

Aula 5

CSS no desktop

Unidade

**Página web: utilizando a responsividade
em aplicações com HTML e CSS**

Parte 2

O que vamos aprender?



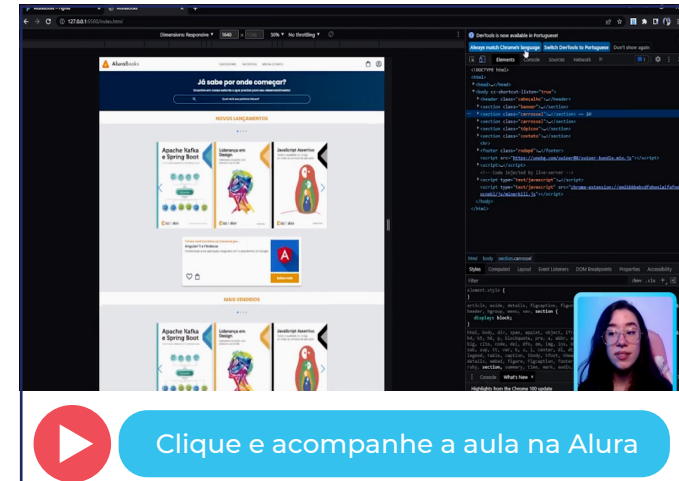
Estilizar o header para o layout Desktop.



Refatorar o CSS.



Utilizar as Media Queries.



Evoluindo o header

Nosso objetivo é que o site funcione corretamente tanto na versão mobile, quanto tablet e desktop. Vamos começar pelo header. Por isso, iniciaremos a estilização para o desktop. Podemos observar, no Figma, que o menu de *Categorias*, *Favoritos* e *Estante* agora ficam à esquerda, enquanto *Sacola* e *Perfil* ficam à direita, e ao lado dos seus respectivos ícones. Além disso, o tamanho de tela que usaremos será de 1728 pixels.



Para adicionarmos o texto ao lado dos ícones, no arquivo “index.html”, vamos criar uma tag **<p>** após as imagens. Para que possamos manipular o CSS, vamos acrescentar uma classe ao componente “pai”, a tag **<a>**, chamada **container__link**.

```
<div class="container">
  <a href="#">
    
  </a>
  <a href="#" class="container__link">
    
    <p class="container__texto">Minha sacola</p>
  </a>
  <a href="#" class="container__link">
    
    <p class="container__texto">Meu perfil</p>
  </a>
</div>
```

No arquivo “header.css”, vamos criar uma Media Query e transformar o componente em “flex item” com a propriedade **display**, para que assim possamos alinhar ao centro com a propriedade **align-items**. Também vamos remover a decoração padrão da âncora com a propriedade **text-decoration**, e atribuir a cor preta ao texto.

```
@media screen and (min-width: 1728px) {  
  .container__link {  
    display: flex;  
    align-items: center;  
    text-decoration: none;  
    color: var(--preto);  
  }  
}
```

Vamos atribuir uma margem interna lateral a todo o header por meio da propriedade **padding**, para que os componentes não fiquem muito grudados às extremidades.

```
.cabeçalho {  
  padding: 0 2em;  
}
```

Com a propriedade **margin**, vamos alinhar as opções *Categorias*, *Favoritos* e *Estante* à direita.

```
.opcoes {  
    margin-right: auto;  
}
```

Por fim, para ocultar os textos nas resoluções menores, vamos atribuir o **display: none** antes das Media Queries.

```
.container_texto  
    display: none;  
}
```

Atribuímos o **display: block** dentro da Media Query para fazer os textos aparecerem na resolução de desktop.

```
.container__texto {  
  display: block;  
}
```

Com essas modificações, o header deve ficar com a aparência da imagem a seguir.



Desafio

Nessa aula, aplicamos a responsividade com Media Queries no header. Utilizamos a propriedade `display` para ocultar ou exibir um componente na tela. O desafio será pesquisar qual é a diferença de se utilizar propriedade **`display: none`** e da propriedade **`visibility: hidden`**.

Avalie este material acessando o link: <https://forms.gle/EcEZdj59zGS9QTWY9>