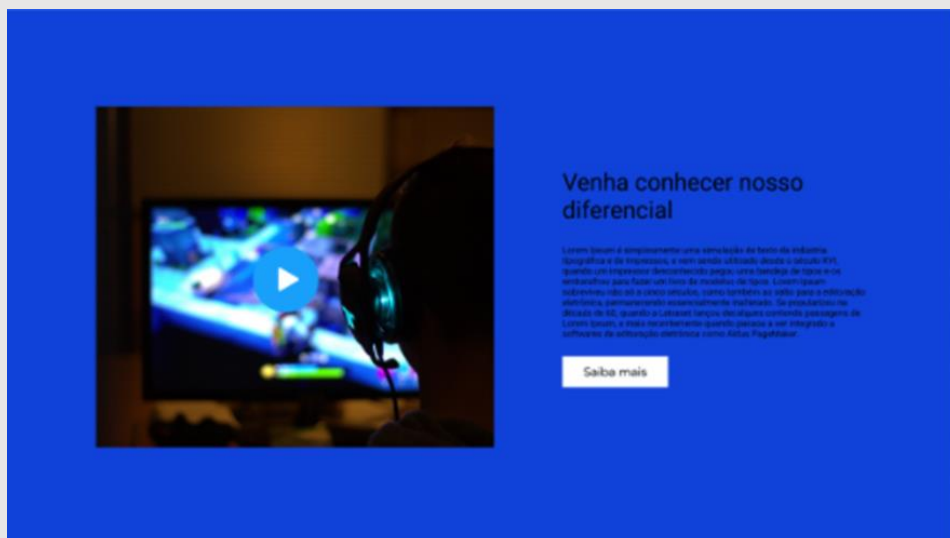


Imagens

Aprenda, agora, a adicionar imagens em sua página.

Inicie criando um banner:



Exemplo de banner – Fonte: Elaborado pela autora.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-12">
      
    </div>
  </div>
</div>
```

Exemplo de tag – Fonte: Elaborado pela autora.

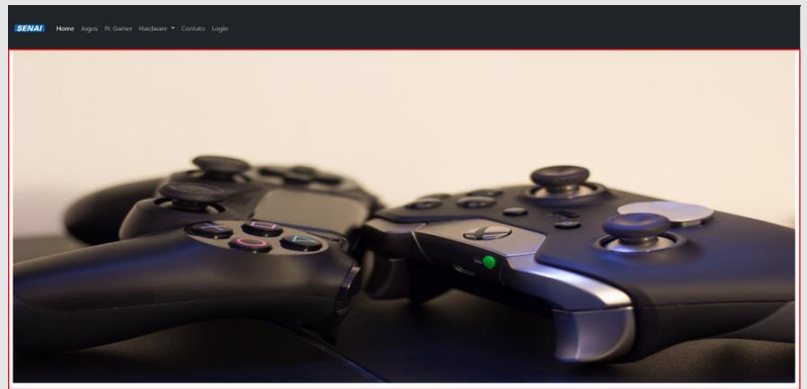
Como boa prática, **toda estilização será feita no arquivo style.css**. As **class** que são do **Bootstrap** não irão sobrescrever o CSS feito, para isso, será preciso **adicionar uma nova class e personalizá-la**.

```
.banner-home{  
  width: 100%;  
  height: 800px;  
}
```

CSS – Fonte: Elaborado pela autora.

E esse é o resultado no navegador.

Nesse caso, a **class .container-fluid** (tela toda) está sendo utilizada, além de a imagem estar **personalizada com um width de 100%**.



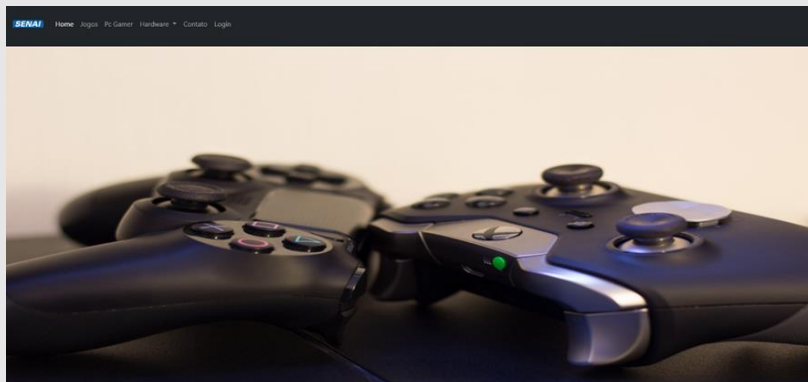
Exemplo no navegador ® – Fonte: Elaborado pela autora.

A **margem presente nas laterais da imagem** é adicionada pela **class .col-12**. Para isso:

```
<div class="container-fluid p-0 m-0">  
  <div class="row m-0">  
    <div class="col-12 p-0">  
        
    </div>  
  </div>  
</div>
```

HTML – Fonte: Elaborado pela autora.

O *Bootstrap* vem com seu CSS3 pronto. Ao adicionar **p-0** e **m-0** na **tag**, você está passando como **valor** **padding:0px** e **margin:0px**.



Exemplo no navegador – Fonte: Elaborado pela autora.

Observe algumas alterações possíveis de serem feitas no site, deixando da seguinte forma:

banner-home: elemento pai.

align-items-center é uma *class* do *bootstrap* que alinha os elementos de modo centralizado.

No ***container-fluid***, é possível adicionar **duas colunas (col-6)** para o conteúdo.

```
<div class="banner-home d-flex align-items-center">
  <div class="container-fluid">
    <div class="row">
      <div class="col-6">
        
      </div>
      <div class="col-6">
        <section class="texto-menu">
          <h1>Venha conhecer nosso diferencial</h1>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.</p>
          <a href="#" target="_blank">Saiba mais</a>
        </section>
      </div>
    </div>
  </div>
</div>
```

HTML – Fonte: Elaborado pela autora.

```
.banner-home{
  width: 100%;
  height: 600px;
  background-color: rgb(69 69 255);
}
.banner-home img{
  width: 60%;
  height: 400px;
  margin: 0 auto;
  display: table;
}
.banner-home .texto-menu{
  margin-top: 30px;
  width: 80%;
}
.banner-home .texto-menu h1{
  font-size: 40px;
  width: 80%;
}
.banner-home .texto-menu p{
  font-size: 18px;
  margin: 30px 0px;
  width: 90%;
}
.banner-home .texto-menu .btn-banner{
  padding: 8px 18px;
  background-color: #fff;
  text-decoration: none;
  color: #000;
}
```

CSS – Fonte: Elaborado pela autora.

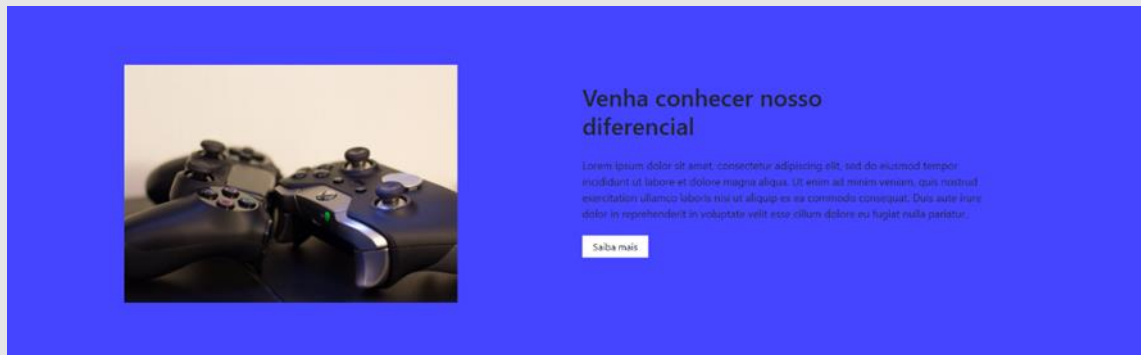
Importante

Ao adicionar **.col-6**, o valor será considerado da maior resolução até a menor, ou seja, para uma resolução de 1920px até 320px, utilize **.col-6**.



Caso deseje promover uma **alteração no layout**, insira **.col-12-sm**, por exemplo, pois, a partir da resolução **576px** até **320px**, o **componente ocupará 100% da coluna**.

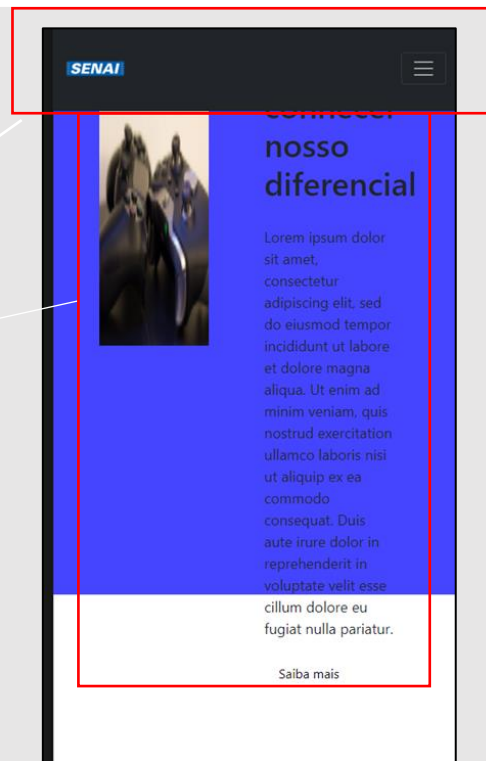
Observe o resultado no navegador:



Resultado no navegador – Fonte: Elaborado pela autora.

Em uma resolução de tamanho grande, o layout fica proporcional, porém, quando a resolução é diminuída, o layout altera-se para:

O menu fica proporcional, mas a *div de banner-home* não.



Resultado no navegador do layout alterado
Fonte: Elaborado pela autora.

Para deixar o layout proporcional, é preciso alterar alguns detalhes no código em HTML5:

```
<div class="banner-home d-flex align-items-center">
  <div class="container-fluid">
    <div class="row">
      <div class="col-12 col-md-6">
        
      </div>
      <div class="col-12 col-md-6">
        <section class="texto-menu">
          <h1>Venha conhecer nosso diferencial</h1>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.</p>
          <a class="btn-banner" href="#" target="_blank">Saiba mais</a>
        </section>
      </div>
    </div>
  </div>
</div>
```

HTML – Fonte: Elaborado pela autora.

Quando **col-md-6** é adicionado, tem-se, que da resolução 768px até a resolução máxima, o layout terá como propriedade .col-6; e, da resolução 320px até 767px, o layout terá .col-12 como propriedade.

Dica!

Outro item de responsividade possível de ser utilizado no site para deixá-lo adaptativo é a utilização da **@media screen**, alterando as propriedades de alguns componentes do código. Observe:

```
@media screen and (max-width: 600px){
  .banner-home img{
    width: 100%;
    height: 300px;
  }

  .banner-home .texto-menu{
    width: 100%;
  }
  .banner-home .texto-menu h1{
    font-size: 22px;
  }
  .banner-home .texto-menu p{
    font-size: 12px;
    margin: 30px 0px;
    width: 100%;
  }
}
```

CSS – Fonte: Elaborado pela autora.

Fazendo essas alterações, o resultado será:



Exemplo no navegador – Fonte: Elaborado pela autora.

Agora, aprenda a estruturar um layout conforme imagem a seguir:



Exemplo de estrutura de layout – Fonte: Elaborado pela autora.

Para realizar tal tarefa, basta fazer:

```
<div class="destaque d-flex align-items-center">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <section class="texto-destaque">
          <h2>Que tal conhecer nossas obras em destaque?</h2>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
        </section>
      </div>
      <div class="col-12 col-md-2">
        
      </div>
      <div class="col-12 col-md-2">
        
      </div>
      <div class="col-12 col-md-2">
        
      </div>
      <div class="col-12 col-md-2">
        
      </div>
      <div class="col-12 col-md-2">
        
      </div>
      <div class="col-12">
        <a href="#">Saiba mais</a>
      </div>
    </div>
  </div>
</div>
```

HTML – Fonte: Elaborado pela autora.

Importante

Lembre-se que `<h1>` é único para cada página e ele irá na parte superior do código.

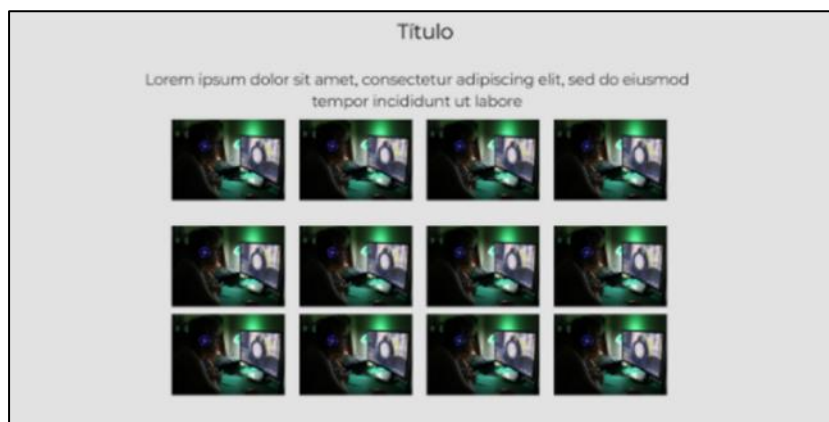


No caso da imagem a seguir, foi adicionado o `<h1>` no banner. Observe o resultado:



Banner com `<h1>` – Fonte: Elaborado pela autora.

A imagem abaixo mostra outra opção de layout:



Outra opção de layout – Fonte: Elaborado pela autora.

Para obter essa estrutura, faça:

HTML – Fonte: Elaborado pela autora.

Observe o resultado:



Exemplo do componente grid – Fonte: Elaborado pela autora.

Carousel

O recurso chamado *carousel* (carrossel) é uma outra maneira de exibir imagens; nesse recurso, uma imagem é deixada em destaque e, por meio de setas ou automaticamente, outras imagens assumem sua posição. Esse recurso é muito utilizado em sites de notícias e de entretenimento.

Dica!

É possível criar um *carousel* com a estrutura do *Bootstrap* e fazer alterações de elementos.



Para isso, acesse

<https://getbootstrap.com/docs/5.0/components/carousel/>; nesse site, alguns layouts de carousel são apresentados.

Escolha e copie um código; depois, adicione-o ao HTML5.

```
<div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Anterior</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Próximo</span>
  </button>
</div>
```

HTML – Fonte: Elaborado pela autora.

Devido ao tamanho da imagem ser grande, uma estilização foi adicionada. Observe:

```
#carouselExampleControls{
  width: 80%;
  margin: 0 auto;
  display: table;
}

#carouselExampleControls img{
  width: 60%;
  height: 600px;
}
```

CSS – Fonte: Elaborado pela autora.

Agora o *carousel* está pronto...

Porém, como ele irá aparecer em uma resolução para mobile, é preciso deixá-lo oculto. Para isso, **adicione a *class* de display** (<https://getbootstrap.com/docs/5.0/utilities/display/>) que a própria documentação do *Bootstrap* proporciona.

Escondendo elementos

Para um desenvolvimento mais rápido e compatível com dispositivos móveis, use classes de exibição responsivas para mostrar e ocultar elementos por dispositivo. Evite criar versões totalmente diferentes do mesmo site, em vez disso, oculte elementos responsivamente para cada tamanho de tela.

Para ocultar elementos, basta usar a `.d-none` classe ou uma das `.d-{sm,md,lg,xl,xxl}-none` classes para qualquer variação de tela responsiva.

Para mostrar um elemento apenas em um determinado intervalo de tamanhos de tela, você pode combinar uma `.d-*-none` classe com uma `.d-*-*` classe, por exemplo, `.d-none .d-md-block`, `.d-xl-none`, `.d-xxl-none` irá ocultar o elemento para todos os tamanhos de tela, exceto em dispositivos médios e grandes.

Tamanho da tela	Classe
Escondido em tudo	<code>.d-none</code>
Oculto apenas em xs	<code>.d-none .d-sm-block</code>
Oculto apenas em sm	<code>.d-sm-none .d-md-block</code>
Oculto apenas no MD	<code>.d-md-none .d-lg-block</code>
Oculto apenas em lg	<code>.d-lg-none .d-xl-block</code>
Oculto apenas em xl	<code>.d-xl-none .d-xxl-block</code>
Oculto apenas em xxl	<code>.d-xxl-none</code>
Visível em todos	<code>.d-block</code>
Visível apenas em xs	<code>.d-block .d-sm-none</code>
Visível apenas em sm	<code>.d-none .d-sm-block .d-md-none</code>
Visível apenas no MD	<code>.d-none .d-md-block .d-lg-none</code>
Visível apenas em lg	<code>.d-none .d-lg-block .d-xl-none</code>
Visível apenas em xl	<code>.d-none .d-xl-block .d-xxl-none</code>
Visível apenas em xxl	<code>.d-none .d-xxl-block</code>

Display – *Bootstrap* – Fonte: <https://getbootstrap.com/docs/5.0/utilities/display/>.

No mobile, o *carousel* irá receber, a princípio, o valor **d.none** (display:none, ou seja, oculto) e aparecerá no dispositivo tablet (768px, ou seja, tamanho md). A galeria ficará no dispositivo de md até xxlg.

Dessa forma, acrescente no código:

```
<div class="galeria d-flex align-items-center d-none d-lg-block">
```

HTML – Fonte: Elaborado pela autora.

```
<div id="carouselExampleControls" class="carousel slide d-block d-lg-none" data-bs-ride="carousel">
```

HTML – Fonte: Elaborado pela autora.

Agora é com você! Faça o teste em seu projeto.