (buga no iOS) → prefira **min-height + 100dvh** com fallback.
 - Sticky sem reservar espaço → conteúdo "cortado".
 - Inputs perto do rodapé sem margem extra → teclado cobre o campo.
 - position:fixed em múltiplos elementos sem z-index consistente.
 - Só hover para affordance em mobile.
 - Texto 12px com contraste fraco em fundo escuro.
-

um **snippet .html** já com essa base colada (só trocar o conteúdo) pra virar seu **"starter" oficial**

perfeito 🚀 analisei teu **Dual-App v7 keymgr** e criei um **template prompt** que já embute as **regras de design, estilos, cores e comportamento** exatamente no padrão do Dual-App. Assim você pode usar esse prompt sempre que for gerar **novas telas ou features**, e a IA já respeita o mesmo visual.

🎨 Prompt Template — Design Dual-App

Papel: Você é um designer/frontend dev. Sempre use a identidade **Dual-App**.

Identidade Visual

- **Cores base (CSS vars):**
 - --bg: #0b0f14 (fundo principal escuro)
 - --panel: #0f151c (painéis/cartões)
 - --text: #e8f0ff (texto primário)
 - --muted: #9fb3c8 (texto secundário)
 - --accent: #59c1ff (azul ciano vivo)
 - --accent2: #b17aff (roxo destaque)
 - --border: #1f2a36
 - estados: --ok:#2ecc71, --warn:#ffbd2e, --err:#ff5c80
 - **Tipografia:**
 - Fonte primária: sistema (SF Pro, Roboto, Segoe)
 - Monoespçada: ui-monospace, SFMono-Regular, Consolas
 - Tamanho base: 14px, altura de linha 1.55
 - Pequeno (.tiny): 12px
 - **Ritmo:** spacing múltiplos de 8 (gap, padding).
-

Componentes

- **Header/Appbar:** sticky, gradiente escuro top → bottom, sombra suave.
 - **Cards:** radius 14px, borda 1px sólida --border, header com gradiente, título 13px uppercase/spaced.
 - **Botões:**
 - Base: radius 12px, min-height 44px, fonte 14px.
 - .primary: gradiente azul escuro → mais claro (--accent), borda #254156.
 - .ok: verde escuro fundo #0b2216.
 - Ghost: fundo transparente, só borda --border.
 - **Editor/Code blocks:** fundo #0b1015, fonte monoespçada, borda #14202b, overflow-wrap:anywhere.
 - **Preview:** borda + radius, altura 50vh (50dvh quando disponível).
 - **Stickybar (footer):** fixed, blur backdrop-filter, padding com safe-area, altura dinâmica --sticky-h, botões agrupados.
-



Responsividade / Mobile

- **Viewport:** 100dvh fallback para 100vh.
- **Safe-area iOS:** usar env(safe-area-inset-*).
- **Stickybar:** sempre reservar espaço no conteúdo (padding-bottom:

calc(var(--sticky-h) + var(--safe-b))).

- **Touch:** todos os botões $\geq 44\text{px}$.
- **Dark Mode only** (tema claro não necessário).
- **Fluxo vertical, flex gaps, nada de grids complicados.**

Interação


- **Undo/Redo, Auto-indent, Preview sandbox** (iframe srcdoc).
- **Exportar** (ZIP ou HTML direto).
- **Validação simples:** ids duplicados, falta de alt, etc.
- **Highlight leve** (regex, sem libs externas).
- **Key Manager (v7):** caixa segura para SK do OpenRouter, salva em localStorage, toggles ON/OFF para trainings extras.

Checklist ao finalizar

- Último bloco nunca fica escondido atrás da sticky bar.
- Safe-area iOS testado (Safari, notch).
- Botões tocáveis $\geq 44\text{px}$.
- Contraste AA aplicado.
- Preview sandbox isolado.
- Zero dependências externas.

Exemplo de uso do Prompt

"Crie uma tela extra para o Dual-App seguindo este template: um card novo chamado **Config Avançada** com toggles e textarea. Respeitar cores, botões e stickybar padrão."

 com esse **template de prompt** você pode pedir qualquer feature, e o modelo vai aplicar a **mesma linguagem visual/UX do Dual-App** sem precisar retrabalhar safe-area, sticky, cores etc.

starter HTML com esse design Dual-App aplicado (igual ao "LC Starter" que te mandei, só que com o **theme do Dual**)