

# OneVision – Guia de Estilo e Padrões (HTML/CSS + Angular)

Este guia é para quem vai tocar as tarefas L1 de frontend no OneVision. A ideia é seguir um **kit de padrões** já pronto para não reinventar CSS a cada página.

## 1) Visão geral de como estilizamos

- **Design tokens** (cores, tipografia, espaçamentos...) vivem em `src/app/shared/styles/onevision-base.css` como variáveis CSS (`--ov-*`).
- **Componentes de layout reutilizáveis** (card, header, filtros, tabela, cards mobile, botões, modal, paginação) também estão no `onevision-base.css` com classes prefixadas por `ov-`.
- **Temas:** Dark é padrão. Light usa `:root[data-theme="light"]` com overrides. O toggle altera o `data-theme` na `<html>`.
- **Angular:** Templates usam **classes globais** `ov-*`. Não duplicar estilos de base nos componentes; se precisar de ajuste local, crie regras **mínimas** no CSS do componente.

## 2) Tokens principais (cores atuais)

Atualizados para casar com o dashboard.

- Fundo: `--ov-bg: #1a1d23`
- Superfícies: `--ov-surface: #1a1d23`, `--ov-surface-raised: #2d3748`, `--ov-surface-overlay: #1f2937`
- Texto: `--ov-text: #e2e8f0`, `--ov-text-secondary: #9ca3af`
- Bordas: `--ov-border: #374151`, `--ov-border-strong: #4b5563`
- Acento: `--ov-accent: #3b82f6`, `--ov-accent-hover: #2563eb`
- Info (gráficos): `--ov-info: #06b6d4`
- Botões subtle: `--ov-btn-bg-subtle: #2d3748`, `--ov-btn-bg-subtle-hover: #374151`

**Regra de ouro:** sempre usar `var(--ov-...)` em vez de hex nos estilos locais.

## 3) Estrutura base de página (Angular template)

Use este esqueleto em novas páginas de recurso:

```
<article class="ov-container" aria-labelledby="page-title">
  <div class="ov-section">
    <div class="ov-header ov-bleed">
      <h1 id="page-title" class="ov-title">Título da Página</h1>

      <!-- Linha 1: Filtros -->
      <div class="ov-filterbar">
```

```

<div class="ov-filter">
  <label for="acc">Account</label>
  <select id="acc" [(change)="filterByAccount($event)]">
    <option value="">All</option>
    <option *ngFor="let acc of uniqueAccounts"
[value]="acc">{{ acc }}</option>
  </select>
</div>

<!-- Repita .ov-filter conforme necessário -->
</div>

<!-- Linha 2: Busca + Ações -->
<div class="ov-header-actions">
  <div class="ov-search">
    <input id="search" type="search" placeholder="Search..."
(input)="onSearch($event)" #searchInput />
    <button type="button" class="ov-clear-btn" *ngIf="searchTerm"
(click)="clearSearch(searchInput)" aria-label="Clear"></button>
  </div>
  <div class="ov-actions">
    <button class="ov-btn ov-btn--subtle" [(click)="exportToCSV()]">Export
CSV</button>
    <button class="ov-btn ov-btn--subtle" [(click)="exportToXLSX()]">Export
XLSX</button>
    <button class="ov-btn ov-btn--subtle"
(click)="resetFilters()">Reset</button>
    <button class="ov-btn ov-btn--subtle"
(click)="refresh()">Refresh</button>
    <button class="ov-btn ov-btn--subtle"
(click)="openColumnCustomizer()">Columns</button>
  </div>
</div>
</div>
</div>

<!-- Tabela desktop -->
<section class="ov-tablewrap"
*ngIf="!loading && filteredResources.length > 0">
  <table class="ov-table">
    <thead>...</thead>
    <tbody>...</tbody>
  </table>
</section>

<!-- Cards mobile -->
<div class="ov-cards" *ngIf="!loading && filteredResources.length > 0">
  <article class="ov-card" *ngFor="let r of paginatedResources"> ... </
article>
</div>

<!-- Estados -->

```

```

<div *ngIf="loading" class="ov-loading">Loading...</div>
<div *ngIf="!loading && filteredResources.length === 0" class="ov-empty">No
items.</div>

<!-- Paginação -->
<div class="ov-pagination" *ngIf="!loading && filteredResources.length >
0"> ... </div>
</article>

```

**Por que** `\*\*\*` Para o header-card alinhar a largura com a tabela, cancelando o padding lateral de .ov-section`.

## 4) Padrões de layout

- **Header card:** .ov-header → card com padding, borda e sombra.
- **Filtros:** .ov-filterbar usa grid com 6 colunas (quebra para 3/2 em breakpoints). Cada filtro usa .ov-filter (label + select estilizados).
- **Linha de busca + ações:** .ov-header-actions é um grid minmax(360px, 1fr) auto. Em ≤ 1200px empilha (busca em cima, botões embaixo alinhados à direita).
- **Tabela desktop:** .ov-tablewrap + .ov-table (fixed layout, células quebram linhas longas).
- **Cards mobile:** .ov-cards aparece ≤ 900px; .ov-tablewrap some nesse breakpoint.
- **Paginação:** .ov-pagination pronto com botões e estado ativo.

## 5) Boas práticas (L1)

- **Não usar inline styles:** prefira classes utilitárias já existentes.
- **Acessibilidade:** sempre aria-label / aria-labelledby em inputs e tabelas; foco visível já é coberto pelo base.
- **Consistência:** use ov-btn e modificadores (--primary, --subtle) em vez de novos botões.
- **Sem** `: se precisar de ajuste pontual, crie um wrapper .minha-pagina e selecione .minha-pagina.ov-\* no CSS do componente.
- **Temas:** se precisar de cor, use tokens var(--ov-...). Nunca hardcode.

## 6) Receitas rápidas

### (a) Adicionar busca sem “quadradozinho”

- Garanta a estrutura .ov-header-actions.
- A busca deve ter min-width via base. Se precisar, ajuste local:

```
.ov-header-actions { grid-template-columns: minmax(360px, 1fr) auto; }
```

## (b) Export CSV/XLSX

- Use `ExportService` já injetado e exporte **apenas colunas visíveis**:

```
exportToXLSX(): void {
  if (!this.filteredResources.length) return;
  const filename = 'page-data.xlsx';
  const visible = this.getVisibleColumns();
  const cols = visible.map(col => ({
    key: col.key,
    label: col.label,
    transform: col.transform || ((row: any) => row[col.key] ?? '')
  }));
  this.exportService.exportDataToXLSX(this.filteredResources, cols,
  filename);
}
```

## (c) Modal padrão

- Use `.ov-modal`, `.ov-modal__dialog`, `.ov-modal__head`, `.ov-modal__body`, `.ov-modal__foot` conforme exemplos de EC2/AMI.

## (d) Status pills

- Para estados: `status-running`, `status-stopped`, `status-pending`, `status-terminated` (ou `available`).

---

## 7) Quebras de layout (breakpoints)

- Filtros:  $6 \rightarrow 3 (\leq 1200\text{px}) \rightarrow 2 (\leq 700\text{px})$
- Header-actions empilha em  $\leq 1200\text{px}$ .
- Tabela some e Cards aparecem em  $\leq 900\text{px}$ .

Verifiquem sempre **1200/900/700** px nas PRs.

---

## 8) Como funciona o tema (Angular)

- Toggle altera `data-theme` no documento:

```
setTheme(isLight: boolean) {
  document.documentElement.setAttribute('data-theme', isLight ?
  'light' : 'dark');
}
```

- O CSS troca tokens no `:root[data-theme="light"]`.

## 9) Onde mexer em estilos

- **Tokens e componentes globais** → `onevision-base.css`.
  - **Ajustes de página** → CSS do componente da página (mínimos, base-first).
  - **Cores** → sempre via tokens. Se cor global mudar, toda a UI acompanha.
- 

## 10) Checklist de entrega (PR L1)

1. Usou o esqueleto `ov-container` → `ov-section` → `ov-header ov-bleed`.
  2. Filtros com `.ov-filterbar` e `.ov-filter` + `label / select`.
  3. Busca + ações com `.ov-header-actions` (teste  $\leq 1200$ px).
  4. Tabela dentro de `.ov-tablewrap` e cards mobile em `.ov-cards`.
  5. Paginação com `.ov-pagination`.
  6. Export CSV/XLSX só com **colunas visíveis**.
  7. Acessibilidade: `aria-*` nos elementos-chave.
  8. Sem CSS duplicado/inline; sem `:ng-deep`.
  9. Teste 1200/900/700 px.
- 

## 11) Exemplos vivos no repo

- **EC2 Instances**: página completa com filtros, busca, export, modal, tabela/cards.
- **AMI Snapshots**: mesmo padrão aplicado.

Se seguirem este guia, novas telas ficam consistentes, responsivas e fáceis de manter. Qualquer dúvida, abrir PR draft com screenshot nos três breakpoints e pingar o revisor. 🚀