Migração para a v4

Bootstrap 4 é uma grande aprimoração de todos os projetos anteriores. As melhorias mais notáveis estão categorizadas abaixo, de mudanças mais específicas até os componentes mais relevantes.

Mudanças estáveis

Da versão Beta 3 até o nosso release estável 4.0, não tem nenhuma incrível mudança, mas algumas notáveis.

Impressão

- Consertamos utilitários de impressão. Anteriormente, usando uma classe `.d-print-*`, sobrescreveríamos qualquer outra classe `.d-*`. No entanto, agora, é possível combinar estes utilitários e o efeito só atinge a determinada media (`@media print`);
- Expandimos a quantidade disponível de utilitários de impressão, para termos mais combinações destes utilitários. A Beta 3 e versões anteriores apenas tinham as variantes `block`, `inline-block`, `inline` e `none`. A versão 4 estável acrescentou `flex`, `inline-flex`, `table`, `table-row` e `table-cell`;
- Corrigimos a renderização da pré-visualização de impressão nos browsers, usando novos estilos que especificam `@page size`.

Mudanças do Beta 3

Apesar da versão Beta 2 ter recebido grandes mudanças, ainda havia algumas poucas que precisavam serem adicionadas, através da Beta 3. Estas mudanças são visíveis, se você está atualizando da Beta 2 (ou qualquer outra versão mais antiga) para a Beta 3.

Miscelânea

- Variável `$thumbnail-transition` era pouco utilizada, portanto, foi removida. Não estávamos passando por nenhuma transição, então, era só código desnecessário;
- O pacote npm não inclui mais nenhum arquivo, além dos arquivos fonte e de distribuição. Se você dependia deles e estava executando nossos scripts através do diretório `node_modules`, deve adaptar seu workflow.

Formulários

- Reescrevemos os checkboxes e radio buttons, tanto os padrões quanto os personalizados. Agora, ambos possuem estruturas HTMLs iguais (`<div>` externa com os elementos irmãos: `<input>` e `<label>`), além do mesmos estilos de layout (empilhado por padrão e inline usando classes modificadoras). Isso nos permitiu estilizar a label baseando-se no estado do input, simplificando o suporte para o atributo `disabled` (exigia uma classe pai, anteriormente) e melhorando o suporte à validação de formulário.

Como parte disso, nós mudamos o CSS para gerenciar múltiplas `background-image`s em checkboxes e radios buttons personalizados. Anteriormente, o agora removido elemento `.custom-control-indicator`, tinha background color, gradiente e ícones SVG. Personalizar o grandiente de backround significava substituir tudo aquilo, toda vez que você precisava alterar apenas um. Agora, nós temos `.custom-control-label::before` para o preenchimento e gradiente, além do `.custom-control-label::after` para lidar com ícones.

Para fazer um checkbox inline personalizado, use `.custom-control-inline`.

- O seletor para grupos de botões baseados em input foi atualizado. Invés de `[data-toggle="buttons"] { }` para estilo e comportamento, nós usamos o atributo `data` só para os comportamentos JavaScript e a classe `.btn-group-toggle` para estilizar;
- A classe `.col-form-legend` foi removida, em prol de outra um pouco melhor: `.col-form-label`. Assim, as classes `.col-form-label-sm` e `.col-form-label-lg` podem serem utilizadas em elementos `<legend>`, com facilidade;
- Inputs de arquivos personalizados receberam mudanças em suas variáveis Sass `$custom-file-text`. Agora, não são mapas Sass aninhados e só afetam uma linha (o botão `Browse`, já que que ele é o único pseudo-elemento gerado por nosso Sass). O texto `Choose file` agora é originado da classe `.custom-file-label`.

Grupos de input

- Addons de grupos de inputs agora são específicos para seus posicionamentos, em relação a um input.
  - Nós defasamos `.input-group-addon` e `.input-group-btn` em prol de duas novas classes: `.input-group-prepend` e `.input-group-append`.
    - Agora, você deve escolher entre as variantes append ou prepend, simplificando o CSS.
    - Dentro de um append ou prepend, use botões assim como faria em qualquer outro lugar, mas textos devem ser colocados dentro de elementos com a classe `.input-group-text`;
- Estilos de validação agora são suportados, assim como múltiplos inputs (apesar de ser possível validar apenas um input, por grupo);
- Classes para dimensionamento devem ser colocadas no `.input-group` pai e, não no elemento de formulário individual.

Mudanças da Beta 2

Durante a beta, não focamos em nenhuma mudança extraordinária. No entanto, as coisas nem sempre são como planejadas. Abaixo estão grandes alterações para se ter em mente, quando migrar da Beta 1 para Beta 2.

Grandes mudanças

- Variável `$badge-color` e seu uso em `.badge` foi removido;
  - Nós usamos uma função de contraste de cores para acharmos uma `cor` específica, baseando-se na `background-color`, então, a variável é desnecessária.
- Renomeamos a função `grayscale()` para `gray()`, assim evitando conflitos com o filtro CSS nativo `grayscale`;
- Renomeamos `.table-inverse`, `thead-inverse` e `.thead-default` para `.*-dark` e `.*-light`, combinando com nosso esquema de cores usado no resto do framework;
- Tabelas responsivas agora possuem classes para cada grid breakpoint;
  - Isso foi pensado depois do Beta 1, quando a classe `.table-responsive` que você usava tava mais para `.table-responsive-md`;
  - Agora, você você pode usar `.table-responsive` ou `.table-responsive-{sm,md,lg,xl}`, como quiser.
- Largamos mão do suporte ao Bower, já que o gerenciador de pacotes está defasado, em relação à alternativas como Yarn ou npm;
  - Veja o issue bower/bower#2298, para mais detalhes.
- Bootstrap ainda requer jQuery 1.9.1 ou maior, mas é aconselhado usar a versão 3.x, já que os browsers suportados por esta versão são os mesmos que o Bootstrap suporta, sem falar que também possui algumas correções de segurança;
- Removemos a classe pouco utilizada `.form-control-label`.
  - Se é que você usava esta classe, saiba que ela era apenas uma cópia da `.col-form-label` que centralizava uma `<label>` verticalmente com seu input associado, em formulários de layout horizontal;
- Alteramos o `color-yiq`, de um mixin que usava a proprieade `color` para uma função que retorna um valor, permitindo que você use ele em qualquer propriedade CSS.
  - Por exemplo, invés de `color-yiq(#000)`, você pode escrever `color: color-yiq(#000);`.

Destaques

- Foi introduzido novos usos de `pointer-events`, em modais.
  - O `.modal-dialog` mais externo possui `pointer-events: none` para que possamos lidar com cliques customizados (isso faz com que só seja possível esperar por cliques no `.modal-backdrop`) e ele seja neutralizado para o próprio `.modal-content` que possui `pointer-events: auto`.

Sumário

Aqui estão as coisas mais importantes que você deve estar atento, quando migrando da v3 para v4.

Suporte de navegadores

- Largou-se mão de suporte ao IE8, IE9 e iOS 6;
  - A v4 só suporta IE10+ e iOS 7+, agora;
  - Para sites que precisam de qualquer uma destas versões não suportadas, use a v3.
- Adicionou-se suporte oficial para o WebView e navegador do Android v5.0 (Lollipop).
  - As versões mais antigas do navegador Android e do WebView continuam suportadas não oficialmente, apenas.

Mudanças globais

- **Flexbox está ativado por padrão.**
  - Em resumo, isto significa que estamos um passo mais longe de floats e outro mais próximo dos nossos componentes;
- Mudamos o pré-processador dos nossos arquivos CSS fonte, do Less para o Sass;
- Deixamos de usar `px` em troca de `rem` como nossa unidade CSS primária (apesar de pixel ainda ser usado para media queries e comportamento de grids, já que viewport de dispositivos não são afetadas por tipo de unidade);
- A font-size global aumentou de `14px` para `16px`;
- Reformamos a graduação do sistema de grades para adicionar uma quinta opção (colocando os menores dispositivos de `576px` para baixo) e removemos o infixo `-xs` destas opções;
  - Exemplo: `.col-6.col-sm-4.col-md-3`.
- Substituímos o tema opcional por opções configuráveis em variáveis SCSS (ex: `$enable-gradients: true`);
- Sistema de build reformado para usar vários scripts npm, invés do Grunt;
  - Veja o `package.json` para ver todos os scripts ou nosso readme do projeto para instruções de desenvolvimento local;
- Uso não responsivo do Bootstrap não é mais suportado;
- Largou-se mão do Customizer online, em prol de uma documentação de configuração mais extensa e build personalizada;
- Adicionou-se várias classes utilitárias novas para pares comuns de propriedade-valor e atalhos para espaçamento de margem e padding.

Grid system

- **Migrou para o flexbox.**
  - Criou-se suporte para flexbox nos mixins de grid e classes pré-definidas;
  - Como parte do flexbox, também foi incluído classes para alinhamento horizontal e vertical.
- **Nome de classes e graduação do sistema de grades foram atualizados.**
  - Adicionamos um novo grau `sm` abaixo de `786px` para um controle mais granular;
    - Agora, nós temos `xs`, `sm`, `md`, `lg` e `xl`.
      - Isso também significa que cada grau subiu um degrau, então, `.col-md-6` na v3 é equivalente a `.col-lg-6` na v4.
  - As classes do grau `xs` foram modificadas de forma que não seja necessário usar o determinado infixo (`xs`), isso para representar, apropriedamente, que elas começam a aplicar estilos em `min-width: 0` e não em um valor diferente.
    - Invés de `.col-xs-6`, agora é `.col-6`. No entanto, todos os outros graus exigem que você use infixos.
- **Tamanhos, mixins e variáveis foram atualizadas.**
  - As gutters do grid agora possuem um mapa Sass para que você possa especificar determinados tamanhos de larguras, em cada breakpoint;
  - Os mixins do grid foram atualizados para utilizarem o mixin `make-col-ready` como base e outro mixin `make-col` para definir `flex` e `max-width` para o tamanho individual de colunas;
  - Alteramos os breakpoints da media query do sistema de grid e larguras dos containers para levar em conta a nova graduação do grid e garantir que todas as colunas estão bem divididas por `12`, na sua largura máxima;
  - Os breakpoints e containers do grid agora são manipulados via mapas Sass (`$grid-breakpoints` e `$container-max-widths`), invés de um punhado de variáveis separadas;
    - Estes mapas substituem as variáveis `@screen-*`, totalmente, e permitem que você customize a graduação do grid.
  - As media queries também mudaram.
    - Invés de ficar repetindo nossas declarações de media queries com o mesmo valor, todas às vezes, nós temos `@include media-breakpoint-up/down/only`, agora. Portanto, invés de escrever `@media (min-width: @screen-sm-min) { ... }`, você pode digitar `@include media-breakpoint-up(sm) { ... }`.

Componentes

- **Deixamos de lado os panels, thumbnails e wells** por um componente mais robusto: o card;
- **Largamos mão da fonte de ícones Glyphicons.**
  - Se você precisa de ícones, aqui estão algumas opções:
    - O próprio site do Glyphicons;
    - Octicons;
    - Font Awesome;
    - Veja a página Extend, para ver uma lista de alternativas;
    - Tem alguma sugestão? Abra um issue, por favor.
- **Desistimos do plugin jQuery Affix;**
  - No lugar, recomendamos usar `position: sticky`;
    - Veja esta página do HTML5 Please para detalhes e recomendações de polyfills específicos;
    - Uma sugestão é usar uma regra `@supports` para implementar isso (ex: `@supports (position: sticky) { ... }`).

    - Se você está usando o Affix para aplicar estilos adicionais que não são derivados de `position`, os polyfills podem não ser adequado a você. Uma alternativa para tais casos é a biblioteca ScrollPos-Styler.
- **Também desistimos do componente pager**, já que era um botão customizado, praticamente;
- **Refizemos quase todos componentes** para usar mais seletores de classes não aninhados, invés de seletores filhos específicos.

Destaques em componentes

Esta lista destaca mudanças chaves em componentes, entre a v3.x.x e v4.0.0.

Reboot

Novo no Bootstrap 4, Reboot é uma folha de estilos construída com base no Normalize e em nossa opinião sobre o que é um bom reset. Seletores que aparecem neste arquivo só usam elementos, portanto, não há nenhum seletor de classe. Isso separa nosso reset dos componentes, criando uma abordagem mais modular. Um dos resets mais importantes que ele inclui são o `box-sizing: border-box`, uso da unidade `rem` invés de `em` em muitos dos elementos, estilos de links e muitos outros resets para elementos de formulários.

Tipografia

- Moveu-se todos os utilitários `.text-` para o arquivo `_utilitários.scss`;
- Desistimos da `.page-header`, já que seus estilos podem ser obtidos usando utilitários;
- Também deixamos a classe `.dl-horizontal` de lado;
    - Invés dela, use `.row` no `<dl>` e classes de coluna do grid (ou mixins) em seus filhos `<dt>` e `<dd>`.
- Os blocos de citações foram redesenhados, movendo seus estilos do elemento `<blockquote>` para a classe `.blockquote` e trocando a classe modificadora `.blockquote-reverse` pelo uso de utilitários de texto;
- Agora, a `.list-inline` exige que seus itens filhos usem a classe `.list-inline-item`.

Imagens

- Renomeamos `.img-responsive` para `.img-fluid`;
- Renomeamos `.img-rounded` para `.rounded`;
- Renomeamos `.img-circle` para `.rounded-circle`.

Tabelas

- Quase todas as aparições do seletor `>` foram removidas, significando que tabelas aninhadas vão herdar estilos de seus pais, automaticamente;
    - Isso simplifica nossos seletores e potenciais customizações.
- Renomeamos `.table-condensed` para `.table-sm`, assegurando consistência;
- Adicionamos uma nova opção `.table-inverse`;
- Adicionamos modificadores para cabeçalhos de tabelas: `.thead-default` e `.thead-inverse`;
- Renomeamos classes contextuais para terem um prefixo `.table-`. Assim, as classes `.active`, `.success`, `.warning`, `.danger` e `.info` se transformam em `.table-active`, `.table-success`, `.table-warning`, `.table-danger` e `.table-info`.

Formulários

- Movemos elementos de resets para o arquivo `_reboot.scss`;
- Renomeamos `.control-label` para `.col-form-label`;
- Renomeamos `.input-lg` e `.input-sm` para `.form-control-lg` e `.form-control-sm`, respectivamente;
- Deixamos as classes `.form-group-*` de lado, em nome da simplicidade;
    - Use as classes `.form-control-*`, invés.
- Desistimos da class `.radio-block` e a substituímos com `.form-text`, para textos de ajuda em block-level;
    - Para textos de ajuda em inline-level e outras opções flexíveis, use classes utilitárias como `.text-muted`.
- Desistimos de `radio-inline` e `checkbox-inline`;
- Consolidamos `.checkbox` e `.radio` em `.form-check` e nas várias classes`.form-check-*`;
- Revisamos formulários horizontais:
    - Removemos a exigência da classe `.form-horizontal`;
    - A classe `.form-group` não recebe mais estilos da `.row`, via mixin. Assim, o uso da `.row` é obrigatório para layouts de grid horizontais, como: `<div class="form-group row">`;
    - Adicionamos uma nova classe `.col-form-label` para centralizar labels com `.form-control`, verticalmente;

    - Adicionamos a nova `.form-row` para layouts de formulários compactos com as classes do grid (em resumo, basta trocar `.row` por `.form-row`).
- Adicionamos suporte a formulários customizados (checkboxes, radios, selects e input de arquivos);
- Substituímos `.has-error`, `.has-warning` e `.has-success` com validação de formulários HTML5 via pseudo-seletores `:invalid` e `:valid`;
- Renomeamos `.form-control-static` para `.form-control-plaintext`.

Botões

- Renomeamos `.btn-default` para `.btn-secondary`;
- Desistimos da classe `.btn-xs` já que `.btn-sm` é, proporcionalmente, muito menor que a da v3;
- O recurso do plugin jQuery `button.js`, stateful button foi largado de mão;
    - Isso também inclui os métodos `$().button(string)` e `$().button('reset')`;
    - Nós aconselhamos usar um pouco de JavaScript customizado, invés, para te dar o benefício de fazer funcionar como você quiser;
    - Perceba que os outros recursos do plugin (botões de alternância, checkboxes, radios) foram preservados.
- Mudamos os estados dos botões de `[disabled]` para `:disabled`, já que este último é suportado por IE9+.
    - Contudo, `fieldset[disabled]` ainda é necessário porque fieldsets desativados nativamente ainda são problemáticos no IE11.

Grupo de botões

- O componente foi reescrito com flexbox;
- Removemos `.btn-group-justified`;
    - Alternativamente, você pode usar `<div class="btn-group d-flex" role="group"></div>` como um envolto em elementos com `.w-100`.
- Desistimos da classe `.btn-group-xs`, dado a remoção do `.btn-xs`;
- Removemos o espaçamento entre grupos de botões, nas barras de botões;
    - Use utilitários de margem, agora.
- Aperfeiçoamos a documentação sobre o uso do grupo de botões com outros componentes.

Dropdowns

- Mudamos de seletores pais para classes em todos os componentes, modificadores, etc;
- Simplificamos os estilos do dropdown para nunca mais se apresentar com flechas para cima ou baixo, fixadas no menu dropdown;
- Dropdowns podem ser construídos com `<div>` ou `<ul>`, agora;
- Reformamos os estilos dos dropdowns para prover um fácil suporte para aqueles baseados em itens com `<a>` ou `<button>`;
- Renomeamos `.divider` para `.dropdown-divider`;
- Os itens dropdown agora necessitam da classe `.dropdown-divider`;
- O botão dropdown não precisa mais de `<span class="caret"></span>`;
    - Isso é feito automaticamente, usando o pseudo-elemento `::after` no `.dropdown-toggle`, agora.

Sistema grid

- Adicionamos um novo breakpoint grid de `576px` como `sm`, significando que agora há cinco graus: `xs`, `sm`, `md`, `lg` e `xl`;
- Renomeamos as classes modificadoras responsivas de `.col-{breakpoint}-{modificador}-{tamanho}` para `.{modificador}-{breakpoint}-{tamanho}`, simplificando as classes do grid;
- Desistimos das classes modificadoras push e pull, em prol das novas classes baseadas na propriedade flexbox `order`;
    - Invés de `.col-8.push-4` e `.col-4.pull-8`, você usaria `.col-8.order-2` e `.col-4.order-1`.
- Adicionamos classes utilitárias flexbox para o sistema grid e componentes.

Grupo de listas

- Reescrevemos o componente, usando flexbox;
- Substituímos `a.list-group-item` por uma classe `.list-group-item-action`, para estilizar versões de link e botões dos grupos de listas;
- Adicionamos a classe `.list-group-flush` para uso com cards.

Modal

- Reescrevemos o componente, usando flexbox;
- Dado a reescrita com flexbox, o alinhamento dos ícones de dispersão no cabeçalho estão meio bugados, já que não usamos mais floats;
    - Coloque seus ícones de dispersão depois dos títulos dos modais, para evitar isso.

- A opção `remote` (a qual é usada para carregar e injetar conteúdo em um modal, automaticamente) e o evento correspondente `loaded.bs.modal` foram removidos.
  - Nós recomendamos usar *client-side templating*, um *framework de data binding* ou invocar o jQuery.load, você mesmo.

Navs

- Reescrevemos o componentes com flexbox;
- Desistimos de quase todos os seletores `>`, para uma estilização simplificada através de classes não aninhadas;
- Invés de seletores HTML específicos como `.nav > li > a`, nós usamos classes separadas como `.nav`, `.nav-item` e `.nav-link`.
  - Isso faz seu HTML ser mais flexível, enquanto traz maior extensibilidade.

Navbar

O navbar foi, inteiramente, reescrito usando flexbox com suporte melhorado à alinhamento, responsividade e customização.

- Comportamentos navbar responsivos agora são aplicados a classe `.navbar`, através do **obrigatório** `.navbar-expand-{breakpoint}`, onde você escolhe expandir o navbar;
  - Anteriormente, isso era feito com uma modificação em variável Less e requeria recompilação.
- A classe `.navbar-default` agora é `.navbar-light`, mas `.navbar-dark` continua a mesma;
  - **Uma dessas classes deve ser usada, num navbar.**
    - No entanto, estas classes não definem mais a cor do background (`background-color`), mas apenas a cor dos caracteres (`color`).
- Navbars agoram precisam de um declaração de background de algum tipo;
  - Escolha entre nossos utilitários de background (`.bg-*`) ou defina o seu próprio com as classes light/inverse acima para uma customização maluca.
- Dado aos estilos flexbox, navbars podem usar utilitários flexbox para fácil alinhamento, agora;
- `.navbar-toggle` agora é `.navbar-toggler`, tem diferentes estilos e uma marcação HTML interna (sem mais tanto `<span>`!);
- Desistimos da classe `.navbar-form`, totalmente;
  - Ela não é mais necessária. Por isso, basta usar `.form-inline` e aplicar utilitários de margem, o quanto quiser.
- Navbars não usam mais `margin-bottom` ou `border-radius`, por padrão;
  - Use utilitários, se necessário.
- Todos os exemplos relacionados a navbars foram atualizados para receberem nova marcação HTML.

Paginação

- Reescrevemos o componente, usando flexbox;
- Classes explícitas (`.page-item` e `.page-link`) agora são obrigatórias, em descendentes de um `.pagination`;
- Desistimos do `.pager`, já que ele nada mais era do que um botão outline customizado.

Breadcrumbs

- A classe `.breadcrumb-item` é obrigatória em descendentes do `.breadcrumb`, agora.

Labels e badges

- Consolidamos `.label` e `.badge` para diferenciar do elemento `<label>` e simplificar componentes relacionados;
- Adicionamos `.badge-pill` como sendo uma classe modificadora para formato arredondado, semelhante a uma pílula;
- Os badges não são mais flutuados automaticamente, em grupos de listas e outros componentes;
  - Classes utilitárias são necessárias para isso.
- A `.badge-default` foi defasada e `.badge-secondary` adicionada para combinar com classes modificadoras de outro componente.

Panels, thumbnails e wells

Todos defasados, em prol do novo componente: card.

Panels

- Repensado de `.panel` para `.card`, usando flexbox;
- `.panel-default` removida e sem substituta;
- `.panel-group` removida e sem substituta.
  - `.card-group` não é uma classe substituta, pois, é diferente.
- `.panel-heading` repensado para `.card-header`;
- `.panel-title` refeito como sendo `.card-title`;

    - Dependendo do visual desejado, você também pode querer usar elementos ou classes de cabeçalho (ex: `<h3>` e `.h3`), elementos negritos ou outras classes (ex: `<strong>`, `<b>` e `.font-weight-bold`);
    - Perceba que `.card-title`, apesar do nome ser similar, tem um resultado visual diferente do `.panel-title`.
- `.panel-body` refeito em `.card-body`;
- `.panel-footer` repensado como `.card-footer`;
- `.panel-primary`, `.panel-success`, `.panel-info`, `.panel-warning` e `.panel-danger` foram defasados, em troca dos utilitários `.bg-`, `.text-` e `.border` gerados à partir do nosso mapa Sass `$theme-colors`.

Progresso

- Substituídas as classes `.progress-bar-*` contextuais por utilitários `.bg-*`;
    - Exemplo: `class="progress-bar progress-bar-danger"` se torna `class="progress-bar bg-danger"`.
- Substituída a classe `.active` para barras de progressos animadas por `.progress-bar-animated`.

Carousel

- Revisado todo o componente para simplificar o estilo e design;
    - Agora temos menos estilos para você sobrescrever, novos indicadores e ícones.
- Todo o CSS foi desaninhado e renomeado, garantindo que cada classe tenha o prefixo `.carousel-`;
    - Os itens do carousel: `.next`, `.prev`, `.left` e `.right` agora são `.carousel-item-next`, `.carousel-item-prev`, `.carousel-item-left` e `.carousel-item-right`;
    - Os controles prev/next: `.carousel-control.right` e `.carousel-control.left` agora são `.carousel-control-next` e `.carousel-control-prev`, significando que eles não precisam mais de um classe base.
- Removido todo estilo responsivo, jogando a responsabilidade para utilitários (ex: mostrar legendas em certas viewports) e estilos customizados, quando necessário;
- Removido sobrescrição de imagens nos itens do carousel, deixando isso para utilitários;
- Arrumamos o exemplo de Carousel para que tenha a nova marcação HTML e estilos CSS.

Tabelas

- Removido o suporte à tabelas aninhadas estilizadas;
    - Todos estilos de tabelas agora são herdados, na v4, para simplificar os seletores.
- Adicionado a tabela inversa.

Utilitários

- **Display, hidden e outros:**
    - Fizemos os utilitários display serem responsivos;
        - Exemplo: `.d-none` e `d-{sm,md,lg,xl}-none`.
    - Desistimos da maioria dos utilitários `.hidden-*`, em troca de novos utilitários display;
        - Invés de `.hidden-sm-up`, use `.d-sm-none`;
        - Renomeamos os utilitários `.hidden-print` para que usem o esquema de nomenclatura dos utilitários display;
        - Saiba mais, na seção de utilitários responsivos, nesta página.
    - Adicionamos as classes `.float-{sm,md,lg,xl}-{left,right,none}` para flutuações responsivas, além de removermos `.pull-left` e `.pull-right` porque são redundâncias de `.float-left` e `.float-right`.
- **Tipo:**
    - Adicionamos variações responsivas para nossas classes de alinhamento de texto `.text-{sm,md,lg,xl}-{left,center,right}`.
- **Alinhamento e espaçamento:**
    - Adicionamos novos utilitários responsivos de margem e padding para os quatro lados, além de atalhos horizontais e verticais;
    - Adicionamos um punhado de utilitários flexbox;
    - Desistimos da `.center-block`, em troca da nova classe `.mx-auto`.
- O clearfix foi atualizado para não dar mais suporte a versões antigas de browsers.

Mixins para prefixos de browsers

Os mixins para prefixos de browsers, no Bootstrap 3, os quais foram defasados na v3.2.0, foram removidos no Bootstrap 4. Já que usamos o Autoprefixer, eles não são necessários.

Removemos os seguintes mixins: `animation`, `animation-delay`, `animation-direction`, `animation-duration`, `animation-fill-mode`, `animation-iteration-count`, `animation-name`, `animation-timing-function`, `backface-visibility`, `box-sizing`, `content-columns`, `hyphens`, `opacity`, `perspective`, `perspective-origin`, `rotate`, `rotateX`, `rotateY`, `scale`, `scaleX`, `scaleY`, `skew`, `transform-origin`, `transition-delay`, `transition-duration`,

`transition-property`, `transition-timing-function`, `transition-transform`, `translate`, `translate3d` e `user-select`.

Documentação

Nossa documentação recebeu um upgrade, também. Segue as novidades:

- Nós continuamos usando Jekyll, mas também usamos plugins:
  - `bugify.rb` é usado para listar, eficientemente, nosso registro na página browser bugs;
  - `example.rb` é um *fork* personalizado do plugin `highlight.rb`, permitindo uma manipulação do código-exemplo, mais fácil;
  - `callout.rb` também é um *fork* personalizado do plugin, mas pensado para nossas chamadas especiais nos documentos;
  - O jekyll-toc é usado para gerar nossa tabela de conteúdos.
- Todo conteúdo dos documentos foram reescritos em Markdown (invés de HTML) para fácil edição;
- Páginas foram reorganizadas para um conteúdo mais simples e hierarquia mais acessível;
- Migramos do CSS ao SCSS para termos toda a vantagem das variáveis, mixins e outras coisas do Bootstrap.

Utilitários responsivos

Todas variáveis `@screen-` foram removidas na v4.0.0. Use os mixins Sass `media-breakpoint-up()`, `media-breakpoint-down()`, `media-breakpoint-only()` ou o mapa `$grid-breakpoints`, invés.

Nossos utilitários responsivos foram removidos, em prol de utilitários `display`.

- As classes `.hidden` e `.show` foram removidas porque entravam em conflito com os métodos `$(...).hide()` e `$(...).show()` do jQuery. Invés delas, tente alternar o atributo `[hidden]` ou usar estilos inline como `style="display: none;"` e `style="display: block;"`.
- Todas as classes `.hidden-` foram removidas, exceto pelos utilitários de impressão, os quais só renomeamos;
  - Removemos da v3: `.hidden-xs`, `.hidden-sm`, `.hidden-md`, `.hidden-lg`, `.visible-xs-block`, `.visible-xs-inline`, `.visible-xs-inline-block`, `.visible-sm-block`, `.visible-sm-inline`, `.visible-sm-inline-block`, `.visible-md-block`, `.visible-md-inline`, `.visible-md-inline-block`, `.visible-lg-block`, `.visible-lg-inline` e `.visible-lg-inline-block`.
  - Removemos das v4 alphas: `.hidden-xs-up`, `.hidden-xs-down,`, `.hidden-sm-up`, `.hidden-sm-down`, `.hidden-md-up`, `.hidden-md-down`, `.hidden-lg-up` e `.hidden-lg-down`.
- Os utilitários de impressão não iniciam mais com `.hidden-` ou `.visible-`, mas com `.d-print-`.
  - Classes antigas: `.visible-print-block`, `.visible-print-inline`, `.visible-print-inline-block` e `.hidden-print`;
  - Novas classes: `.d-print-block`, `.d-print-inline`, `.d-print-inline-block` e `.d-print-none`.

Invés de usar classes explícitas `.visible-*`, faça um elemento visível, simplesmente, não escondendo-o naquele tamanho de tela. Você pode combinar uma classe `.d-*-none` com outra `.d-*-block` para mostrar um elemento só em um intervalo de tamanhos de telas (ex: `.d-none.d-md-block.d-xl-none` mostra o elemento só em dispositivos médios e grandes).

Com as mudanças nos breakpoints do grid (na v4), você vai ter que usar um breakpoint maior que o usual, se quiser atingir o mesmo resultado. As novas classes utilitárias não tentam acomodar menos casos comuns, onde a visibilidade de um elemento não pode ser expressa como um simples intervalo de tamnhos de viewports. Nesses casos, você precisará de CSS personalizado, invés.