**07/08 – BOOTSTRAP**

O Bootstrap é um framework amplamente utilizado para o desenvolvimento front-end e estilizações de páginas web. Ele foi criado pelo Twitter e oferece um conjunto de estilos predefinidos e componentes de layout que facilitam a criação de páginas web. Com o Bootstrap, podemos aproveitar uma série de recursos prontos para uso, como CSS Grid, formulários estilizados, botões, barras de navegação, modais, carrosséis, ícones e muito mais, tudo isso podendo ser adaptável a dispositivos móveis.

A principal vantagem do Bootstrap é sua abordagem de desenvolvimento ágil, permitindo que criemos rapidamente interfaces atraentes e responsivas sem ter que escrever todo o código CSS do zero. Além disso, ele é altamente personalizável, permitindo que os desenvolvedores ajustem facilmente o estilo e o comportamento dos componentes de acordo com as necessidades do projeto.

Para usar o Bootstrap em algum projeto seu, basta fazer o download dele pelo site oficial, por um link do CDN ou pelo npm. É necessário ter o jQuery e o Popper.js instalado, para que o Bootstrap funcione plenamente. Após a instalação, basta linkar os arquivos bootstrap.min.css, bootstrap.min.js, jquery.min.js e popper.min.js no seu arquivo HTML. A partir deste momento, você já poderá fazer uso do Bootstrap.

Site para obter templates prontos em Bootstrap: <https://wrapbootstrap.com/>.

**CONTEÚDO DOS ARQUIVOS BOOTSTRAP**

**.map:** arquivos com essa extensão mapeiam o código-fonte de um arquivo minificado. Ou seja, caso o desenvolvedor queira reverter um arquivo minificado para sua origem, ele pode fazer isso usando arquivos **.map.**

**.min:** arquivos com essa extensão são arquivos minificados, ou seja, comprimidos. Nesses arquivos, o conteúdo dele é enxugado ao máximo, reduzindo o seu tamanho.

**bootstrap-grid.css:** arquivo que possui todas as configurações do sistema de grid do bootstrap.

**bootstrap-reboot.css:** arquivo que garante a normalização, ou seja, a padronização da página em todos os navegadores.

**bootstrap.css:** arquivo principal, que possui configurações do **sistema de grid, normalização e componentes** do Bootstrap.

**bootstrap.bundle.js:** arquivo que contém os plugins do Bootstrap e o conteúdo do Popper, biblioteca utilitária para uso de Tooltips e Popovers.

**bootstrap.js:** arquivo que contém apenas os plugins do Bootstrap.

**CONTAINERS**

Os containers são a base do Bootstrap. Ele é como se fosse a caixa principal da nossa página, todos os componentes dela estão localizados nele. Existem dois tipos de containers que podemos usar:

**container:** possui uma largura fixa de 1100px.

**container-fluid:** sua largura ocupa o tamanho inteiro da tela e é fluido, ou seja, á medida que o tamanho da tela vai diminuindo, os elementos vão se ajustando automaticamente.

**MOBILE FIRST**

O Mobile First é um conceito onde o foco de desenvolvimento de uma aplicação é em como ela vai se comportar primeiro em telas menores, como a de dispositivos menores. Então, ao desenvolvermos um site, por exemplo, começamos a desenvolvê-lo para celular, e então ajustamos para telas maiores, como a de desktops.

O Bootstrap trabalha com esse conceito, possuindo o que chamamos de **Responsive Breakpoints.** Eles são checkpoints de medidas de tela. Com eles, nós vamos determinar o comportamento da nossa página de acordo com cada tamanho de tela. Veja a seguir quais são os Responsive Breakpoints do Bootstrap:

**xs:** largura até 576px (padrão);

**sm:** largura entre 576px e 768px;

**md:** largura entre 768px e 992px;

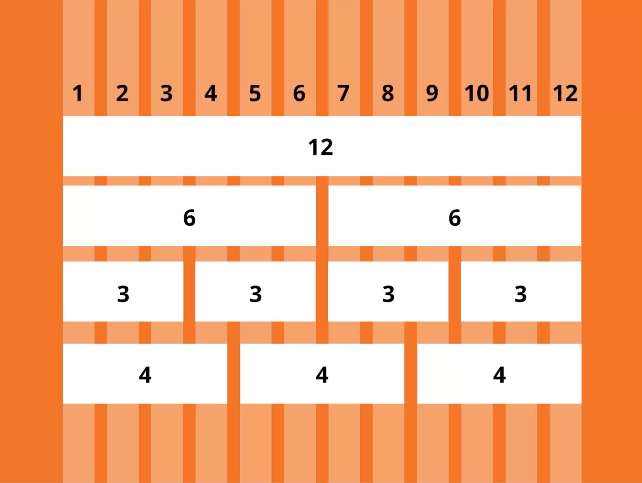
**lg:** largura entre 992px e 1200px;

**xl:** largura a partir de 1200px.

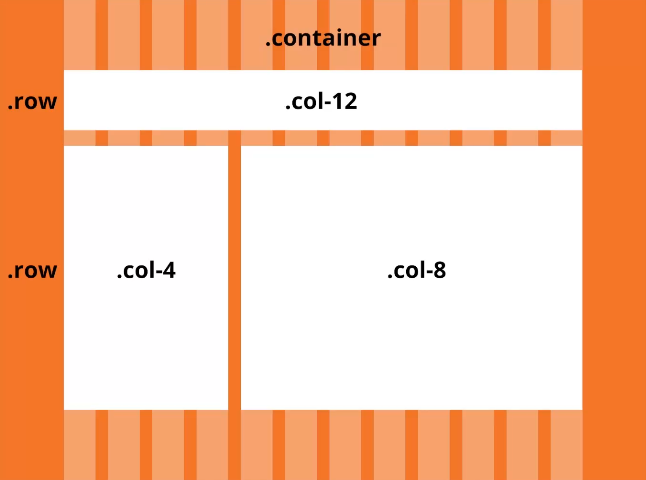
‘

**GRID SYSTEM**

O sistema de Grid do Bootstrap divide a tela em 12 colunas e 5 camadas (que são os breakpoints responsivos). Dessa forma, poderemos trabalhar de forma mais prática na criação de telas responsivas. Veja a seguir a disposição do Grid System:



Agora, veja um exemplo prático de uma aplicação usando esse sistema de Grid do Bootstrap:



**ORDER**

É possível alterar a ordem das colunas do Grid e determinar que uma coluna seja exibida antes de outra, sem alterar diretamente o HTML. Basta colocar em cada coluna a classe “order-x”, onde x é um número de 1 a 12. Quanto menor for o número, mais para o começo a coluna vai ficar. Quanto maior for o número, mais para o final a coluna vai ficar.

**OFFSET**

É possível deslocar uma coluna de maneira lateral no Grid System utilizando a classe “offset-x”, onde x é o número de colunas com base no qual a coluna determinada será deslocada á direita.

**FLEXBOX CONFIGS**

**d-flex** = display: flex

**flex-row =** flex-direction: row

**flex-row-reverse =** flex-direction: row-reverse

**flex-column =** flex-direction: column

**flex-column-reverse =** flex-direction: column-reverse

**justify-content-x =** justify-content: x (x pode ser center, around, between, end, start)

**align-items-x =** align-ttems: x (x pode ser stretch, center, start, end ou baseline)

**flex-fill:** distribui os elementos de maneira igual. É bem semelhante ao usarmos col em grid system, mas a diferença é o seguinte: num grid system, temos um limite. Caso o conteúdo do elemento ultrapasse esse limite, ele é quebrado em outra linha. No caso do flex-fill, ele não acontece. Ele prioriza o conteúdo, então dependendo do conteúdo, ele não irá distribuir por igual.

**flex-grow-1:** determina o potencial de crescimento de um elemento dentro da flexbox. Quando um elemento possui essa propriedade, ele cresce em relação aos demais elementos.

**flex-shrink-1:** determina o potencial de encolhimento de um elemento dentro da flexbox.

**flex-wrap:** determina que, quando os itens de uma flexbox estourarem o limite de largura dela, os itens que ultrapassarem serão colocados na linha de baixo.

**align-content-x:** determina o alinhamento dos flex items quando a propriedade flex-wrap está ativada. (x pode ser start, end, center, between e around).

**TIPOGRAFIA**

***Título -* class=”h1 h2 h3 h4 h5 h6”:** deixa o conteúdo textual daquele elemento com o estilo de uma tag h1, ou h2, e assim por diante.

***Parágrafos -* class=”lead”:** diminui o peso da fonte de um parágrafo, deixando-o mais profissional.

***Citações -* class=”blockquote”:** se dentro de uma tag <blockquote>, vai formatar o texto para um citação.

***Alinhamento -* class=”text-center text-left text-right text-justify”:** alinha o conteúdo textual de um elemento.

***Listas –* class=”list-unstyled”:** tira o estilo padrão das listas, que é a bolinha antes de cada item.

***Listas –* class=”list-inline”:** deixa cada elemento da lista com o display “inline-block”, ou seja, eles ficam em linha, um do lado do outro.

***Quebra de texto –* class=”text-wrap”:** permite que o texto do elemento seja quebrado caso ele ultrapasse o limite de largura.

***Quebra de texto –* class=”text-nowrap”:** não permite que o texto do elemento seja quebrado caso ele ultrapasse o limite de largura.

***Quebra de texto –* class=”text-truncate”:** adiciona reticências ao texto caso ele ultrapasse o limite de largura do elemento.

***Caixa de texto –* class=”text-uppercase”:** transforma o conteúdo textual do elemento em caixa alta (maiúsculas).

***Caixa de texto –* class=”text-lowercase”:** transforma o conteúdo textual do elemento em caixa baixa (minúsculas)

***Caixa de texto –* class=”text-capitalize”:** transforma o conteúdo textual do elemento, deixando as iniciais de cada palavra em caixa alta (maiúsculas).

***Peso da fonte –* class=”font-weight-bold”:** transforma o conteúdo textual do elemento em negrito.

***Peso da fonte –* class=”font-weight-light”:** transforma o conteúdo textual do elemento, diminuindo drasticamente o peso da fonte dele.

***Peso da fonte –* class=”font-weight-normal”:** transforma o conteúdo textual do elemento, deixando o peso da fonte dele como normal.

***Decoração do texto* – class=”text-decoration-none”:** remove qualquer tipo de decoração do texto (sublinhado, riscado, etc).

***Decoração do texto –* class=”text-reset”:** reseta configurações padrão de decoração de texto (muito usada em links).

***Tipo da fonte* – class=”font-italic”:** transforma o tipo da fonte do texto para itálico.

***Tipo da fonte –* class=”text-monospace”:** transforma o tipo da fonte do texto para monospace.

**CORES**

**primary: azul escuro**

**secondary: cinza**

**success: verde**

**danger: vermelho**

**warning: amarelo**

**Info: azul claro**

**light: cinza claro**

**dark: cinza escuro**

**body: preto**

**muted: cinza apagado**

**white: branco**

**black: preto**

**white-50** e **black-50:** branco e preto com metade da transparência.

**IMAGENS**

**class=”img-fluid”:** deixa a imagem responsive, fazendo com que ela se adeque a diferentes tamanhos de tela.

**class=”img-thumbnail”:** cria uma borda cinza de 1px em volta da imagem.

**<picture><source></picture> =** tags HTML usadas para definir diferentes resoluções de imagens, para diferentes tamanhos de tela.

**class=”figure”:** usada para determinar uma tag <figure>, que é usada com peso semântico para anexar uma imagem com sua legenda.

**class=”figure-caption”:** usada para dar estilo á legenda de uma imagem, dentro de uma tag <figcaption>.

**TABELAS**

**class=”table”:** permite que o Bootstrap formate aquele elemento como uma tabela.

**class=”table-dark”:** aplica na tabela o modo escuro.

**class=”table-light”:** aplica na tabela o modo claro.

**class=”table-striped”:** aplica na tabela o efeito zebrado.

**class=”table-hover”:** indica a linha da tabela em que o cursor do mouse está passando, ajudando na usabilidade.

**class=”table-sm”:** reduz a altura das linhas da tabela.

**class=”table-color”:** aplica uma cor de fundo na célula, linha, ou tabela em que essa classe for aplicada. No lugar de color, deve-se colocar o nome das cores padrão do Bootstrap (primary, danger, success, etc).

**class=”table-bordered”:** aplica bordas em cada célula da tabela.

**class=”table-borderless”:** remove as bordas das células da tabela.

**ALERTAS**

**class=”alert”:** permite que o Bootstrap formate aquele elemento como um alerta.

**class=”alert-color”:** define uma cor de fundo para o alerta. No lugar de ‘color’, deve-se usar as cores padrão do Bootstrap (primary, danger, success, etc).

**class=”alert-dismissible fade show”:** define que aquele alerta pode ser fechado por meio de um clique em um botão. Esse botão deve possui a classe “close”.

**BOTÕES**

**class=”btn”:** permite que o Bootstrap formate aquele elemento como um botão.

**class=”btn-color”:** aplica uma cor de fundo ao botão. No lugar de ‘color’, deve-se usar as cores padrão do Bootstrap (primary, danger, success, etc).

**class=”btn-outline-color”:** aplica um efeito especial no botão, destacando mais o seu contorno do que o seu fundo.

**class=”btn-lg”:** deixa o botão mais largo.

**class=”btn-sm”:** deixa o botão mais estreito.

**class=”btn-block”:** ativa o display block no botão, fazendo com que ele ocupe a largura inteira do elemento pai.

**class=”active”:** dá o efeito de que o botão está clicado.

**[data-toggle=”button”]:** atributo que aplica o efeito de toggle ao botão, ou seja, quando ele é clicado e está ativo, ele é desativado, e vice-versa.

**BADGES**

**class=”badge”:** permite que o Bootstrap formate aquele elemento como um distintivo.

**class=”badge-color”:** aplica uma cor de fundo ao distintivo. No lugar de ‘color’, deve-se usar as cores padrão do Bootstrap (primary, danger, success, etc).

**class=”badge-pill”:** deixa o distintivo em um formato arredondado, como uma pílula.

**LIST GROUP**

**class=”list-group”:** permite que o Bootstrap formate aquele elemento como um grupo de lista.

**class=”list-group-item”:** define elementos que são itens de um grupo de lista.

**class=”list-group-item-action”:** ativa o efeito de hover no item da lista.

**class=”list-group-item-color”:** aplica uma cor de fundo ao item de uma lista. No lugar de ‘color’, deve-se usar as cores padrão do Bootstrap (primary, danger, success, etc).

**FORMULÁRIOS**

**class=”form-group”:** aplicada no elemento que agrupa as tags <label> e <input> dentro de um formulário.

**class=”form-control”:** aplicada no elemento <input>, <select> ou <textarea>, permite que o Bootstrap formate e reconheça o input.

**class=”form-control-file”:** aplicada no elemento <input> do tipo file, permite que o Bootstrap reconheça e formate esse input.

**class=”form-control-range”:** aplicada no elemento <input> do tipo range, permite que o Bootstrap reconheça e formate esse input.

**class=”form-control-lg”:** aumenta o tamanho da fonte em um input.

**class=”form-control-sm”:** diminui o tamanho da fonte em um input.

**class=”form-control-plaintext”:** exibe um texto fixo, mas como se ele estivesse em um input.

**class=”form-text”:** texto que vem logo depois do input.

**CHECKBOX e RADIO**

**class=”form-check”:** aplicada no elemento que agrupa as tags <label> e <input>, mas apenas inputs que sejam do tipo checkbox ou radio.

**class=”form-check-inline”:** aplicada no elemento que agrupa as tags <label> e <input>, mas apenas inputs que sejam do tipo checkbox ou radio. **Exibe esses elementos em linha.**

**class=”form-check-input”:** aplicada no elemento <input> que seja do tipo checkbox ou radio.

**class=”form-check-label”:** aplicada no elemento <label> que está associado ao input do tipo checkbox ou radio.

**VALIDAÇÃO DE FORMULÁRIOS**

**class=”needs-validation”:** aplicada em uma tag <form>, indica que aquele formulário possuirá um sistema visual intuitivo de validação.

**class=”is-valid”:** força a validação de um input.

**[novalidate]:** atributo aplicado na tag <form> que retira a validação dos campos com o atributo “required”.

**$jQuery = e.target.checkValidity():** e.target se refere ao formulário, e nesse contexto ele está dentro de uma escuta de evento para quando o formulário é submetido. A função checkValidity() faz a validação de todos os campos do formulário. Caso true, todos os campos foram validados com sucesso. Caso false, algum campo está inválido.

**$jQuery = formEl.classList.add('was-validated'):** formEl se refere ao elemento do formulário no DOM. Essa classe adiciona um confere verde ao lado do campo quando ele é validado, e um X vermelho ao lado do campo quando ele está inválido.

**class=”valid-tooltip”:** pequena tooltip que vem logo após um input, acionado quando o input é validado.

**CUSTOMIZAÇÃO DE ELEMENTOS DE FORMULÁRIO**

**class=”custom-control”:** aplicada no elemento que agrupa as tags <label> e <input>, para indicar que aquele input será customizado.

**class=”custom-control-input”:** aplicada na tag <input> para indicar que aquele input será customizado.

**class=”custom-control-label”:** aplicada na tag <label> para indicar que aquele elemento tem um input customizado associado.

**class=”custom-radio”:** aplicada em inputs do tipo radio, para deixar aquele elemento customizado.

**class=”custom-checkbox”:** aplicada em inputs do tipo checkbox, para deixar aquele elemento customizado.

**class=”custom-range”:** aplicada em inputs do tipo range, para deixar aquele elemento customizado.

**class=”custom-select”:** aplicada em inputs do tipo select, para deixar aquele elemento customizado.

**class=”custom-switch”:** aplicada em inputs do tipo checkbox, para deixar aquele input customizado com o formato de um switch.

**INPUT GROUPS (transformam labels em ícones)**

**class=”input-group”:** aplicada no elemento que envolve a tag <input>, para determinar que ali será um input group.

**class=”input-group-prepend”:** aplicada numa div irmã anterior do <input>, que representa o label a ser personalizado.

**class=”input-group-append”:** aplicado numa div irmã posterior do <input>, que representa o label a ser personalizado,

**class=”ïnput-group-text”:** aplicado num elemento <span> ou <button> que se encontra dentro de uma tag com a classe .input-group-prepend ou .input-group-append. Determina o conteúdo do label.

**MEDIA OBJECTS (agrupar imagens com suas descrições)**

**class=”media”:** aplicada no elemento que envolve a imagem e sua descrição.

**class=”media-body”:** aplicada no elemento em que está a descrição da imagem.

**BREADCRUMB**

**(migalha de pão, indica a navegação do usuário pelo site)**

**class=”breadcrumb”:** aplicada geralmente na tag <ul> ou <ol>, no elemento que envolve todos os itens da breadcrumb.

**class=”breadcrumb-item”:** aplicado geralmente na tag <li>, nos elementos itens da breadcrumb.

**NAV**

**(itens de navegação)**

**class=”nav”**

**class=”nav-pills”**

**class=”nav-tabs”:**

aplicada no elemento que envolve todos os links de navegação (geralmente aplicada em uma tag <nav> ou <ul>).

Nav Pills (links que se parecem com pílulas) e Nav Tabs (links com funcionalidades de aba) são outros estilos para os componentes de navegação.

**É possível amarrar cada aba dos links de navegação com seus respectivos conteúdos. Veja a documentação.**

**class=”nav-link”:** aplicada nos elementos que possuem os links de navegação (geralmente aplicada em uma tag <a>).

**COLLAPSE**

**(exibir ou ocultar um conteúdo a partir do clique de um botão)**

**data-toggle=”collapse”:** atributo que vai no botão, determinando que ele irá exibir ou ocultar um conteúdo a partir do clique.

**data-target=”#id”:** atributo que vai no botão, apontando qual o id do elemento que deve ser exibido/ocultado.

**class=”collapse”:** aplicada no elemento que vai ser exibido/ocultado, indicando que ele depende do clique de um botão na página para ser exibido ou ocultado.

**CARDS**

**(pequenos cartões)**

**class=”card”:** aplicada no elemento que será o card.

**class=”card-header”:** aplicada no elemento que determina o cabeçalho do cartão.

**class=”card-title”:** aplicada no elemento que será o título do cartão.

**class=”card-body”:** aplicada no elemento que será o corpo do cartão.

**class=”card-img-top”:** aplicada na imagem do cartão, que geralmente fica acima do texto.

**class=”card-img-overlay”:** aplicada no elemento que envolve o título e o texto do cartão, e determina que esses itens ficarão dentro da imagem do cartão.

**class=”card-text”:** aplicada no elemento em que se encontra o texto do cartão.

**class=”card-footer”:** aplicada no elemento que será o rodapé do cartão.

**class=”card-link”:** aplicada em uma tag <a> ou <button> dentro do cartão, para determinar que aquele é um botão do cartão.

**class=”card-group”:** aplicada para agrupar vários cards em um único card.

**ACCORDION**

**(exibir elementos de uma lista, um por vez)**

**class=”accordion”:** aplicada no elemento pai, que reunirá os itens e será o accordion. Esse elemento **deve** ter um id.

**data-parent=”#accordion”:** aplicada no elemento que será exibido/ocultado após o clique de um botão. Esse atributo amarra aquele item ao accordion, que é seu elemento pai.

**data-toggle=”collapse”:** atributo que vai no botão, determinando que ele irá exibir ou ocultar um conteúdo a partir do clique.

**data-target=”#id”:** atributo que vai no botão, apontando qual o id do elemento que deve ser exibido/ocultado.

**class=”collapse”:** aplicada no elemento que vai ser exibido/ocultado, indicando que ele depende do clique de um botão na página para ser exibido ou ocultado.

**NAVBAR**

**(barra de navegação)**

**class=”navbar”:** aplicada no elemento pai, que será a barra de navegação.

**class=”navbar-light navbar-dark”:** aplicada no elemento pai, determina se a barra de navegação possuirá tema claro ou escuro.

**class=”navbar-expand-lg”:** aplicada no elemento pai, determina que em telas grandes, a barra de navegação ficará expandida.

**class=”navbar-brand”:** aplicada no elemento em que está a logo da empresa, que ficará em destaque na barra de navegação.

**class=”navbar-toggler”:** aplicada no botão que exibirá/ocultará os itens da barra de navegação em telas menores.

**class=”navbar-toggler-icon”:** aplicada numa tag <span> que se encontra dentro do botão com a classe “navbar-toggler”, serve para aplicar o ícone de sanduíche ao botão.

**data-toggle=”collapse”:** atributo que vai no botão, determinando que ele irá exibir ou ocultar um conteúdo a partir do clique.

**data-target=”#id”:** atributo que vai no botão, apontando qual o id do elemento que deve ser exibido/ocultado.

**class=”navbar-collapse collapse”:** aplicada no elemento que envolve a lista com os itens da barra de navegação. Essas classes determinarão que a lista será exibida/ocultada a partir do clique de um botão com data-toggle=”collapse”. Esse elemento **deve** possuir um id.

**class=”navbar-nav”:** aplicada na tag <ul>, determinando a lista de itens da barra de navegação.

**class=”nav-item”:** aplicada nas tags <li>, determinando os itens da barra de navegação.

**class=”nav-link”:** aplicada nas tags >a>, determinando os links da barra de navegação.

**PAGINATION**

**(paginação dos dados)**

**class=”pagination”:** aplicada na tag <ul>, determina o elemento que envolverá os itens da paginação.

**class=”page-item”:** aplicada nas tags <li>, determinam os itens da paginação.

**class=”page-link”:** aplicada nas tags <a>, determinam os links da paginação.

**class=”active”:** determina que aquele item é o ativo, o atual.

**class=”disabled”:** determina que aquele item está desabilitado.

**POPOVER**

**(botões com dicas)**

**data-toggle=”popover”:** atributo que vai no botão, determinando que ele usará o plugin Popover.

**data-content=”conteúdo”:** atributo que vai no botão, determinando o conteúdo da mensagem do popover.

**data-placement=”top”:** atributo que vai no botão, determina de que lado a mensagem do popover será exibida em relação ao botão (aceita top, right, bottom, left).

**data-trigger=”hover”:** atributo que vai no botão, determina que o popover será ativado assim que o usuário passar o mouse pelo botão.

**title=”Título”:** atributo que vai no botão, atribui um título na mensagem do popover.

**É necessário selecionar todos os elementos com o atributo data-toggle=”popover” e invocar a função popover(), usando jQuery. Isso irá habilitar o plugin.**

**PROGRESS**

**(barra de progresso)**

**class=”progress”:** aplicada no elemento que vai envolver a barra de progresso.

**class=”progress-bar”:** aplicada no elemento que será a barra de progresso. A largura dessa barra de progresso poderá ser manipulada para causar o efeito de loading desejado.

**class=”progress-bar-striped”:** aplicada no elemento que será a barra de progresso, mas causa um efeito listrado na barra.

**class=”progress-bar-animated”:** aplicada no elemento que será a barra de progresso, onde junto com a classe striped, causará o efeito de listras animadas.

**SCROLLSPY**

**(navegação com scroll)**

**data-spy=”scroll”:** atributo que vai no elemento que terá scroll, determinando que o scrollspy será utilizado nesse elemento.

**data-target=”#id”:** atributo que vai no elemento que terá scroll, determinando o elemento navbar ou list que será usado como referência para o scrollspy.

**SPINNER**

**(ícones de carregamento/loading)**

**class=”spinner-border”:** atribuída ao elemento que será o ícone de loading. O estilo do ícone será as bordas de uma bolinha rodando.

**class=”spinner-grow”:** atribuída ao elemento que será o ícone de loading. O estilo do ícone será uma bolinha piscando.

**\* É possível adicionar os prefixos -lg e -sm nas classes dos spinners para aumenta-los ou diminuí-los, respectivamente.**

**\*\* É recomendado adicionar um <span> com a classe “sr-only” para que leitores de tela possam identificar que aquilo é um ícone de carregamento.**

**TOASTS**

**(notificações flutuantes)**

**class=”toast”:** atribuída ao elemento que será o toast.

**class=”toast-header”:** atribuída ao elemento que será o cabeçalho do toast.

**class=”toast-body”:** atribuída ao elemento que será o conteúdo do toast.

**É necessário selecionar todos os elementos com a classe “toast” e invocar a função toast(‘show’), usando jQuery. Isso irá exibir o componente.**

**MODAL**

**(caixa de diálogo)**

**class=”modal”:** atribuída ao elemento que será o modal.

**\* É possível adicionar os profixos sm, lg e xl na classe “modal” para ajustar o tamanho do modal.**

**class=”modal-dialog”:** atribuída ao elemento filho direto do modal.

**class=”modal-dialog-scrollable”:** atribuída ao elemento filho direto do modal, determina que o modal vai possuir scroll caso o conteúdo dele ultrapasse o limite de sua altura.

**class=”modal-dialog-centered”:** atribuída ao elemento filho direto do modal, determina que o modal ficará centralizado na tela.

**class=”modal-content”:** atribuída ao elemento filho direto de modal-dialog, determina o conteúdo do modal.

**class=”modal-header”:** determina o cabeçalho do modal.

**class=”modal-title”:** determina o título do modal.

**class=”modal-body”:** determina o corpo do modal.

**class=”modal-footer”:** determina o rodapé do modal.

**data-dismiss=”modal”:** atributo que vai no botão responsável por fechar o modal ao ser clicado.

**Para o botão que vai abrir o modal:**

**data-toggle=”modal”:** atributo que vai no botão responsável por abrir o modal.

**data-target=”#id-modal”:** atributo que vai no botão, determina o modal que será aberto após o clique do botão (o modal é obtido através do id).

**TOOLTIP**

**(ferramenta de dica)**

**data-toggle=”tooltip”:** atributo que vai no botão que terá o tooltip.

**title=”titulo”:** atributo que vai no botão, determinando o conteúdo do tooltip.

**data-placement=”top”:** atributo que vai no botão, determina de que lado a mensagem do popover será exibida em relação ao botão (aceita top, right, bottom, left).

**É necessário selecionar todos os elementos com o atributo data-toggle=”tooltip” e invocar a função tooltip(), usando jQuery. Isso irá habilitar o plugin.**