



do básico ao avançado



O que é Bootstrap?

- O **Bootstrap** é uma biblioteca orientada a **componentes**;
- Ou seja, **partes do layout já vem criadas** e nos cabe alterá-las para o que desejamos;
- **Agilizando** muito a parte de desenvolvimento front-end;
- Criando também um **padrão de grid**, facilitando a exibição de conteúdo baseado na resolução;
- E é o **framework mais utilizado desta categoria**;



Novidades da versão 5

- O **Bootstrap 5** foi uma grande revolução no framework;
- Temos agora **ícones** próprios;
- **Novos componentes**;
- Independência do **jQuery**;
- Melhoria nos componentes de formulário;
- **Simplificação das classes** de alguns componentes;
- Facilitação e melhorias na **customização**;



Instalando o VS Code

- O **VS Code** é um dos editores de texto mais utilizados atualmente;
- Vamos utilizar ele como **principal editor do curso**;
- Além disso, possui recursos muito interessantes para devs, como o **terminal integrado**;
- Você pode fazer o download em: <https://code.visualstudio.com/>



Instalando o Node.js

- O **Node.js** é uma runtime de JS, que nos permite criar projetos de JavaScript, tanto para o back-end quanto para o **front**;
- **É comum também instalar o Bootstrap nestes casos**, e a instalação é um pouco diferente;
- Vamos instalar ele com o **NPM**, um gerenciador de pacotes;
- Tornando o framework uma **dependência** do nosso projeto;
- Vamos utilizar em algumas situações;



Primeiro projeto com Bootstrap

- Nesta aula vamos criar nosso **primeiro projeto com Bootstrap**;
- Instalaremos por meio do **CDN**;
- CDN é um servidor que distribui geralmente arquivos estáticos em cache, **carregando de maneira mais rápida** no nosso projeto;
- Há **prós e contras** em usar o CDN;
- Vamos a instalação;



Como se programa com Bootstrap?

- **Difícilmente vamos gravar todas as classes**, ninguém faz isso;
- Precisamos **conhecer os componentes** (diferente de gravar);
- Consultar **documentação**;
- Copiar o componente e excluir o que não vamos utilizar ou criar o nosso a partir da referência das classes;
- **Então por que um curso?**



Como tirar um melhor proveito?

- **Programar junto** e não ficar só assistindo;
- Criar **seus próprios exemplos** (para conteúdos das aulas);
- **Projetos próprios** ao final do curso ou durante;
- Realizar os **exercícios** propostos;
- **Responder** ou criar novas perguntas no Q&A;
- Dica extra: ver depois fazer;





Introdução

conclusão da seção





Primeiros passos

introdução da seção



Diferentes formas de instalação

- No Bootstrap temos formas diferentes de fazer o setup;
- **CDN**: a que já vimos, servidor externo que detém os arquivos estáticos;
- **Arquivos internos**: Instalamos o Bootstrap a partir de arquivos no nosso servidor;
- **npm**: Gerenciador de pacote do Node.js, que pode instalar o Bootstrap como dependência;
- **Há outras formas**, porém estas são as mais utilizadas/comuns;



Arquivos de JS do Bootstrap

- No Bootstrap temos também a possibilidade de instalar **arquivos de JS**;
- **Não são obrigatórios**, mas nos dão algumas funcionalidades;
- Algumas delas são: Menu de dropdown, slider carousel, Menu de hamburger, tooltips e outros;
- **Para projetos mais simples** apenas o CSS já costuma resolver o problema;
- Os arquivos são divididos em: **Popper** e os do Bootstrap;



Instalação com arquivos

- Para instalar com arquivos no nosso servidor, precisamos **fazer o download do Bootstrap**;
- Os arquivos estão disponíveis no site oficial;
- Colocaremos os arquivos em **diretórios do nosso projeto**;
- E **realizaremos o link de CSS e JavaScript**, a partir dos caminhos relativos;
- Vamos lá!



Instalação com npm

- Para instalar com o npm, vamos precisar do Node.js (instalamos anteriormente), rodar o comando **npm init -y**;
- Instalar a dependência com: **npm i bootstrap@next**;
- Fazer os links no HTML (**node_modules/bootstrap/dist**);
- Vamos lá!



Configurações importantes

- Há **dois elementos necessários** para o correto funcionamento do Bootstrap que precisamos nos atentar no HTML;
- **DOCTYPE**: O Bootstrap necessita das tags corretas de DOCTYPE;
- **viewport**: E também da tag que controla a responsividade;
- Sem estas configurações ele pode funcionar, porém estamos sujeitos a **instabilidades**;
- Vamos vê-las no HTML!



O que vem nos arquivos do BS?

- Quando fazemos o download ou instalamos por npm, **vários arquivos estão disponíveis**;
- Neles teremos arquivos mais simplificados, **minificados** ou também com apenas alguns dos componentes;
- Os de **JavaScript** também seguem este padrão;
- Vamos dar olhada!



Navegadores suportados

- **Nem tudo são flores**, alguns navegadores não são totalmente suportados pelo Bootstrap;
- Temos que **ficar atentos** a isso quando o utilizamos;
- A compatibilidade mínima se baseia nas **versões dos browsers**;
- Vamos ver quais são eles!





Primeiros passos

Conclusão da seção





Layout

Introdução da seção



Inserindo nosso CSS

- O estilo do Bootstrap pode não ser suficiente para finalizar uma aplicação, **então inserimos nosso CSS**;
- Vamos **linkar sempre abaixo do BS**;
- Nossas **regras precisam ter um nível de seleção pelo menos igual o** do BS para substituí-las;
- Vamos a prática!



O que são breakpoints?

- São **pontos específicos de resolução** criados através de **media query** para deixar o site responsivo;
- Eles seguem um padrão que é encontrado na **maioria dos dispositivos**;
- Ou seja, temos um breakpoint para tablet e outro para celular;
- **Não há para todos**, mas para os mais utilizados;
- Lembrando que é utilizado um padrão **mobile first**, ou seja, estilização de menor resolução para maior;



Os breakpoints do Bootstrap

- Os breakpoints do Bootstrap são utilizados por classes como: **sm, md, lg**;
- Cada um representa uma **media query**;
- Então normalmente **adequamos outros elementos** do nosso site a estes breakpoints;
- Assim é possível **trabalhar com conjunto do Bootstrap** e deixar o site responsivo;
- Vamos ver na prática!



Containers

- Containers são **entidades super importantes** do Bootstrap, estruturam o layout;
- E como o nome diz são containers, **depósitos de outros elementos** =)
- Temos **tipos diferentes de containers**, vamos ver como funcionam cada um deles;
- Mas o uso é o mesmo: separar o site em áreas;



Container

- Para inserir um container normal, vamos precisar adicionar apenas a classe **container** a uma div;
- Esta variação se comporta **obedecendo uma max-width** em cada breakpoint;
- Este acaba sendo o container mais utilizado;
- Costuma **não encostar nas bordas** da página;
- Vamos a prática;



Container Fluid

- O Container Fluid **utiliza uma área maior** para abrigar os elementos;
- Sempre **representa 100% da área útil**;
- Eliminando as margens laterais;
- A classe é **container-fluid**;



Container por breakpoint

- Podemos também moldar um container pelo breakpoint;
- Utilizamos a classe **container-x**;
- Onde x pode ser: **sm, md, lg** e outras classes de breakpoint também;
- Com esta versão, queremos um container que **respeite uma largura máxima, que se baseia em uma resolução;**



Grid

- **Grid** é a estrutura para criar layouts com Bootstrap;
- Temos um **grid colunar**, dividido em **12 colunas**;
- A estrutura do Grid é sempre: **container > row > colunas**;
- Onde temos o **container** para abrigar a seção;
- **Row** para determinar uma linha (largura 100%);
- **Colunas** para subdividir o container em até 12 partes;



Criando nosso grid

- Vamos agora criar **nosso primeiro grid**;
- Envolver todos os elementos com um **container**;
- **Inserir uma ou mais rows**, dependendo do conteúdo;
- E **dividir em colunas**;
- Vamos para a prática!



Tamanho das colunas

- Podemos **ajustar o tamanho das colunas**;
- Para ocupar três espaços: **cols-3**;
- Lembrando que podemos separar em no máximo **12 divisões**;
- E o espaço ocupado será proporcional, ou seja, em: col-11 e col-1, teremos uma coluna ocupando quase 100% da largura da row;



Limitando número de colunas

- Podemos também **limitar a quantidade de colunas** em uma row;
- A classe a ser utilizada é: **row-cols-***;
- Onde ***** é o número limite;
- Assim o **grid sempre vai respeitar este número** máximo na linha;



Alinhamento vertical

- As colunas dentro de uma row **estão condicionadas ao flex box**;
- Ou seja, temos **classes bem parecidas com as regras de flex** para alinhá-las;
- Exemplo: **align-items-end**, alinha no fim da row;
- Lembrando que esta classe **sempre deve ficar no elemento pai** (row);



Alinhamento horizontal

- O **alinhamento horizontal** segue a mesma premissa do vertical;
- **Utilizamos classes no elemento pai (row)** para condicionar os itens na linha horizontal;
- Para centralizar elementos podemos por a classe: **justify-content-center**;
- As regras **end e start** também funcionam;
- Podemos **combinar** também vertical e horizontal;



Alinhamento de itens

- O alinhamento de item serve para **alinhar**mos um item individualmente;
- Utilizamos classes como: **align-self-center**;
- Que alinha um elemento na vertical;
- **Não há alinhamento** unitário para a horizontal;



Ordem dos itens

- É possível **alterar a ordem que os itens são exibidos**;
- Vamos colocar a classe: **order-***;
- Onde * é a posição do elemento, sendo **1 o primeiro**;
- Este recurso é útil no **desenvolvimento mobile**, onde os elementos costumam mudar de ordem;



Offset de colunas

- Podemos também **criar um espaçamento** com offset;
- Ou seja, teremos uma margem da borda definida por uma **quantidade de colunas**;
- Exemplo: **offset-md-3**;
- Neste caso temos um offset de 3 colunas a esquerda;



Gutters

- Os **gutters** são intervalos que podemos inserir entre as colunas;
- É possível inserir no **eixo x ou y**;
- Exemplo: **gx-2** (horizontal);
- Ou: **gy-4** (vertical);
- Os números que utilizamos são **baseados em rem da página**, fonte do elemento root;
- O número **3 é a referência para 1rem**;



Tarefa 01

1. Crie um projeto com Bootstrap e um arquivo de CSS próprio do projeto;
2. Crie uma estrutura de grid;
3. Limite a estrutura para 4 colunas por linha;
4. Crie 8 colunas no total;
5. As colunas ímpares e pares devem ter cor diferente, altere a cor de fundo com background-color;
6. Defina bem as divisões das colunas, pode ser com borda;





Layout

Conclusão da seção





Ícones

Introdução da seção



Bootstrap Icons

- Na versão 5 do Bootstrap temos disponíveis ícoes em **SVG e de forma gratuita**;
- São **mais de 1300** atualmente;
- Isso nos possibilita a **não depender de bibliotecas de terceiros** para este fim;
- Podemos usar **com ou sem** o Bootstrap;
- É possível instalar por **npm, arquivos ou CDN**;



Instalando os ícones

- **Vamos instalar por CDN**, pela maior simplicidade;
- Para utilizar os ícones vamos criar uma **tag i**;
- Nesta tag vamos **inserir classes** respectivas para exibir um determinado ícone, ex: **bi bi-x-lg**
- Estas classes apresentam um ícone de x;
- Podemos utilizar os ícones em menus, botões e em outras partes do site;
- Vamos lá!



Como encontrar os ícones?

- Sempre vamos **utilizar a documentação como referência**;
- Buscamos o ícone necessário;
- **Implementamos a suas classes e tag**, conforme o exemplo;
- Depois podemos **fazer as alterações necessárias** para inserir no projeto;
- Vamos ver como encontrar ícones!



O que são os ícones?

- Os ícones **são considerados fontes**, como Arial;
- Ou seja, vamos alterar seu tamanho com **font-size** e sua cor com **color**;
- É um elemento inline, por causa da **tag i**;
- Então precisamos mudar o display, caso precise ser de bloco;
- Alterando a cor de fundo, mudamos o fundo da tag, e não há alteração no ícone;



Tarefa 02

1. Crie um projeto com Bootstrap, fazendo a inserção da biblioteca como você preferir;
2. Crie um grid com três linhas;
3. Onde cada uma deve ter um ícone e ao lado deste ícone uma descrição do mesmo;
4. Altere a cor e o tamanho de cada ícone, todos devem ser diferentes;
5. Centralize o conteúdo do na horizontal, adicione padding também;





Ícones

Conclusão da seção





Conteúdo

Introdução da seção



Headings

- O Bootstrap implementa estilos próprios das **tags h1 a h6** para servir como títulos no nosso site;
- Além da fonte diminuir, também temos uma **redução da margem inferior**;
- Podemos utilizar **classes de h1 a h6** para elementos que não são headings terem o mesmo visual;
- Lembrando que os **headings são utilizados devido a semântica (SEO)**, estilos não substituem isso;



Display

- As classes de **display** são utilizadas para aumentar os headers;
- Utilizamos **display-***, onde * é um número de 1 a 6;
- Lembrando que **o efeito é visual**, não altera SEO;
- Esta classe também trás uma **fonte mais fina**;
- **Obs:** normalmente seguimos o design do projeto, onde estas classes podem não representar o tamanho correto da fonte;



Elementos inline

- O **Bootstrap modificou algumas tags** para representarem estilo e trabalharem de forma inline;
- Lembrando que tags refletem sempre em SEO, CSS não;
- **mark**: para dar um highlight;
- **del**: texto “deletado”;
- **strong**: texto em negrito;
- Lembrando que **há classes para representar as tags em estilo**;



Citações

- Há um **componente para citações** em Bootstrap;
- Utilizaremos a **tag blockquote**, que é a tag de HTML para citação;
- Com algumas classes auxiliares do Bootstrap;
- Vamos ver na prática!



Listas

- Nas classes de Bootstrap **podemos modificar um pouco as listas**;
- **list-unstyled**: remove os estilos da lista;
- **list-inline**: deixa horizontal;
- **Geralmente removemos estes estilos**, desta maneira nossa vida fica mais fácil =)



Imagens

- Para adicionar uma imagem de forma responsiva, podemos colocar a classe **img-fluid**;
- A **imagem vai se adaptar a resoluções** diferentes;
- **A escala é referente ao elemento pai** da imagem;
- Vamos ver na prática!



Thumbnail

- Temos um componente para criar **thumbnails**;
- **Uma borda é aplicada a imagem** e um pequeno raio também;
- Criando uma espécie de **moldura**;
- Vamos ver na prática!



Figure

- A figure é uma tag utilizada para **quando há a necessidade de legenda para uma imagem**;
- E o **Bootstrap possui um componente** que estiliza esta tag;
- Podemos então deixar num **padrão interessante** o conjunto de foto + texto;
- Vamos ver na prática;



Tabelas

- O Bootstrap também tem um componente muito interessante para **tabelas**;
- **Vamos estilizá-las de forma interessante**;
- E ainda conseguimos controlar algumas **propriedades de cores**;
- Vamos ver na prática!



Tarefa 02

1. Crie uma estrutura de grid;
2. Nesta estrutura insira um heading, tamanho 2 com texto como um título;
3. Embaixo deste texto insira um parágrafo, com algum texto de impacto (para sinalizar utilize a tag strong);
4. Embaixo do parágrafo coloque uma imagem responsiva, em 6 colunas;
5. Depois da imagem coloque duas listas, uma ao lado da outra (3 colunas), descrevendo a imagem em cada item das listas;





Conteúdo

Conclusão da seção





Formulários

Introdução da seção



Formulários

- Os formulários possuem **diversos inputs no HTML**;
- Consequentemente teremos **vários componentes no Bootstrap**;
- O interessante é que **temos alguns padrões** que são repetidos entre os componentes, tornando mais fácil o aprendizado;
- Vamos criar um form!



Desabilitando formulários

- Podemos também **desabilitar um formulário ou campos dele**;
- **Para um form**: precisamos de uma tag fieldset com atributo disabled, envolvendo um form;
- **Para um input**: precisamos do atributo disabled no campo;
- Vamos ver na prática!



Form control

- **form-control** é uma classe para ajustar os inputs;
- Fazendo com que eles fiquem no padrão do Bootstrap;
- Ou seja, **na maioria dos inputs adicionamos esta classe**;
- Temos variações com: **form-control-***;
- Onde * pode ser sm, lg (**variações de tamanho**);
- E a label leva a classe **form-label**;



Somente leitura

- O **readonly** é um atributo que indica somente leitura;
- Temos duas formas de apresentação no Bootstrap:
- **Atributo**: colocamos o atributo readonly e o input recebe os estilos do componente;
- **Classe do BS**: o input fica parecendo apenas um texto;
- Obs: readonly é uma forma de enviar os dados para o servidor, já o disabled não envia!



Arquivos

- O **input de arquivos** tem uma forma diferente de receber CSS, o que torna a vida dos devs mais difícil;
- Mas a notícia boa é o Bootstrap tem um componente para isso;
- Deixando o **input de file mais bonito**;
- Vamos ver na prática!



Datalist

- **Datalist** é um select, porém com busca;
- Então você pode **inserir as opções e também encontrar alguma** que está cadastrada;
- Temos um componente no Bootstrap para este input;
- Vamos lá!



Cor

- Há também um componente para **input de cor**;
- Teremos uma **espécie de color picker** bem estilizado;
- Este input **não é muito utilizado**, mas é interessante o cuidado que o Bootstrap teve em incluir ele;
- Vamos ver na prática!



Select

- Há **dois tipos de select**:
- **Valor único**: usuário só pode escolher uma opção;
- **Valores múltiplos**: usuário pode escolher uma ou mais;
- O Bootstrap possui componente para os dois;
- Vamos ver na prática!



Checkbox e radio button

- Outros inputs bem conhecidos são:
- **Checkbox**: onde podemos selecionar mais de uma opção;
- **Radio**: selecionamos apenas uma opção;
- Temos componentes de Bootstrap para estes também;
- Vamos ver na prática!



Switch

- O **switch** é um checkbox;
- Porém ele tem uma **estilização diferente**;
- **Parecendo um botão de “on/off”**, bem interessante para aplicações modernas;
- Vamos ver na prática!



Range

- **Range** é um input interessante;
- Onde o usuário pode **regular um parâmetro por uma barra**;
- E ainda podemos definir um **valor mínimo e máximo**;
- Temos componentes no Bootstrap para isso também;
- Vamos ver!



Input groups

- Os **input groups** servem para darmos uma experiência melhor para o usuário;
- Às vezes **agrupando funções**;
- Ou tornando o input mais objetivo;
- Vamos ver na prática;



Label flutuante

- As **floating labels** são uma forma de deixar a label dentro do input;
- Semelhante aos **formulários do Google**;
- **Ocupando menos espaço** e deixando o layout bonito;
- Vamos ver na prática!



Layout com forms

- Os conhecimentos que aprendemos na **seção de Layout** são úteis em formulários;
- Podemos condicionar a **exibição dos inputs pelo grid** de colunas;
- Ou seja, há uma forma fácil de **inserir inputs em uma linha**;
- E também de diferentes larguras;
- Vamos ver na prática!



Validações

- O Bootstrap também nos dá uma possibilidade de **criar validações**;
- Vamos precisar inserir algumas **classes de validação**;
- Informar um feedback para o que está certo e o que está errado;
- E também **observar o envio do formulário**, para poder ativar a validação antes que isso ocorra;
- Vamos ver na prática!



Validação do HTML

- O Bootstrap funciona bem também com a **validação padrão do HTML**;
- Onde os **atributos são utilizados** para dizer se um dado está errado ou não;
- Vamos ver na prática!



Tarefa 03

1. Crie um formulário de cadastro de produto;
2. Coloque campos como: título, descrição, preço e outros que você achar interessante;
3. Crie também uma checkbox, marcar sinaliza que o produto está em estoque;
4. Crie um select para seleção de categoria;
5. Insira validações, se desejar;





Formulários

Conclusão da seção





Componentes

Introdução da seção



O que são componentes?

- Até agora utilizamos muitos componentes, porém eles parecem muito com o que já temos do HTML;
- Estes componentes que veremos são elementos como **um card**;
- **Muitas tags e classes em conjunto** para chegar num resultado final mais complexo;
- Ou seja, os primeiros que utilizamos **era uma questão de CSS**;
- Agora é **estrutural** e também com **funcionalidades (precisa do JS)**;



Accordion

- O **accordion** é uma lista com elementos ocultos;
- **Ao clicar no título** de cada item da lista, uma descrição é exibida;
- Enquanto isso **o outro item que estava aberto, é fechado**;
- Este componente é interessante quando precisamos **poupar espaço**;
- Vamos ver na prática!



Alertas

- Os componentes de alerta servem para **exibir mensagens**;
- Temos **diversas cores disponíveis**, sinalizando cada ação possível;
- Como: erros, confirmações, informações e etc;
- Estas mensagens são como **feedbacks** do sistema para o usuário final;



Badge

- O componente **Badge** serve para exibir alguma **informação ao lado de texto**;
- Como **label**, exemplo: produtos novos (new);
- Ou **notificações**, com um número, exemplo: Mensagens 9;
- Vamos ver na prática!



Breadcrumb

- Este componente serve para mostrar ao usuário o caminho que já percorreu no site;
- Podemos mostrar as páginas principais;
- Desta maneira ele consegue voltar para as anteriores;
- Vamos ver na prática!



Buttons

- O componente de botões tem **diversas variações**;
- Utilizamos a classe base **btn**;
- Podemos inserir o estilo em **outras tags**, como a de links ``;
- Há uma **variação para outline**, que exibe cor só no contorno;
- Podemos também inserir estilo de **botão desabilitado**;
- Vamos ver!



Button Group

- O Button Group é um **agrupamento de botões**;
- Onde podemos criar componentes como **paginação**;
- Ou se precisarmos, **botão unidos** que configuram algo;
- Vamos ver na prática!



Cards

- Os **cards** são componentes bem interessantes;
- Podemos inserir **imagem, descrição e um botão** de chamada;
- São utilizados para: produtos, avatares e outros itens;
- Há **diversas variações** também;
- Vamos ver na prática!



Slider

- Podemos criar também um **slider** com Bootstrap;
- O nome do componente é **Carousel**;
- Há recursos para **passar a imagem** e também para **exibir a quantidade delas**;
- Vamos ver na prática!



Close button

- O close button é um **componente de botão fechar**;
- Porém **não precisamos dos ícones** instalados;
- O que torna uma **vantagem** neste sentido, caso não queremos utilizar outros ícones;
- Vamos ver na prática!



Collapse

- **Collapse** é um componente que tem duas partes;
- **Um texto**, que será exibido condicionado a um evento de click em um botão;
- **Um botão**, que exhibe ou esconde o texto;
- Vamos ver na prática!



Dropdown

- O dropdown é um **componente para menu**;
- Onde podemos colocar **um botão, com um evento**;
- Ao clicar nele, **abrimos uma lista** com os demais itens do menu;
- Vamos ver na prática!



Modal

- O **modal** possui dois elementos;
- **Um botão** de ativação;
- E **um banner** que aparece após o clique no botão;
- É útil para colocar mensagens ou realizar ações dentro do modal;
- Vamos ver na prática!



Navbar

- A navbar é o componente para as **barras de navegação**;
- Temos diversas variações;
- Podemos colocar **ícones, links, dropdowns e mais**;
- Vamos ver na prática!



Offcanvas

- Offcanvas é parecido com o Modal;
- Um conteúdo que fica oculto, a espera de um botão;
- Porém tem uma perspectiva diferente, costuma aparecer pela lateral;
- Interessante para carrinhos de e-commerce, por exemplo;
- Vamos ver na prática!



Paginação

- Podemos também utilizar um **componente próprio para paginação**;
- Ele é o **pagination**;
- Onde há alguns recursos a mais que o group button;
- Vamos ver na prática!



Popover

- O **popover** é um elemento que necessita da ação de um botão;
- Ao clicar no botão, **uma área informativa aparece**;
- Precisamos de um **código JS para ativar os popovers**;
- Vamos ver na prática!



Progress

- O **Progress** é um componente para barra de progresso;
- Que vai se completando a medida da **alteração de um atributo**;
- Pode ser utilizado para **loading** ou **upload de arquivos**;
- Vamos ver na prática!



Scrollspy

- O **Scrollspy** é um componente para levar a barra de navegação a um local indicado;
- Geralmente utilizado para **fixar as seções do site**;
- Ao clicar em um botão **o scroll desce**;
- Vamos ver na prática!



Toast

- O Toast é um componente de **push notification**;
- Que **pode ser exibido através de um evento**;
- Serve para exibir mensagem para usuários;
- Vamos ver na prática!



Tooltip

- O **Tooltip** é um componente de “dica”;
- Onde podemos oferecer **informações em algum elemento**;
- Vamos precisar de um código **JavaScript para ativar**;
- Vamos ver na prática!



Tarefa 04

1. Crie um projeto com Bootstrap;
2. Coloque uma navbar no topo;
3. Embaixo da navbar insira um slider com 3 imagens;
4. O slider precisa conter as setas de mudança e marcação de imagens embaixo;
5. E depois crie um grid com 6 cards, no máximo 4 por linha;
6. Preencha os itens com o máximo de conteúdos reais que conseguir;





Componentes

Conclusão da seção





Helpers

Introdução da seção



O que são helpers?

- **Helpers** geralmente são funções que servem em qualquer local do projeto, para algum fim específico;
- No Bootstrap são **opções disponíveis para resolver problemas** ou dar variedade no nosso projeto (**geralmente classes**);
- **Exemplo:** resolver float e cores para links;
- São úteis para algumas situações, por isso é bom conhecê-los;



Clearfix

- **Clearfix** é uma forma de resetar o float;
- No CSS moderno **não utilizamos muito o float**;
- Porém em algum **projeto mais antigo** podemos encontrar esta regra;
- Se não utilizarmos o clear, **os elementos quebram na tela**;
- O Bootstrap nos oferece uma fácil solução para isso;
- Vamos ver na prática!



Links coloridos

- Temos também **diversas classes para estilização de links**;
- Os nomes são **parecidos com as classes de botão**;
- Como em outros elementos com cor do Bootstrap;
- **Exemplo:** azul = primary;
- Vamos ver na prática!



Ratio

- Um helper que serve para **manter a proporção de um elemento**;
- Podemos utilizar para **embed de vídeos**;
- Desta maneira o item não vai ficar desproporcional ao seu tamanho original;
- Vamos ver na prática!



Posições

- Temos um helper para **definir posições também**;
- Podemos por meio de classes **fixar elementos no topo ou no rodapé** da tela;
- **Menos regras de CSS para gerenciarmos**, apenas classes;
- Vamos ver na prática!



Visually Hidden

- Este helper serve para **esconder o conteúdo**;
- **Porém com um detalhe**: ele fica acessível para motores de busca e também para acessibilidade;
- Ou seja, podemos **colocar informações importantes** para pessoas com deficiência visual, por exemplo;
- Deixando a experiência mais rica para eles;



Link em elemento

- Este helper serve para **deixar o elemento todo clicável**;
- Ou seja, **podemos permitir o click em todo o card**, para levar a um link específico;
- **Sem esta classe envolvemos o elemento com uma tag a**, o que pode não ficar bom para a estética do código;
- Vamos ver na prática!



Limitação de texto

- Um helper que **coloca reticências no fim do texto**;
- **Dando uma ideia de resumo**, para não prejudicar o layout com a quantidade de caracteres;
- Ideal também para funcionalidades de “Continuar lendo”;
- Vamos ver na prática!





Helpers

Conclusão da seção





Utilities

Introdução da seção



O que são utilities?

- **Parecidos com helpers**, porém como nome diz: utilitários;
- Vamos poder **utilizá-los em diversos elementos**;
- Por meio das classes de cada um;
- **Exemplos:** cores, cores de background, bordas e etc;
- Alguns já utilizamos ao longo do curso, chegou a hora de ver todos!



Backgrounds

- Há algumas classes no Bootstrap para **adicionar cor de fundo** a elementos;
- Utilizamos a classe: **bg-***;
- **Onde * é uma classe de cor**, como primary;
- Estas cores são similares às de botões;



Bordas

- Temos também **utilities para bordas**;
- Onde podemos: **alterar cor, tamanho, raio e etc**;
- Tudo por meio de classes;
- Vamos ver na prática!



Cores

- Assim como as cores de background, também temos **cores de texto** como utilities;
- As **classes são bem parecidas também**;
- Podemos alterar as cores dos ícones, já que estes são fontes;
- Vamos ver na prática!



Display

- Podemos também **alterar a propriedade display com classes**;
- **Transformar elementos de bloco para inline** ou até escondê-los;
- Tornando prática a mudança de comportamento;
- Vamos ver na prática!



Flex

- Podemos também **alterar um elemento para o display de flex**;
- Desta maneira **é possível também alterar como os elementos internos se comportam**;
- Lembrando que o container já deixa as colunas em flex;
- Vamos ver na prática!



Interações

- Podemos alterar algumas **interações do usuário** com a página;
- Limitação de seleção de texto;
- Impossibilitar o acesso em algum link;
- Vamos ver na prática!



Sombreamento

- Também é possível **aplicar sombras** via classes;
- Há diferentes classes para diferentes **intensidades**;
- Vamos ver na prática!



Largura e Altura

- Há classes que nos permitem alterar os atributos de **altura e largura** de um elemento;
- **Podemos então criar “colunas”** com diferentes opções além do grid de 12;
- Vamos ver na prática!



Espaçamentos

- Podemos **alterar o espaçamento** dos elementos via classes;
- Ou seja, é possível alterar o **padding e margin**;
- Podemos também escolher a direção que será colocado espaçamento: **top, right, bottom ou left**;
- Vamos ver na prática;



Textos

- Há utilities para **alteração de textos** também;
- Configurações como: **alinhamento, overflow, case** e outras, podem ser alteradas;
- Vamos ver na prática;





Utilities

Introdução da seção

