

Aula 10

Integrando o carrossel com swiper JS

Curso

**HTML e CSS: responsividade
com mobile-first**

Cards

Dando sequência ao projeto da AluraBooks, podemos observar no Figma o próximo elemento a ser desenvolvido. Temos um cartão que nos oferece uma recomendação. Vamos baixar a imagem do *Angular* em formato *svg* e renomeá-lo como *Angular*. Não precisamos baixar as imagens dos ícones pois são as mesmas que utilizamos no header.



Antes do fechamento da tag **<section>** referente ao carrossel, vamos criar uma sequência de tags **<div>** para estruturar o card, dentro de *index.html*.

```
<div class="card">  
  <div class="card_descricao">  
    <div class="descricao">  
  
    </div>  
  </div>  
</div>  
</section>
```

Podemos utilizar as tags **<h3>**, **<h2>** e **<p>** para criar as descrições de título e texto, respectivamente.

```
<div class="descricao">

  <h3 class="descricao_titulo">Talvez você também se interesse por...</h3>

  <h2 class="descricao_titulo-livro">Angular 11 e Firebase</h2>

  <p class="descricao_texto">

    Construindo uma aplicação integrada com a plataforma do Google.

  </p>

</div>
```

Vamos acrescentar a imagem do *Angular* após o fechamento da tag **<div>** referente à descrição.

```
<div class="descricao">  
  
  <h3 class="descricao__titulo">Talvez você também se interesse por...</h3>  
  
  <h2 class="descricao__titulo-livro">Angular 11 e Firebase</h2>  
  
  <p class="descricao_texto">Construindo uma aplicação integrada com a  
plataforma do Google.</p>  
  
</div>  
  

```

Agora temos toda a estrutura da nossa primeira linha, dividida em duas colunas, conforme os comentários no código.

```
<div class="card">
  <!-- Primeira linha -->
  <div class="card_descricao">
    <!-- Primeira coluna -->
    <div class="descricao">
      <h3 class="descricao_titulo">
        Talvez você também se interesse por...
      </h3>
      <h2 class="descricao_titulo-livro">
        Angular 11 e Firebase
      </h2>
      <p class="descricao_texto">
        Construindo uma aplicação integrada com a plataforma do Google.
      </p>
    </div>
    <!-- Segunda coluna -->
    
  </div>
</div>
```

Ao atribuir à classe **.card_descricao** um **display: flex**, todos os elementos se tornam flexbox e automaticamente separam-se em duas colunas. Vamos também atribuir um espaçamento entre os elementos com **justify-content**, além de uma margem na parte inferior do texto, no arquivo *carrossel.css*.

```
.card_descricao {  
    display: flex;  
    justify-content: space-between;  
    margin-bottom: 0.5em;  
}
```

O resultado esperado é o seguinte:



Seguindo a mesma estrutura, vamos agora iniciar a segunda linha do nosso card. A primeira coluna será composta por uma lista não ordenada, e colocaremos as imagens de favoritos e da sacola de compras em cada uma das linhas.

```
<!-- Segunda coluna -->

</div>

<!-- Segunda linha -->
<div class="card_botoes">
  <!-- Primeira coluna -->
  <ul class="botoes">
    <li class="botoes_item"></li>
    <li class="botoes_item"></li>
  </ul>
</div>
```

Logo abaixo da lista, vamos inserir uma tag **<a>** para criar o botão de *Saiba mais*, sendo esta a nossa segunda coluna da segunda linha.

```
<!-- Primeira coluna -->
<ul class="botoes">
  <li class="botoes_item">
    
  </li>
  <li class="botoes_item">
    
  </li>
</ul>
<!-- Segunda coluna -->
<a href="#" class="botoes__ancora"> Saiba mais </a>
```

Podemos transformar a segunda linha em um flexbox repetindo o que fizemos anteriormente no CSS. Assim, o conteúdo já ficará disposto em duas colunas.

```
.card_botoes {  
  display: flex;  
  justify-content: space-between;  
}
```

Para que os ícones fiquem na mesma linha, basta transformá-los em flexbox.

```
.botoes {  
  display: flex;  
}
```

O resultado da página após as alterações anteriores deve ficar assim:



Agora que a estrutura do card já está completa, vamos partir para a estilização. Vamos nos basear no Figma, utilizando a aba *Inspect* para verificar os elementos CSS e incorporá-los ao nosso projeto.



Vamos atribuir uma cor de fundo, sombreamento e arredondamento de bordas ao nosso card. Para que o card não fique encostado nas bordas da tela, nem que seu conteúdo fique encostado nas bordas, podemos adicionar uma margem externa e interna a ele utilizando **margin** e **padding**, respectivamente, no arquivo *carrossel.css*.

```
.card {  
  background: var(--branco);  
  box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);  
  border-radius: 10px;  
  margin: 1em; padding:  
    1em;  
}
```

Partindo para o texto, iniciaremos com o primeiro título, atribuindo cor e peso da fonte.

```
.descricao_titulo {  
  color: var(--laranja);  
  font-weight: 700;  
}
```

O próximo título contém uma cor azul que não utilizamos anteriormente em nosso código. Vamos criar uma variável dentro do **:root** no arquivo *styles.css* para representá-la.

```
:root {  
  --cor-de-fundo: #EBECEEE;  
  --branco: #FFFFFF;  
  --laranja: #EB9B00;  
  --azul: #002f52;  
}
```

Vamos definir cor, tamanho e peso de fonte ao título do livro, além de uma margem para o espaçamento, no arquivo *carrossel.css*.

```
.descricao_titulo-livro {  
  color: var(--azul);  
  font-size: 18px;  
  font-weight: 700;  
  margin: 0.5em 0;  
}
```

Por fim, vamos definir o tamanho de fonte da descrição do livro.

```
.descricao_texto {  
  font-size: 14px;  
}
```


O resultado deve ser tal como o da imagem a seguir.



Para separar os ícones que estão grudados, vamos definir uma margem lateral no CSS.

```
.botoes_item {  
    margin: 0 0.5em;  
}
```

Por fim, para estilizar a âncora e torná-la parecida com um botão, vamos definir cor de fundo, margem interna, cor e peso da fonte. Também temos que remover a decoração padrão da tag <a> atribuindo **text-decoration: none;**.

```
.botoes__ancora {  
    background-color: var(--laranja);  
    padding: 1em 2.2em;  
    color: var(--branco);  
    font-weight: 700;  
    text-decoration: none;  
}
```

O resultado deve ser este:

