

# BOOTSTRAP



**Por: Profa. Adriana Belon**

**[www.adrianabelon.pro.br](http://www.adrianabelon.pro.br)**

---

## Instruções para um satisfatório aproveitamento da Obra

---

A Semiótica é uma área que estuda os processos de significação e os sistemas de comunicação, bem como os processos envolvidos na produção e interpretação de signos.

A interface (no caso da obra, as páginas) é vista como uma mensagem sendo enviada pela autora ao leitor.

Desta forma, foram adicionados nesta obra, elementos de semiótica, que induzem o usuário a uma melhor interpretação dos conteúdos apresentados, evidenciando tópicos importantes, explicações e momentos de prática.

Veja abaixo os elementos e seu significado, e atente-se a eles no decorrer do livro.

 <i>Explicações</i>	 <i>Muito Importante</i>
 <i>Exercícios e prática</i>	 <i>Projetos</i>
 <i>Maior Nível de Dificuldade</i>	

# BOOTSTRAP

## 1.1 Design Responsivo

---

Com o advento móvel (celulares, smartphones, tablets), tendo assim telas com diferentes padrões e tamanhos, além da quantidade e diversidade dos navegadores, o design responsivo ganhou notoriedade. No passado, grande parte dos sites eram acessados por navegadores muito semelhantes.

Até pouco tempo atrás bastava fazer um site que funcionava em Internet Explorer com resolução máxima de 1024×768 pixels que tudo estava resolvido.

Hoje tudo mudou, temos TVs de 50" polegadas acessando internet, temos celulares que tem telas de 2" até 5" , os tablets~ de 5" até 7" . , tablets de 6" até 14" polegadas (ou até mais) e sem contar os próprios computadores, que tem telas de netbook até os iMacs da Apple, colocando a margem de 11" até mais de 26" polegadas.

Design responsivo é uma abordagem de design web que visa criar páginas da web que oferecem uma experiência de usuário ideal em diferentes dispositivos e tamanhos de tela. Isso é alcançado por meio de uma combinação de técnicas flexíveis de layout, imagens e media queries CSS.

A ideia por trás do design responsivo é garantir que os sites se adaptem automaticamente ao ambiente de exibição do usuário, seja em um computador desktop, laptop, tablet ou smartphone. Em vez de criar diferentes versões do mesmo site para diferentes dispositivos, o design responsivo permite que o layout e o conteúdo se ajustem dinamicamente com base no tamanho da tela e nas capacidades do dispositivo usado para acessar o site.



**Desafio do Designer** → Projetar interfaces que funcionem em qualquer dispositivo com a mesma qualidade.

---

“O design responsivo, como o próprio nome já indica, consegue responder ao tamanho da tela para se adequar da melhor forma. Ao invés de criar dois sites separados, um para mobile (smartphones e tablets) e um paradesktops, como também é possível fazer, você faz apenas um site que vai se adaptar muito bem a qualquer tela em que ele for carregado.”

Para desenvolver de forma responsiva, você pode utilizar frameworks para facilitar seu trabalho. Frameworks são estruturas de código já prontas que trazem consigo diversas linhas em CSS, JavaScript e HTML já escritas, facilitando bastante o trabalho do desenvolvedor do site.

Entre os frameworks mais conhecidos podemos citar os seguintes:

- **Bootstrap**

O Bootstrap é o mais popular frameworks para desenvolvimento responsivo. Ele foi criado pela equipe do Twitter e atualmente é utilizado por uma quantidade imensa de sites, e-commerce e sistemas. Particularmente, é o mais vantajoso dessa lista, mas é importante que você considere bem os seus objetivos para escolher qual framework utilizará.

- **Wirefy**

Por ser um pouco mais simples, conta com uma curva de aprendizagem menor, porém é menos popular. Ele tem paginação, formulários, slideshow, divisão em grid, além de outras funcionalidades.

- **Skeleton**

Ele é limpo, contém somente o que é básico e necessário: formulários, botões, tipografia e grid. Sua curva de aprendizado é baixíssima. Ideal para quem quer desenvolver funcionalidades/aplicações.

- **Less Framework**

Não conta com nenhum complemento, sendo ideal para quem desenvolve designs e layouts únicos e quer apenas um framework que ajude a torna-los responsivos. Por ser o mais simples dos exemplos aqui citados e trabalhar apenas com grid responsivo, ele é super leve.

- **Foundation**

Segundo eles é “o framework mais avançado do mundo” e até é possível acreditar. Com ele, você consegue desenvolver mobile first e ir passando para telas maiores ou vice-versa, além de conseguir trabalhar com a prototipagem do site de forma rápida.



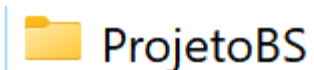
**BootStrap** → utilizaremos o bootstrap por sua riqueza de recursos, sua velocidade, compatibilidade, responsividade e por ser open source

---

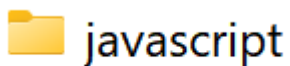
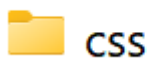
## 1.2 Instalando e utilizando o BootStrap

---

**1ª – Crie uma pasta para seu 1º. Projeto**



**2ª – Crie a pasta para colocar o JQuery e o CSS**



**3º. Baixe o JQuery e coloque na pasta javascript**

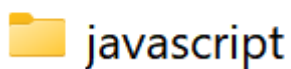
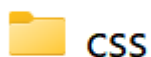
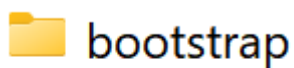
Acessar o site oficial da JQuery; (<https://jquery.com/download/>)

Clicar em “download JQuery”;

**4º. Acesse o site do bootstrap**

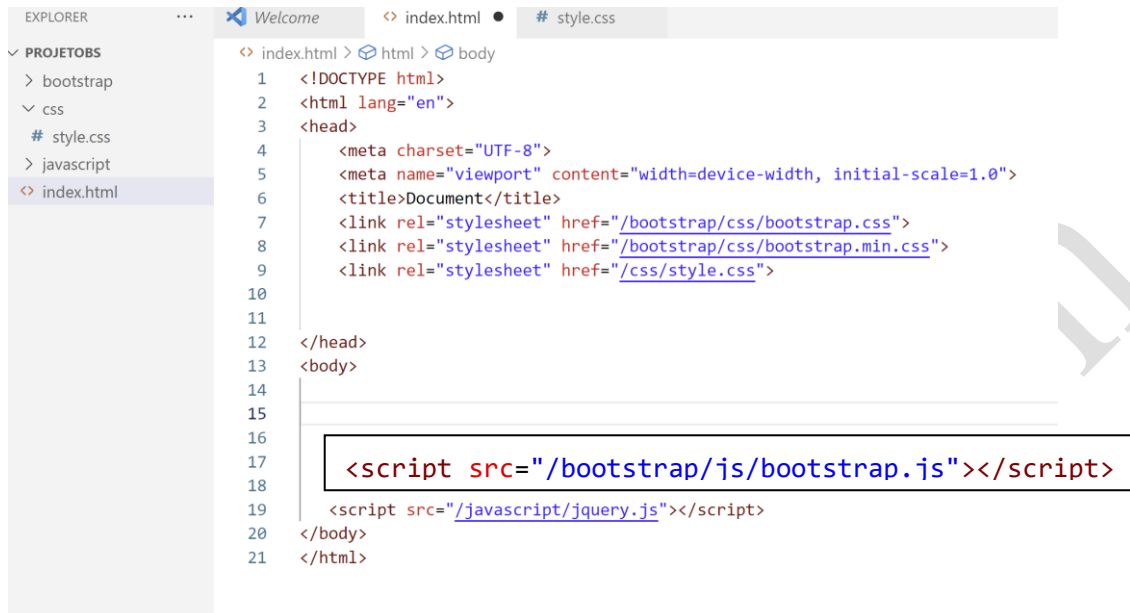
<https://getbootstrap.com.br/>

- a) Faça o download do bootstrap
- b) Descompacte BootStrap na pasta de seu projeto e renomeie a pasta para bootstrap



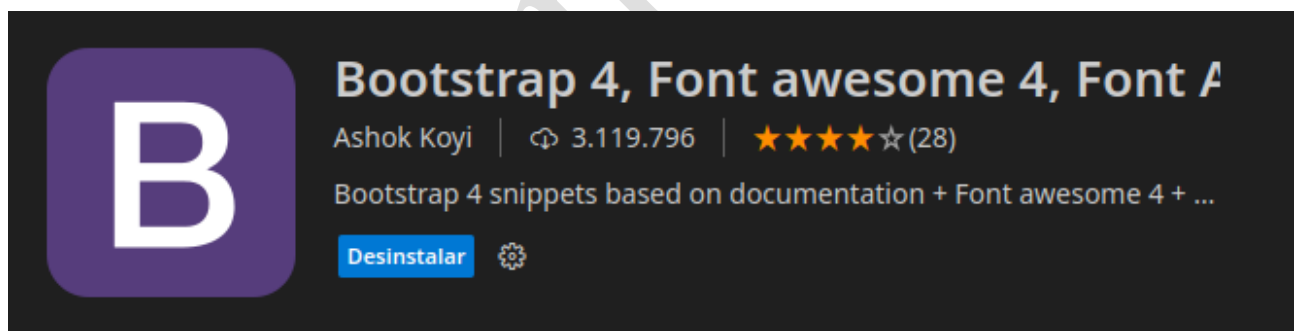
## 5º. Abra o Projeto do Visual Code

- Crie o arquivo index.html e incorpe o bootstrap, javascript e o jquery
- Crie o arquivo style.css dentro da pasta css e incorpore-o também.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <link rel="stylesheet" href="/bootstrap/css/bootstrap.css">
8   <link rel="stylesheet" href="/bootstrap/css/bootstrap.min.css">
9   <link rel="stylesheet" href="/css/style.css">
10
11
12 </head>
13 <body>
14
15
16
17   <script src="/bootstrap/js/bootstrap.js"></script>
18
19   <script src="/javascript/jquery.js"></script>
20 </body>
21 </html>
```

## 6º. Instale a Extensão no Visual Code



## 1.3 Criando Layouts

---

### 1ª – Crie a base para um layout

```
<body>
  <div class="container text-center" >

  </div>

  <!--incorporando jquery-->
  <script src="/javascript/jquery.js"></script>
</body>
```

### Lembre-se sempre de colocar também o js do bootstrap

```
<script src="/bootstrap/js/bootstrap.js"></script>
```

---



**O Container** → é algo maior que vai envolver toda a nossa interface.

O class .container tem uma largura fixa para cada tamanho de tela em bootstrap (xs,sm,md,lg), já o class .container-fluid expande para preencher a largura disponível.

---

### 2ª – Visite o site <https://getbootstrap.com.br/>

- Vá em documentação
- E em pesquisar busque por layout Grid

layout

Ponto de partida

Layout

Visão geral

Grid

Objeto de mídia

Utilitários de layout

c) Vá em **parametros** da Grid para entender este comportamento

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Largura máxima do container	Não tem (auto)	540px	720px	960px	1140px
Prefixo em classe	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Número de colunas	12				
Largura da gutter	30px (15px em cada lado da coluna)				
Aninhável	Sim				
Ordenamento de coluna	Sim				



**Parametros da Grid →** Observe as diferentes classes para cada tamanho de tela. Quando você não define o parametro, ele só vai se modificar quando você estiver usando o layout em uma interface < 576px.

d) Vamos criar o nosso primeiro layout

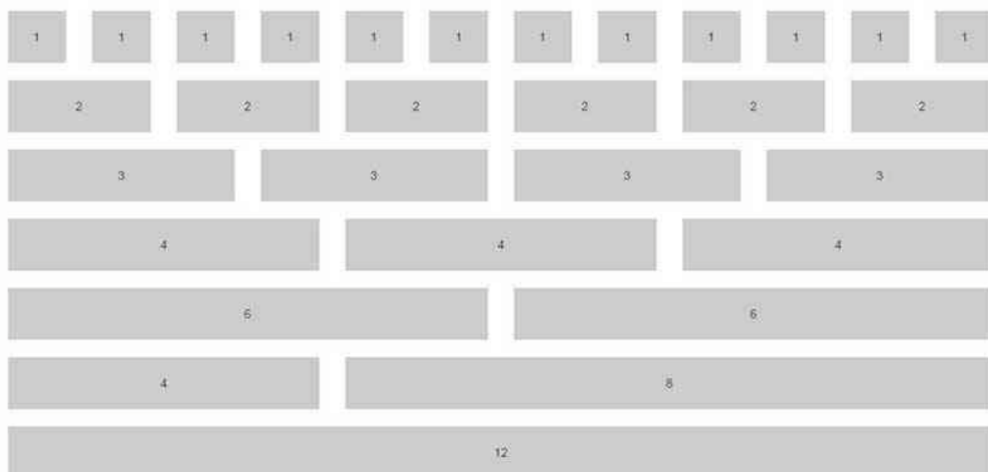
```
<div class="container text-center" >
  <div class="row">

    <div class="col-2 ">IFSP</div>
    <div class="col-8 ">Capivari</div>
    <div class="col-2 ">Dri</div>
  </div>

  <div class="row">

    <div class="col-2">IFSP</div>
    <div class="col-5">Capivari</div>
    <div class="col-5">Dri</div>
  </div>
</div>
```

O **Sistema de Grid** do Bootstrap, é a base para organizarmos o Layout da nossa aplicação Web. Este sistema divide a UI em linhas e colunas. Na raiz temos uma área básica que é um container. Dentro deste container podemos criar linhas e dentro das linhas colunas, para assim organizarmos a posição dos elementos na tela.



**12 →** lembre-se que o limite de cols é 12. Quando você não defini um tamanho elas serão distribuídas até atingir o tamanho 12.

---

## 1.4 Cores de Texto e Fundo

---

Podemos utilizar a estilização do próprio bootstrap para configurar cores de texto e fundo nos diversos elementos.

```
<div class="container text-center" >
  <div class="row bg-dark text-primary">

    <div class="col-2 ">IFSP</div>
    <div class="col-8 ">Capivari</div>
    <div class="col-2 ">Dri</div>
  </div>

  <div class="row bg-light text-secondary">

    <div class="col-2">IFSP</div>
    <div class="col-5">Capivari</div>
    <div class="col-5">Dri</div>
  </div>
</div>
```

Entre elas temos:



## 1.5 Criando suas próprias cores e estilos

---

Para isso você deve criar a sua própria folha de estilo que irá se sobrepor a estilização padrão do bootstrap

```
# style.css >  .cor1
.cor1{
  background-color:  #ffcc88;
  text-align: justify;
  color:  blue;
}

.cor2{
  background-color:  #ccff88;
}

.cor3{
  background-color:  #bb88cc;
}
```

Feito isso, você troca a classe nativa do bootstrap por sua classe de cor, utilizando-a tanto na linha como na coluna, como mostrado abaixo:

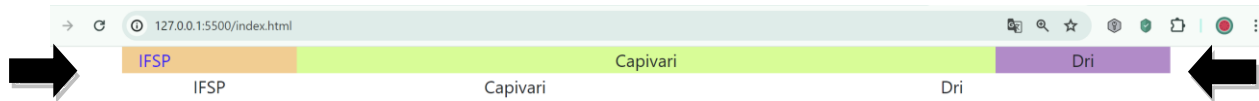
```
<div class="container text-center" >
  <div class="row">

    <div class="col-2 cor1">IFSP</div>
    <div class="col-8 cor2">Capivari</div>
    <div class="col-2 cor3">Dri</div>
  </div>

  <div class="row">

    <div class="col-2 cor1">IFSP</div>
    <div class="col-5 cor2">Capivari</div>
    <div class="col-5 cor3">Dri</div>
  </div>
</div>
```

PERCEBA que nosso layout não é fluid, pois em cada lateral ainda sobra uma espaço em branco



Para resolver isso, você deve então colocar a opção fluid no container

```
<div class="container-fluid text-center" >
```

## 1.6 Testando a responsividade do layout

---

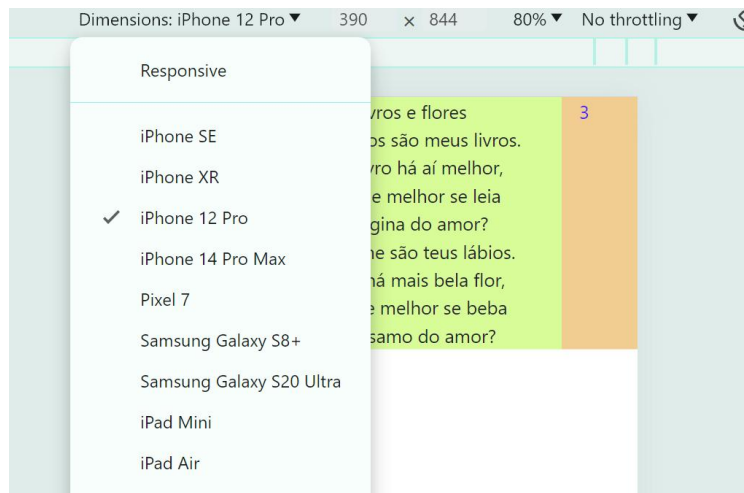
Para isso deixe seu código da seguinte forma, organizando o layout e cores.

```
<div class="container-fluid text-center" >
  <div class="row">

    <div class="col-2 cor1">1</div>
    <div class="col-8 cor2">Livros e flores <br>

    Teus olhos são meus livros.<br>
    Que livro há aí melhor,<br>
    Em que melhor se leia<br>
    A página do amor?<br>
    Flores me são teus lábios.<br>
    Onde há mais bela flor,<br>
    Em que melhor se beba<br>
    O bálsamo do amor?
  </div>
  <div class="col-2 cor1 ">3</div>
</div>
```

Faça o teste com este código e habilite as ferramentas do desenvolvedor analisando sua tela em diferentes dispositivos.



Faça o teste com o IPHONE 12 PRO.



Perceba que quando ele está muito pequeno ele simplesmente espremeu o conteúdo, então ele não está responsivo. Então eu posso dizer ao código que quando ele for muito pequeno, ele deve assumir um novo layout.

```
<div class="container-fluid text-center" >
  <div class="row">

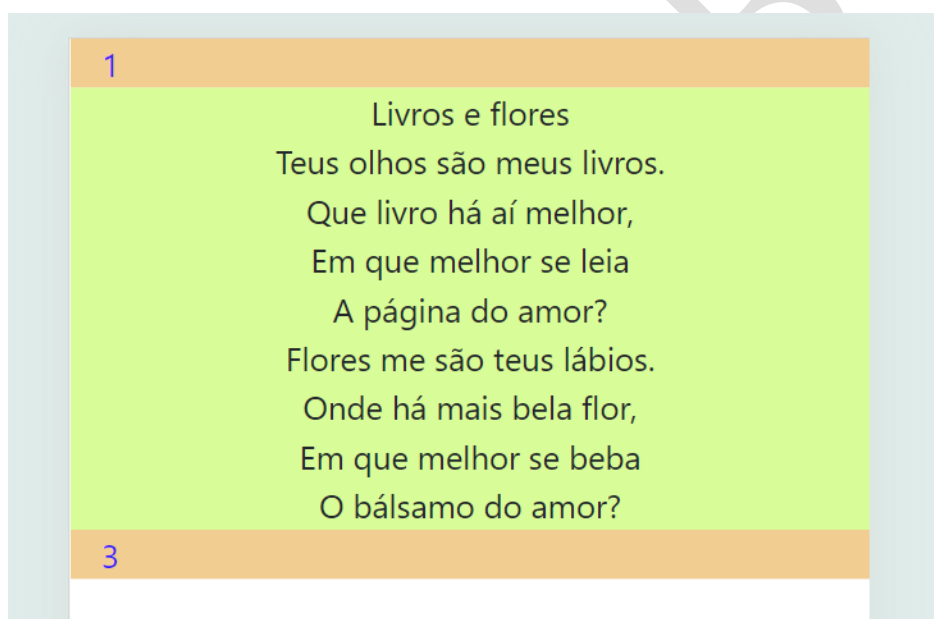
    <div class="col-sm-2 cor1">1</div>
    <div class="col-sm-8 cor2">Livros e flores <br>

    Teus olhos são meus livros.<br>
    Que livro há aí melhor,<br>
    Em que melhor se leia<br>
    A página do amor?<br>
    Flores me são teus lábios.<br>
    Onde há mais bela flor,<br>
    Em que melhor se beba<br>
    O bálsamo do amor?
  </div>
  <div class="col-sm-2 cor1 ">3</div>
</div>
```

Desta forma, toda vez que a interface for menor que 576 ela adotará um novo layout.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Largura máxima do container	Não tem (auto)	540px	720px	960px	1140px
Prefixo em classe	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Número de colunas	12				
Largura da gutter	30px (15px em cada lado da coluna)				
Aninhável	Sim				
Ordenamento de coluna	Sim				

Veja o resultado:



## 1.7 Criando uma separação nas colunas

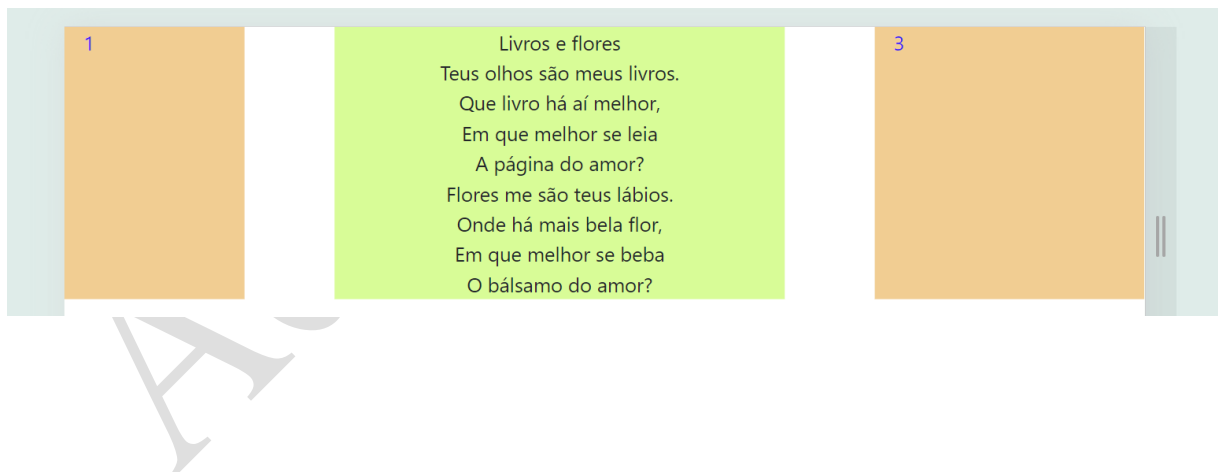
---

```
<div class="container-fluid text-center" >
  <div class="row">

    <div class="col-sm-2 cor1">1</div>
    <div class="col-sm-5 cor2 offset-md-1">Livros e flores <br>

      Teus olhos são meus livros.<br>
      Que livro há aí melhor,<br>
      Em que melhor se leia<br>
      A página do amor?<br>
      Flores me são teus lábios.<br>
      Onde há mais bela flor,<br>
      Em que melhor se beba<br>
      O bálsamo do amor?
    </div>
    <div class="col-sm-3 cor1 offset-md-1">3</div>
  </div>
</div>
```

Resultado:





## Exercício

Para desafio pessoal (não vale nota)

---

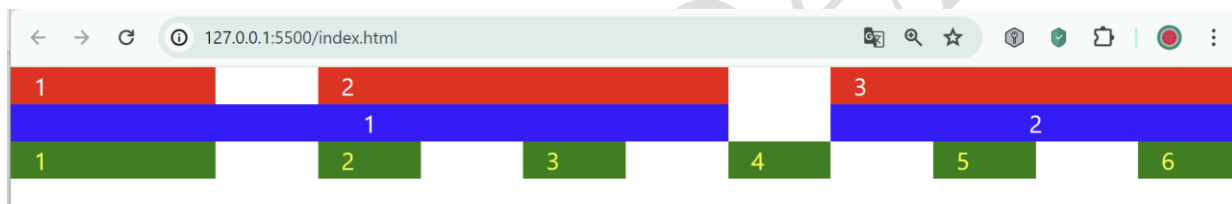
**Neste tipo de exercício a proposta é que o aluno se lance a prática sem a obrigação de entrega da atividade. Se auto avalie ao final.**

---

Com base nas explicações de criação de layout no bootstrap faça:

---

- Crie um layout fluido com a seguinte estrutura e que se adapte a um novo layout quando a medida for menor que 756px.
- Crie a folha de estilo para padronização da interface.





## Projeto Prático TUTORADO

### JOGO DA VELHA

*Java Script e BootStrap*

**01- Crie uma layout bootstrap para criação de um jogo da velha como mostrado abaixo:**



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="/bootstrap/css/bootstrap.css">
  <link rel="stylesheet" href="/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="/css/style.css">
</head>
<body>
  <div class="container-fluid text-justify" >
    <div class="row">
      <div class="col-sm-12 cor1"><h1> JOGO DA VELHA </h1></div>
    </div>
    <div class="row">
      <div class="col-sm-4 cor1"></div>
      <div class="col-sm-4 cor2">
        aqui nasce o JOGO
      </div>
      <div class="col-sm-4 cor1"> </div>
    </div>

    <div class="row">
      <div class="col-sm-12 cor1">Desenvolvido por: Adriana Belon</div>
    </div>
  </div>
  <!--incorporando jquery-->
  <script src="/javascript/jquery.js"></script>
  <script src="/bootstrap/js/bootstrap.js"></script>
</body>
</html>
```

## 02- Crie a estrutura do jogo na segunda linha do layout

```
<div class="row">
  <div class="col-sm-4 cor1"></div>
  <div class="col-sm-4 cor2">
    <main>
      <div class="game">
        <button data-i="1"></button>
        <button data-i="2"></button>
        <button data-i="3"></button>
        <button data-i="4"></button>
        <button data-i="5"></button>
        <button data-i="6"></button>
        <button data-i="7"></button>
        <button data-i="8"></button>
        <button data-i="9"></button>
      </div>
      <h2 class="currentPlayer"></h2>
    </main>
  </div>
  <div class="col-sm-4 cor1"> </div>
</div>
```

## 03- Crie o arquivo style.css

```
.cor1{
  background-color:#ffcc88;
  text-align: center;
  color:white;
}
.cor2{
  background-color:#ccff88;
  display: flex;
  justify-content: center;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: cursive;
}

main {
  display: flex;
  flex-direction: column;
  gap: 5px;
}
```

```

h1 {
  text-align: center;
}

hr {
  font-weight: bold;
  height: 3px;
  background: black;
  margin-bottom: 10px;
}

.game {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}

.game button {
  width: 70PX;
  height: 70PX;
  margin: 5px;
  cursor: pointer;
  font-size: 50px;
  background: #f7f7f7;
}

```

### 03- Crie o arquivo script.js com a lógica do game e salve dentro da pasta javascript

```

const currentPlayer = document.querySelector(".currentPlayer");

let selected;
let player = "X";

let positions = [
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9],
  [1, 4, 7],
  [2, 5, 8],
  [3, 6, 9],
  [1, 5, 9],
  [3, 5, 7],
];

function init() {
  selected = [];

  currentPlayer.innerHTML = `JOGADOR DA VEZ: ${player}`;

  document.querySelectorAll(".game button").forEach((item) => {

```

```

        item.innerHTML = "";
        item.addEventListener("click", newMove);
    });
}

init();

function newMove(e) {
    const index = e.target.getAttribute("data-i");
    e.target.innerHTML = player;
    e.target.removeEventListener("click", newMove);
    selected[index] = player;

    setTimeout(() => {
        check();
    }, [100]);

    player = player === "X" ? "O" : "X";
    currentPlayer.innerHTML = `JOGADOR DA VEZ: ${player}`;
}

function check() {
    let playerLastMove = player === "X" ? "O" : "X";

    const items = selected
        .map((item, i) => [item, i])
        .filter((item) => item[0] === playerLastMove)
        .map((item) => item[1]);

    for (pos of positions) {
        if (pos.every((item) => items.includes(item))) {
            alert("O JOGADOR " + playerLastMove + " GANHOU!");
            init();
            return;
        }
    }

    if (selected.filter((item) => item).length === 9) {
        alert("DEU EMPATE!");
        init();
        return;
    }
}

```

### 03- Incorpore o arquivo script ao código

```
<script src="/javascript/jquery.js"></script>  
<script src="/bootstrap/js/bootstrap.js"></script>  
<script src="/javascript/script.js"></script>
```

## RESOLUÇÕES:

### EXERCÍCIO 1

```
<div class="container-fluid text-center" >
  <div class="row">
    <div class="col-sm-2 cor1">1</div>
    <div class="col-sm-4 cor1 offset-md-1">2 </div>
    <div class="col-sm-4 cor1 offset-md-1">3</div>
  </div>

  <div class="row">
    <div class="col-sm-7 cor2">1</div>
    <div class="col-sm-4 cor2 offset-md-1">2 </div>
  </div>

  <div class="row">
    <div class="col-sm-2 cor3">1</div>
    <div class="col-sm-1 cor3 offset-md-1">2 </div>
    <div class="col-sm-1 cor3 offset-md-1">3 </div>
    <div class="col-sm-1 cor3 offset-md-1">4 </div>
    <div class="col-sm-1 cor3 offset-md-1">5 </div>
    <div class="col-sm-1 cor3 offset-md-1">6 </div>
  </div>

</div>
```