

Aula 2

# Ajustes de tela

Unidade

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

**Parte 2**

# O que vamos aprender?



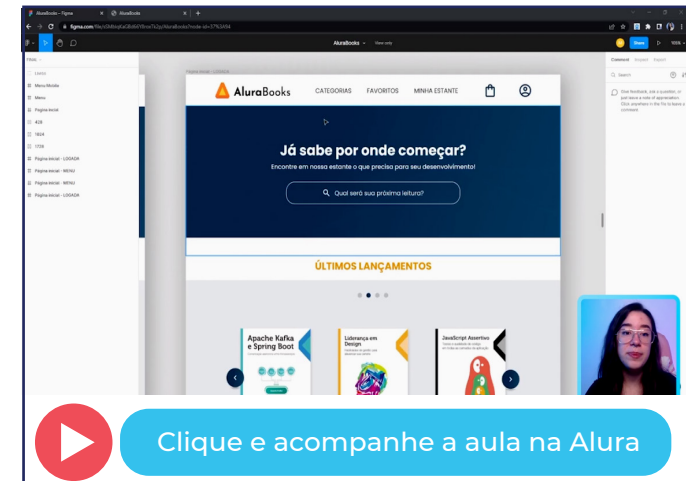
Aplicar as Media Queries.



Criar o layout responsivo do header.



Adaptar a lista de categorias.



# Aumentando o Header

Agora vamos começar a desenvolver o header adaptado para a versão tablet. Observe no Figma: podemos perceber os elementos novos e os que devem sofrer alguma alteração quando utilizado nesse tamanho de tela. Agora temos um título *Alura Books* ao lado da logomarca, além das opções *Categorias*, *Favoritos* e *Minha estante*.

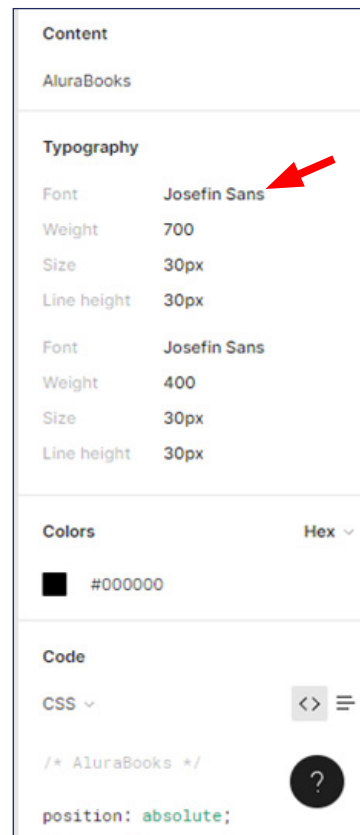


Dentro da tag **<header>**, após a tag **<img>** com o logotipo da AluraBooks, vamos inserir uma tag **<h1>** para criar o título.

```

<h1 class="container__titulo"><b class="container__titulo--negrito">
Alura</b>Books</h1>
```

Podemos observar no Figma, no bloco *Typography*, que o título do header possui uma fonte chamada *Josefin Sans*.



Vamos importar a fonte *Josefin Sans* do Google Fonts, da mesma forma como fizemos anteriormente. Como essa etapa já foi feita em outros momentos deste curso, fica como exercício fazermos sozinhos ou com a ajuda de um colega. Relembrando que devemos acessar o Google Fonts e pesquisar pela fonte desejada. Selecione os pesos clicando no local indicado pela seta (na imagem a seguir).



Em seguida, copie o código dado na última tag **<link>**, já que as duas primeiras servem para criar a conexão do Google Fonts com nosso projeto e já as temos na tag **<head>**.

To embed a font, copy the code into the  
<head> of your html

☒ <link> ☐ @import

```
<link rel="preconnect" href="https://  
fonts.googleapis.com">  
<link rel="preconnect" href="https://  
fonts.gstatic.com" crossorigin>  
<link href="https://fonts.googleapis.  
com/css2?family=Josefin+Sans:wght@40  
0;700&display=swap" rel="stylesheet">
```

CSS rules to specify families

```
font-family: 'Josefin Sans', sans-ser  
if;
```

[Read our FAQs](#)

[API docs](#) [Download all](#)

Em seguida, acrescentaremos a tag copiada anteriormente no arquivo *index.html*, abaixo da tag **<link>** que importa a fonte Poppins, dentro da tag **<head>**.

```
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;700&display=swap" rel="stylesheet">  
<link href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@400;700&display=swap" rel="stylesheet">
```

No arquivo *styles.css*, vamos criar no **:root** uma nova variável nomeada como **--fonte-secundária: "Josefin Sans"**; para nossa nova fonte.

```
:root {  
  --cor-de-fundo: #EBECEEE;  
  --branco: #FFFFFF;  
  --laranja: #EB9B00;  
  --azul-degrade: linear-gradient(97.54deg,  
#002F52 35.49%, #326589 165.37%);  
  --fonte-principal: "Poppins";  
  --azul: #002f52;  
  --fonte-secundaria: "Josefin Sans";  
}
```



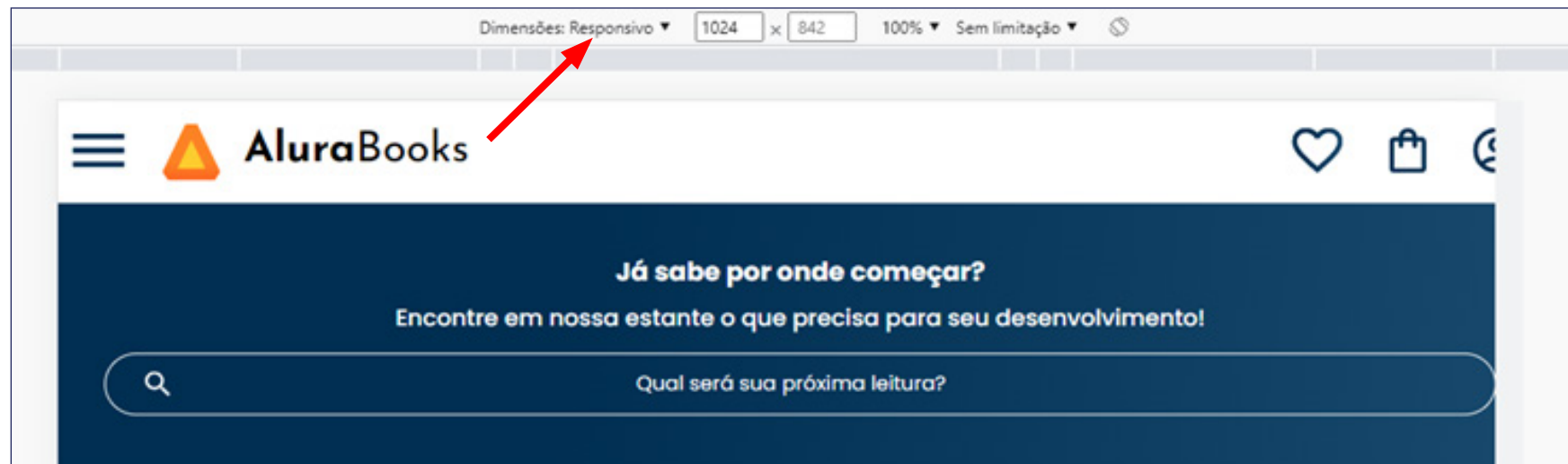
No arquivo *header.css*, dentro de **@media**, vamos começar a criar a estilização para os dispositivos tablets. Primeiramente, definiremos a fonte, em **font-family** e seu tamanho, em **font-size**, tanto para o título quanto para sua parte em negrito.

```
@media screen and (min-width: 1024px) {  
  .container__titulo,  
  .container__titulo--negrito {  
    font-family: var(--fonte-secundario);  
    font-size: 30px;  
  }  
}
```

Em seguida, definiremos separadamente seus pesos.

```
.container__titulo {  
  font-weight: 400;  
}  
  
.container__titulo--negrito {  
  font-weight: 700;  
}
```

Para visualizarmos as alterações no navegador, precisamos alterar as dimensões para *Responsivo* e definir o primeiro parâmetro referente à largura como 1024 pixels.



Para que o título fique visível apenas na resolução de tablet, podemos definir um **display** fora da Media Query para que ele não apareça a princípio.

```
.container_titulo {  
  display: none;  
}
```

Já dentro da Media Query, vamos definir um **display** para que o título apareça quando a tela atingir a largura mínima de 1024 pixels.

```
.container_titulo {  
  font-weight: 400;  
  display: block;  
}
```

Para a lista de opções, vamos utilizar a tag `<ul>` para criar uma lista não ordenada, logo abaixo do título que criamos há pouco, no arquivo *index.html*.

```
<h1 class="container__titulo"><b class="container__titulo--negrito">Alura</b>Books</h1>
</div>

<ul class="opcoes">
  <li class="opcoes__item">Categorias</li>
  <li class="opcoes__item"><a href="#" class="opcoes__link">
Favoritos</a></li>
  <li class="opcoes__item"><a href="#" class="opcoes__link">Minha
estante</a></li>
</ul>
```

Para exibir a lista lado a lado, vamos definir um **display** apropriado na estilização, no arquivo *header.css*, dentro da nossa Media Query.

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

Em seguida, em cada item da lista, vamos aplicar uma margem interna lateral utilizando o **padding** e transformar todas as letras em maiúsculas com **text-transform**.

```
.opcoes__item {  
  padding: 0 1em;  
  text-transform: uppercase;  
}
```

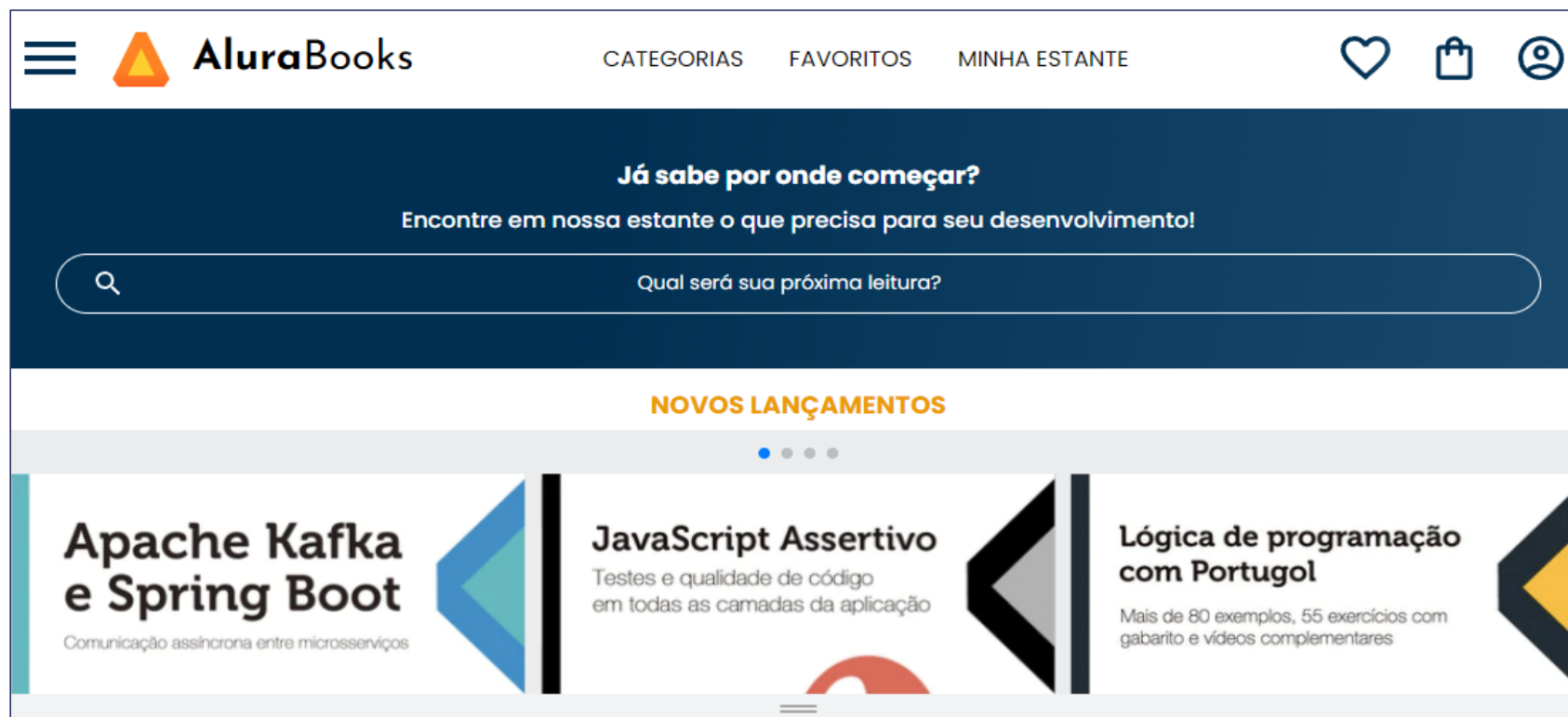
Em *styles.css*, vamos criar uma variável dentro de **:root** para armazenar a cor preta.

```
--preto: #000000;
```

E, retornando ao *header.css*, vamos retirar a decoração padrão da âncora e definir a cor preta para os links da lista.

```
.opcoes__link {  
  text-decoration: none;  
  color: var(--preto);  
}
```

O resultado deve ser o seguinte:



Para que a lista fique visível apenas na resolução para tablet, vamos ocultá-la fora da Media Query utilizando o **display**.

```
.opcoes {  
    display: none;  
}
```

No arquivo *index.html*, vamos definir uma nova classe para o ícone de favoritos, para que possamos ocultá-lo no CSS.

```
<a href="#"></a>
```

De volta ao arquivo *header.css*, vamos ocultar o ícone de favoritos dentro da Media Query utilizando o **display**.

```
.container__imagem-transparente {  
    display: none;  
}
```

Por fim, vamos ocultar o menu hambúrguer com a mesma técnica, utilizando o **display**.

```
.cabeçalho__menu-hamburguer {  
    display: none;  
}
```

O resultado das modificações deve ser o seguinte:





# Lista de categorias

Para que a lista seja exibida, vamos utilizar as mesmas técnicas que usamos na construção do menu hambúrguer. Primeiramente, vamos criar um **checkbox** e associá-lo à opção **Categorias**.

```
<ul class="opcoes">
  <input type="checkbox" id="opcoes-menu" class="opcoes__botao">
  <label for="opcoes-menu">
    <li class="opcoes__item">Categorias</li>
  </label>
```

Vamos adicionar a lista após o fechamento da tag `<label>`.

```
<label for="opcoes-menu">
  <li class="opcoes__item">Categorias</li>
</label>

<ul class="lista-menu">
  <li class="lista-menu__item">
    <a href="#" class="lista-menu__link">Programação</a>
  </li>
  <li class="lista-menu__item">
    <a href="#" class="lista-menu__link">Front-end</a>
  </li>
  <li class="lista-menu__item">
    <a href="#" class="lista-menu__link">Infraestrutura</a>
  </li>
  <li class="lista-menu__item">
    <a href="#" class="lista-menu__link">Business</a>
  </li>
  <li class="lista-menu__item">
    <a href="#" class="lista-menu__link">Design & UX</a>
  </li>
</ul>
```

De volta ao arquivo *header.css*, dentro da Media Query, vamos utilizar o combinador CSS `~` para estilizar a classe **lista-menu** quando o checkbox estiver selecionado.

```
.opcoes__botao:checked~.lista-menu {  
    display: block;  
    width: auto;  
}
```

Por fim, vamos ocultar o checkbox em si utilizando o **display**.

```
.opcoes__botao {  
    display: none;  
}
```

O resultado deve ser como o mostrado a seguir:



# Desafio

Nesta aula, aprendemos a aplicar a responsividade com Media Queries. Para ocultar e exibir elementos na tela, utilizamos a propriedade CSS **display**. O desafio será pesquisar quais são os valores atribuíveis a display.

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