

Análise de Responsividade CSS

Seguem os principais pontos que estão comprometendo a responsividade e como podemos melhorar de forma lógica e estruturada:

1. Uso excessivo de unidades fixas (px, vh) em vez de unidades fluidas:

Ex.: `.hero { height: 110vh; padding: 6vh 2rem 8vh; }`

- Solução: Definir `min-height: 100vh` em vez de `height: 110vh`, usar `clamp()` para padding e font-size, trocar pixels por `rem`.

2. Duplicação de animações e keyframes:

Há três definições idênticas de `@keyframes bounce` espalhadas e múltiplas versões de `.navbar`.

- Solução: Consolidar cada `@keyframes` e blocos repetidos em um só lugar.

3. Falta de abordagem mobile-first:

O CSS usa estilos para desktop e depois `@media(max-width:...)`.

- Solução: Inverter lógica: estilos base para mobile, depois `@media(min-width:768px)` para tablet, `@media(min-width:1024px)` para desktop.

4. Tamanhos de imagens/vídeos estáticos:

Ex.: `.about-image img { width: 280px; }`

- Solução: `width:100%; height:auto; object-fit:cover` e controlar proporção com `aspect-ratio`.

5. Espaçamentos e tipografia inconsistentes:

Vários `padding:100px 2rem;` em seções tornam tudo muito espaçado.

- Solução: Unificar espaçamentos com variáveis CSS e usar `clamp()`.

Plano de ação para deixar o site 100% responsivo:

1. Refatoração inicial:

- Criar variáveis CSS no `:root`.

- Reset global com `box-sizing:border-box`.

2. Abordagem mobile-first:

- Layout de coluna única por padrão, colunas extras em breakpoints.

3. Layout fluido:

- Containers com `width:90%; max-width:1200px; margin:0 auto`.

- Grades com `grid-template-columns: repeat(auto-fit, minmax(200px, 1fr))`.

4. Tipografia adaptável:

- `html { font-size: clamp(14px, 2vw, 18px); }`

- Títulos com `font-size: clamp(1.5rem, 4vw, 3rem);`.

5. Imagens e vídeos responsivos:

- `max-width:100%; height:auto; aspect-ratio`.

6. Menus e navegação:

- Ocultar `.nav-menu` em mobile e exibir via `.nav-menu.active`.

- Botão hamburger com área clicável mínima de 44px.

7. Testes em vários dispositivos:

- Empregar Chrome DevTools e BrowserStack.

Exemplo de CSS refatorado para o hero:

```
:root {
  --spacing: clamp(1rem, 2vw, 3rem);
  --container-max: 1200px;
}
```

```
* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}
```

```
body {
  font-family: 'Helvetica Neue', sans-serif;
  background: #0e0e0e;
  color: #fff;
}

.container {
  width: 90%;
  max-width: var(--container-max);
  margin: 0 auto;
}

.hero {
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  padding: var(--spacing) 0;
  background: linear-gradient(to bottom, #0e0e0e, #1a1a1a);
}

.hero-grid {
  display: grid;
  grid-template-columns: 1fr;
  gap: var(--spacing);
  width: 100%;
}

@media (min-width: 768px) {
  .hero-grid {
    grid-template-columns: repeat(2, 1fr);
  }
}
```