

© JMA 2015. All rights reserved

RWD – Responsive Web Design

DISEÑO WEB ADAPTATIVO

© JMA 2015. All rights reserved

Diseño Adaptativo

- Es un enfoque de diseño destinado a la elaboración de sitios/aplicaciones para proporcionar un entorno óptimo de:
 - Lectura Fácil
 - Navegación correcta con un número mínimo de cambio de tamaño
 - Planificaciones y desplazamientos
- Con la irrupción multitud de nuevos dispositivos y el que el acceso a internet se realiza ya mayoritariamente desde dispositivos diferentes a los tradicionales ordenadores ha obligado a seguir dicho enfoque en las aplicaciones WEB.
- Contempla la definición de múltiples elementos antes de la realización de la programación real.
 - Elementos de página en las unidades de medidas correctas
 - Imágenes flexibles
 - Utilización de CSS dependiendo de la aplicación

© JMA 2015. All rights reserved

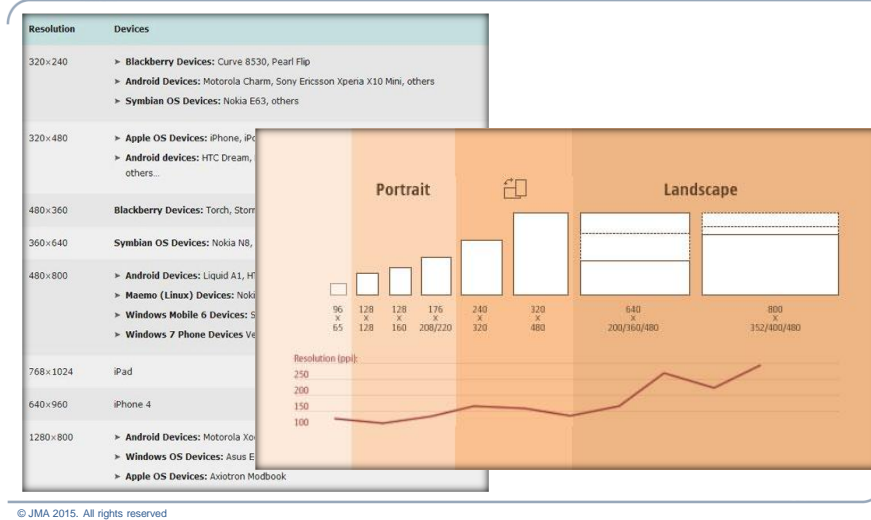
Resolución

- Los dispositivos móviles tienen una característica distintiva, y es su resolución de pantalla.
- Es necesario conocer cuales son las resoluciones más comunes en este tipo de dispositivos móviles, de los gadgets más utilizados, etc.
- Las resoluciones van cambiando de forma muy rápida y en dispositivos nuevos



© JMA 2015. All rights reserved

Resolución



© JMA 2015. All rights reserved

Resolución

- También deberemos tener en cuenta la resoluciones de otros dispositivos como:
 - Tablets
 - TV SmartTV
 - Pizarras electrónicas, etc



Pizarras 10.1" y 11.6" (2560x1440, 1920x1080, 1366x768), 17" (1920x1080)

PC 12" (1280x800), 14" (1920x1080, 1366x768), 15.6" (1920x1080)

Family hub 23" (1920x1080), 27" (2560x1440)

© JMA 2015. All rights reserved

Orientación de Página

- La orientación del papel es la forma en la que una página rectangular está orientada y es visualizada.
- Los dos tipos más comunes son:
 - Landscape (Horizontal)
 - Portrait (Vertical)



© JMA 2015. All rights reserved

Recomendaciones de Diseño

1. Utilizar porcentajes y “ems” como unidad de medida en lugar de utilizar los valores determinados como definición de pixel.

Las ems son unidades relativas,
así que más exactamente 1 em equivale
al cien por cien del tamaño inicial de
fuente.

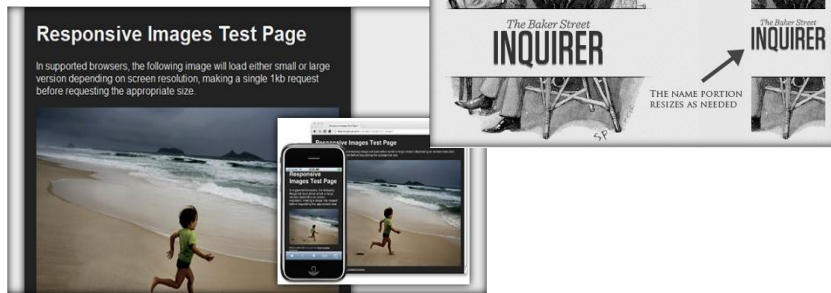
El tamaño de la fuente de los elementos de texto debe adaptarse a la pantalla que se utilice.

© JMA 2015. All rights reserved

Recomendaciones de Diseño

2. Determinar el tamaño y definición de las imágenes a utilizar

- Recortar, Ajustar, etc



© JMA 2015. All rights reserved

Recomendaciones de Diseño

3. El contenido y funcionalidad BÁSICA debe de ser accesible por todos los navegadores



© JMA 2015. All rights reserved

Recomendaciones de Diseño

4. Definición correcta de contenidos en función del dispositivo



© JMA 2015. All rights reserved

Recomendaciones de Diseño

5. Navegación responsiva

- Tradicionalmente, la navegación de una página web se muestra en la parte superior de la página, normalmente como una lista horizontal de menús con submenús que se abren al pasar el ratón. Este tipo de navegación es claramente inadecuado para los dispositivos móviles: no hay suficiente espacio y no hay ratón.
- Hay varios enfoques para abordar la navegación en los dispositivos móviles. Todos ellos ahorran espacio y no requieren un ratón. La atención del usuario suele dirigirse a la navegación con una animación. Los enfoques más populares para la navegación responsiva incluyen:
 - El icono “hamburger menu” (tres líneas horizontales paralelas): este elemento se toca para activar el menú.
 - Navegación “Off canvas”: el menú se desliza desde el borde de la pantalla y empuja el contenido actual de la pantalla hacia un lado.

© JMA 2015. All rights reserved

Ventajas

- Soporte de dispositivos móviles.
- Con una sola versión en HTML y CSS se cubren todas las resoluciones de pantalla.
- Mejora la experiencia de usuario.
- Se reducen los costos de creación y mantenimiento cuando el diseño de las pantallas es similar entre dispositivos de distintos tamaños.
- Evita tener que desarrollar aplicaciones específicas para cada sistema operativo móvil.
- Facilita la referenciación y posicionamiento en buscadores, versión única contenido/página.

© JMA 2015. All rights reserved

Mobile First



Responsive Web Design

Mobile First Web Design



© JMA 2015. All rights reserved

Mobile First

- En 2015, Google anunció que por primera vez había recibido más búsquedas realizadas desde dispositivos móviles que de escritorio.
- Mobile First es un concepto de diseño web optimizado para móviles, donde inicialmente se crea una página web para tablets y smartphones, para adaptarla luego a las necesidades de un navegador de escritorio:
 - Se concentra en lo esencial
 - No invierte mayor esfuerzo en la programación de lo necesario
 - Pensado para un máximo rendimiento en todos los dispositivos
 - Su información es de rápido acceso
 - Cuenta con diseños personalizados y pensados para smartphones
 - Evita el uso de imágenes de gran tamaño y características innecesarias
 - Incluye reducciones en el código fuente
 - La web es programada directamente en HTML5, renunciando así a JavaScript

© JMA 2015. All rights reserved

Mobile friendly

- Se considera una página web como mobile friendly si tiene un diseño optimizado para móviles.
- Para comprobar si una página web es considerada como mobile friendly basta con introducir la url en la herramienta Google Webmaster Tools que hará una evaluación de los factores que se enumeran a continuación.
- Una página web es mobile friendly cuando:
 - Se puede acceder a su contenido fácilmente
 - El contenido es adaptativo y fácil de leer
 - Cuenta con tiempos cortos de carga
 - Los enlaces son visibles y se puede hacer clic en ellos
- Una página web es mobile unfriendly cuando:
 - Se utiliza un software poco común para dispositivos móviles (por ejemplo Flash)
 - El tipo de letra es pequeño e ilegible
 - Los enlaces están demasiado cerca, o se ven uno encima del otro
 - El contenido es muy grande para la pantalla

© JMA 2015. All rights reserved

viewport

- Hace referencia a la región visible del navegador, o sea, la parte de la página que está visualizándose actualmente en el navegador.
- Podemos redimensionar la ventana del navegador para reducir el tamaño del viewport y simular que se trata de una pantalla y dispositivo más pequeño.
`<meta name="viewport" content="initial-scale=1, width=device-width">`
- Los parámetros de comportamiento para el viewport son:
 - width: Indica un ancho para el viewport.
 - height: Indica un alto para el viewport.
 - initial-scale: Escala inicial con la que se visualiza la página web.
 - minimum-scale: Escala mínima a la que se puede reducir al hacer zoom.
 - maximum-scale: Escala máxima a la que se puede aumentar al hacer zoom.
 - user-scalable: Posibilidad de hacer zoom en la página web.

© JMA 2015. All rights reserved

Unidades Relativas

Unidad	Relativa a
em	Tamaño de letra del elemento padre, en el caso de propiedades tipográficas como font-size, y tamaño de la fuente del propio elemento en el caso de otras propiedades, como width.
ex	Altura x de la fuente del elemento.
ch	La medida de avance (ancho) del glifo "0" de la letra del elemento.
rem	Tamaño de la letra del elemento raíz.
lh	Altura de la línea del elemento.
vw	1% del ancho de la ventana gráfica.
vh	1% de la altura de la ventana gráfica.
vmin	1% de la dimensión más pequeña de la ventana gráfica.
vmax	1% de la dimensión más grande de la ventana gráfica.

© JMA 2015. All rights reserved

Consultas de medios

- Las consultas de medios en CSS3 extiende la idea de CSS2: en lugar de buscar un tipo de dispositivo, se mira la capacidad del dispositivo.
- Las consultas de medios se pueden utilizar para comprobar muchas cosas, tales como:
 - anchura y la altura de la ventana gráfica
 - anchura y la altura del dispositivo
 - orientación (es la tableta / teléfono en modo horizontal o vertical)
 - resolución

```
@media not|only mediatype and (expressions) {
    CSS-Code;
}
```

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">
```
- Las consultas de medios son una técnica popular para la entrega de una hoja de estilo a medida para tabletas, iPhone y Androids.


```
@media all and (min-width: 480px) and (max-width: 767px) {
    body {
        background-color: lightgreen;
    }
}
```

© JMA 2015. All rights reserved

Consultas de medios

- Breakpoints más comunes


```
@media all and (min-width:1200px){ ... }
@media all and (min-width: 960px) and (max-width: 1199px) { ... }
@media all and (min-width: 768px) and (max-width: 959px) { ... }
@media all and (min-width: 480px) and (max-width: 767px){ ... }
@media all and (max-width: 479px) { ... }
```
- La etiqueta <picture> brinda más flexibilidad para especificar recursos de imagen, se pueden diseñar varias imágenes para adecuarlas al tamaño de visualización. Contiene dos etiquetas: una o más etiquetas <source> y una etiqueta como una opción alternativa si ninguna de las etiquetas de origen coincide.


```
<picture>
  <source media="(min-width:650px)" srcset="img_media.jpg">
  <source media="(min-width:465px)" srcset="img_pequeña.jpg">
  
</picture>
```

© JMA 2015. All rights reserved

INTRODUCCIÓN

© JMA 2015. All rights reserved

Características

- Twitter Bootstrap es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web adaptativo (diseño).
 - Bootstrap se puede descargar y usar de forma totalmente gratuita.
 - Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales que son opcionales.
 - Bootstrap es el framework libre de cliente más rápido y fácil para el desarrollo web adaptativo apto para dispositivos móviles
 - Permite la creación de sitios web que se ajustan a sí mismos automáticamente para que visualicen correctamente en todos los dispositivos, desde pequeños teléfonos a grandes televisores.
 - Muchos de los componentes requieren el uso de JavaScript para funcionar.
-

© JMA 2015. All rights reserved

Desventajas

- ¿Es difícil aprender Bootstrap 5?
Jugaremos al memorice... solo es cosa de aprender sus clases y ya, aunque hay [extensiones](#) que asisten en la escritura (IntelliSense). A medida que realizas proyectos verás lo fácil de su implementación.
- ¿Con Bootstrap todos los sitios web son iguales?
Si utilizamos todos sus clases sin modificar nada, se verán todos nuestros proyectos parecidos, pero se pueden personalizar los estilos.
- ¿Se pueden realizar diseños muy complejos?
Con Bootstrap 5 vainilla se puede complicar la cosa si necesitamos un diseño altamente complejo, pero permite agregar tus propios estilos en CSS.
- ¿Mi CSS serán 60.000 líneas de código?
Si no optimizas Bootstrap se generará un archivo enorme, pero actualmente existen herramientas como [PurgeCSS](#) que nos ayudan a eliminar las clases CSS que no utilizamos, siendo el CSS resultante una fracción del original.

© JMA 2015. All rights reserved

Añadir Bootstrap a una página

- Bootstrap proporciona:
 - CSS y JS compilados (bootstrap.*), y minimizados (bootstrap.min.*).
 - Los mapas de origen (bootstrap.*.map) están disponibles para su uso con las herramientas de depuración de los navegadores.
 - Los archivos bundle JS (bootstrap.bundle.js y bootstrap.bundle.min.js) incluyen la dependencia a [Popper](#) para los tooltips (información emergente) y popovers (ventanas emergentes).
 - El código fuente de Bootstrap con las fuentes en Sass, JavaScript y documentación.
- Se pueden utilizar dos estrategias diferentes para añadirlo a una página:
 - Se pueden descargar los ficheros locales con uno de los administradores de paquetes (npm install Bootstrap) o desde <http://getbootstrap.com/>

```
<link rel="stylesheet" href="css/bootstrap.min.css">
```
 - Las CDN (Content Delivery Network) permiten compartir contenidos comunes entre diferentes sitios y evitar descargas al aprovechar la cache de los navegadores:

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
```

© JMA 2015. All rights reserved

Añadir Bootstrap a una página

- Muchos de los componentes requieren el uso de JavaScript para funcionar. Bootstrap tiene como dependencia a [Popper](#).
- Se pueden añadir por separado:


```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.2/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"></script>
```
- Está disponible un bundle con los dos paquetes:


```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
```
- Para habilitarlos, se recomienda colocarlos justo antes de la etiqueta de cierre `</body>`, por delante de los script que utilizan sus componentes.
- Los complementos se pueden incluir individualmente (usando archivos `js/dist/*.js` individuales de Bootstrap), o todos a la vez usando `bootstrap.js` o `bootstrap.min.js`. Los empaquetadores (Webpack, Rollup...) pueden usar los archivos `/js/dist/*.js` que están preparados para UMD. La versión de Bootstrap creada como ESM (`bootstrap.esm.js` y `bootstrap.esm.min.js`) permite usar Bootstrap como un módulo en el navegador, si este admite módulos.
- Casi todos los complementos de Bootstrap se pueden habilitar y configurar solo a través de HTML con atributos de datos (`data-`).

© JMA 2015. All rights reserved

Contenidos descargados

Versión compilada

```
bootstrap/
├── css/
│   ├── bootstrap-grid.css
│   ├── bootstrap-grid.css.map
│   ├── bootstrap-grid.min.css
│   ├── bootstrap-grid.min.css.map
│   ├── bootstrap-grid.rtl.css
│   ├── bootstrap-grid.rtl.css.map
│   ├── bootstrap-grid.rtl.min.css
│   ├── bootstrap-grid.rtl.min.css.map
│   ├── bootstrap-reboot.css
│   ├── bootstrap-reboot.css.map
│   ├── bootstrap-reboot.min.css
│   ├── bootstrap-reboot.min.css.map
│   ├── bootstrap-reboot.rtl.css
│   ├── bootstrap-reboot.rtl.css.map
│   ├── bootstrap-reboot.rtl.min.css
│   ├── bootstrap-reboot.rtl.min.css.map
│   ├── bootstrap-utilities.css
│   ├── bootstrap-utilities.css.map
│   ├── bootstrap-utilities.min.css
│   ├── bootstrap-utilities.min.css.map
│   ├── bootstrap-utilities.rtl.css
│   ├── bootstrap-utilities.rtl.css.map
│   ├── bootstrap-utilities.rtl.min.css
│   ├── bootstrap-utilities.rtl.min.css.map
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   └── bootstrap.rtl.css
```

```
├── bootstrap.rtl.css.map
├── bootstrap.rtl.min.css
├── bootstrap.rtl.min.css.map
├── js/
│   ├── bootstrap.bundle.js
│   ├── bootstrap.bundle.js.map
│   ├── bootstrap.bundle.min.js
│   ├── bootstrap.bundle.min.js.map
│   ├── bootstrap.esm.js
│   ├── bootstrap.esm.js.map
│   ├── bootstrap.esm.min.js
│   ├── bootstrap.esm.min.js.map
│   ├── bootstrap.js
│   ├── bootstrap.js.map
│   ├── bootstrap.min.js
│   └── bootstrap.min.js.map
```

Versión original

```
bootstrap/
├── dist/
│   ├── css/
│   ├── js/
│   ├── site/
│   └── content/
│       ├── docs/
│       │   └── 5.1/
│       └── examples/
```

© JMA 2015. All rights reserved

Personalización

- Bootstrap no solo permite sino que también alienta su personalización por parte los autores para adecuarlo a sus necesidades, suministrando el código fuente Sass que genera el CSS y el JavaScript de los componentes.
- Hay varias formas de personalizar Bootstrap:
 - Crear tu propia hoja de estilos con Sass que importe Bootstrap para aprovechar variables, mapas, mixins y funciones para que puedas modificarla y ampliarla.
 - Personalizar Bootstrap con variables Sass integradas (scss/_variables.scss) para alternar fácilmente las preferencias globales de CSS, para controlar el estilo y el comportamiento, y recompilar.
 - Utilizar las propiedades personalizadas de CSS (variables) de Bootstrap para un diseño y desarrollo rápidos, que no requieren compilación y sobre escriben los valores predeterminados.

© JMA 2015. All rights reserved

Accesibilidad

- La accesibilidad general de cualquier proyecto creado con Bootstrap depende en gran medida del marcado del autor, el estilo adicional y las secuencias de comandos que haya incluido. Sin embargo, siempre que se hayan implementado correctamente, debería ser perfectamente posible crear sitios web y aplicaciones con Bootstrap que cumplan con las WCAG 2.1 (A/AA/AAA), Sección 508, y estándares de requisitos de accesibilidad similares.
- Los componentes interactivos de Bootstrap, como los cuadros de diálogo modales, los menús desplegables y los tooltips personalizados, están diseñados para funcionar con usuarios táctiles, de mouse y de teclado. A través del uso de funciones y atributos WAI-ARIA, estos componentes también deben ser comprensibles y operables utilizando tecnologías de asistencia (como lectores de pantalla). Debido a que los componentes de Bootstrap están diseñados a propósito para ser bastante genéricos, es posible que los autores deban incluir más funciones y atributos de ARIA, así como el comportamiento de JavaScript, para transmitir con mayor precisión la naturaleza precisa y funcionalidad de su componente. Esto generalmente se indica en la documentación.
- Algunas combinaciones de colores que actualmente conforman la paleta predeterminada de Bootstrap pueden dar lugar a un contraste de color insuficiente (por debajo de la relación de contraste de color de texto recomendada WCAG 2.1 de 4.5:1 y la relación de contraste de color sin texto WCAG 2.1 de 3:1), especialmente cuando se utiliza contra un fondo claro. Se alienta a los autores a probar sus usos específicos del color y, cuando sea necesario, modificar/ampliar manualmente estos colores predeterminados para garantizar proporciones de contraste de color adecuadas.
- Bootstrap incluye soporte para la función de medios prefers-reduced-motion. En los navegadores/entornos que permiten al usuario especificar su preferencia por el movimiento reducido, la mayoría de los efectos de transición CSS en Bootstrap se desactivarán y las animaciones significativas se ralentizarán.

© JMA 2015. All rights reserved

Plantilla

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Plantilla</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="/css/bootstrap.min.css">
</head>
<body>
  <div class="container">
  :
  </div>
  <script src="/js/bootstrap.bundle.min.js"></script>
  <script >...</script>
</body>
</html>
```

© JMA 2015. All rights reserved

Reiniciar

- Para mejorar la representación entre navegadores, Bootstrap usa Reboot para corregir las inconsistencias entre los navegadores y dispositivos.
- Reboot se basa en Normalize, proporcionando muchos elementos HTML comunes con estilos un tanto pertinaz utilizando solo selectores de elementos. El estilo adicional se realiza solo con clases. Por ejemplo, reinicia algunos estilos `<table>` para una línea base más simple y luego proporciona `.table`, `.table-bordered` y más.
- Estas son las pautas y razones para elegir qué sobrescribir en Reboot:
 - Actualizar algunos valores predeterminados del navegador para usar rems en lugar de ems para el espaciado de componentes escalables.
 - Evitar `margin-top`. Los márgenes verticales pueden contraerse y producir resultados inesperados. Sin embargo, lo que es más importante, una sola dirección de `margin` es un modelo mental más simple.
 - Para escalar más fácilmente entre tamaños de dispositivos, los elementos de bloque deben usar rems para márgenes.
 - Mantener al mínimo las declaraciones de propiedades relacionadas con `font`, usando `inherit` siempre que sea posible.

© JMA 2015. All rights reserved

Elementos del Bootstrap

- **Layout:** sistema de cuadrilla y diseño adaptativo que contienen, rellenan y alinean el contenido dentro de un dispositivo o viewport.
- **Ajustes globales:** definiciones básicas de estilo para todos los elementos de HTML que otorga una uniformidad a los navegadores y al sistema de anchura.
- **Componentes reutilizables:** combinaciones de estilos y código JavaScript para crear interfaces de elementos comúnmente usados.
- **Formularios:** pautas de uso para los estilos de controles de formulario, opciones de diseño y componentes personalizados para crear una amplia variedad de formularios.
- **Helpers y Utilidades:** clases de estilo predefinidas que permiten definir el estilo sin establecer directamente propiedades CSS.
- **API programática JavaScript:** provee elementos adicionales al interfaz de usuario para programar transiciones, cuadros modales, avisos, tooltips, carruseles y otros aspectos.

© JMA 2015. All rights reserved

ESTRUCTURA

© JMA 2015. All rights reserved

Sintaxis

- Bootstrap define muchas clases CSS para personalizar los diferentes elementos.
- Utiliza una sintaxis declarativa:
 - Atributo CLASS
 - Atributos personalizados
 - Estilos por defecto
- El sistema de cuadrícula (grid) de Bootstrap utiliza una serie de contenedores, filas y columnas para diseñar y alinear el contenido. Está construido con flexbox y es completamente responsive.
- El elemento base es la rejilla o cuadrícula. Bootstrap incluye una rejilla o retícula fluida, pensada para móviles, que administra el espacio disponible (posicionamiento y tamaños) para cumplir con el diseño web adaptativo.
- Esta retícula crece hasta 12 columnas a medida que crece el tamaño de la pantalla del dispositivo.
- Bootstrap incluye clases CSS para utilizar la cuadrícula directamente en los diseños.

© JMA 2015. All rights reserved

Cuadrícula

```
<div class="container">
  <!-- Stack the columns on mobile by making one full-width and the other half-width -->
  <div class="row">
    <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
    <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  </div>
  <!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
  <div class="row">
    <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
    <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
    <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  </div>
  <!-- Columns are always 50% wide, on mobile and desktop -->
  <div class="row">
    <div class="col-xs-6">.col-xs-6</div>
    <div class="col-xs-6">.col-xs-6</div>
  </div>
</div>
```

© JMA 2015. All rights reserved

Rejilla

- Las filas siempre se definen dentro de un contenedor de tipo `.container` (anchura de columna fija) o de tipo `.container-fluid` (anchura de columna variable). De esta forma las filas se alinean bien y muestran el padding correcto.
- Las filas se utilizan para agrupar horizontalmente a varias columnas.
- El contenido siempre se coloca dentro de las columnas, ya que las filas sólo deberían contener como hijos elementos de tipo columna.
- La separación entre columnas se realiza aplicando padding (llamado gutter). Para contrarrestar sus efectos en la primera y última columnas, las filas (elementos `.row`) aplican márgenes negativos.
- Las columnas de la rejilla definen su anchura especificando cuántas de las 12 columnas de la fila ocupan.
- Una columna puede tener varias definiciones, separadas por espacios, para los diferentes tamaños de dispositivos.

© JMA 2015. All rights reserved

Sistema de cuadrícula

- **Breakpoints:** se basan en media queries min-width, lo que significa que afectan ese breakpoint y todos los anteriores. Esto significa que puedes controlar el tamaño y el comportamiento del contenedor y la columna en cada breakpoint.
- Los **contenedores** centran y rellenan horizontalmente su contenido, las filas.
- Las **filas** son envoltorios para las columnas: Hay 12 columnas de plantilla disponibles por fila, lo que te permite crear diferentes combinaciones de elementos que abarcan cualquier número de columnas.
- Las **columnas** son increíblemente flexibles: Las columnas envuelven al contenido a presentar y pueden indicar el número de columnas de plantilla que abarcan.
- Los **gutters** también son responsive y personalizables. Cada columna tiene un padding horizontal (llamado gutter) para controlar el espacio entre ellas, que se contrarresta en las filas con márgenes negativos para garantizar que el contenido de tus columnas esté alineado visualmente en el lado izquierdo. Las filas también admiten clases modificadoras para aplicar uniformemente el tamaño de columna y clases de gutter para cambiar el espaciado de tu contenido. Las clases Gutter están disponibles en todos los breakpoints, con los mismos tamaños que nuestro espaciado de margen y padding. Cambia los gutters horizontales con las clases `.gx-*`, los gutters verticales con `.gy-*` o todos los gutters con las clases `.g-*`. `.g-0` también está disponible para eliminar gutters.

© JMA 2015. All rights reserved

Breakpoints

- Los breakpoints son anchos personalizables que determinan cómo se comporta el diseño responsive en los tamaños de dispositivo o viewport en Bootstrap.
- Los breakpoints son los componentes básicos del diseño responsive. Permiten controlar cuándo se puede adaptar el diseño a un viewport particular o tamaño de dispositivo.
- Utilizan media queries para diseñar el CSS por breakpoint. Las media queries son una función de CSS que permite aplicar estilos de forma condicional en función de un conjunto de parámetros del navegador y del sistema operativo. Comúnmente usan min-width en las media queries.
- Mobile first como objetivo, aplicar el mínimo de estilos para hacer que un diseño funcione en el breakpoint más pequeño, y luego capas de estilos para ajustar ese diseño para dispositivos más grandes. Esto optimiza el CSS, mejora el tiempo de renderizado y proporciona una gran experiencia para los visitantes.
- Bootstrap incluye seis breakpoints predeterminados, denominados a veces niveles de cuadrícula.

© JMA 2015. All rights reserved

Breakpoints

Breakpoint	Infijo de clase	Dimensiones	Dispositivos
X-Small	None	<576px	Dispositivos muy pequeños: Smartwatch
Small	sm	≥576px	Dispositivos pequeños: Teléfonos
Medium	md	≥768px	Dispositivos medianos: Teléfonos, Tablets
Large	lg	≥992px	Dispositivos grandes: Tablets, Laptop
Extra large	xl	≥1200px	Dispositivos muy grandes: Ordenadores
Extra extra large	xxl	≥1400px	Dispositivos muy muy grandes: Ordenadores, TV, Pizarras

© JMA 2015. All rights reserved

Contenedores

- Los contenedores son el elemento de diseño más básico en Bootstrap y son necesarios cuando se usa el sistema de cuadrícula predeterminado (flexbox). Los contenedores se utilizan para contener, rellenar y (a veces) centrar el contenido dentro de ellos. Los contenedores pueden anidarse aunque la mayoría de los diseños lo no requieren.
- Bootstrap viene con tres tipos de contenedores diferentes:
 - .container: contenedor adaptativo de ancho fijo, lo que significa que su max-width cambia en cada breakpoint.
 - .container-fluid: contenedor de ancho completo (width: 100%) que abarque todo el ancho del viewport.
 - .container-{breakpoint}: contenedor adaptativo que tiene un 100% de ancho hasta que se alcanza el breakpoint especificado, después usa el max-width de cada breakpoint superior.

© JMA 2015. All rights reserved

Filas

- La filas distribuyen las columnas que comienzan apilando (verticalmente) y se vuelve horizontal en los breakpoint superiores cuando el ancho de las columnas lo permiten. Con .row se establecen las filas.
- Usa las clases responsive .row-cols-* para establecer rápidamente la cantidad de columnas que mejor representen tu contenido y diseño. Con .row-cols-{breakpoint}-* se establece la cantidad de columnas para un determinado breakpoint. Con .row-cols-auto puedes dar a las columnas su ancho natural.


```
<div class="container">
  <div class="row row-cols-2">
    <div class="col">Columna</div>
    <div class="col">Columna</div>
    <div class="col">Columna</div>
    <div class="col">Columna</div>
  </div>
</div>
```

© JMA 2015. All rights reserved

Columnas

- El auto-layout para las columnas de cuadrícula de flexbox significa que puedes establecer el ancho de una columna y hacer que las columnas hermanas cambien de tamaño automáticamente a su alrededor, se mostraran horizontalmente o apiladas.
- Las clases para definir las columnas:
 - .col: divide el ancho según el número de columnas definidas y cada columna tendrá el mismo ancho.
 - .col-{cols}: establece el ancho de una columna y hace que las columnas hermanas cambien de tamaño automáticamente a su alrededor.
 - col-{breakpoint}-auto: para dimensionar las columnas en función del ancho natural de su contenido.
 - .col-{breakpoint}-{cols}: establece diferentes anchos de columna para cada breakpoint y superiores, permitiendo diferentes diseños en función del breakpoint.
- Se puede usar una combinación de diferentes clases para cada nivel según sea necesario.

© JMA 2015. All rights reserved

Alineamiento

- Las columnas se basan en la arquitectura flexbox de la cuadrícula.
- Las utilidades de alineación de flexbox para alinear son:
 - .align-items-*: establece la alineación vertical de toda la fila.
 - .align-self-*: establece la alineación vertical de una columna dentro de la fila.
 - .justify-content-*: establece la alineación horizontal de las columnas en la fila.

```

<div class="container">
  <div class="row align-items-center">
    <div class="col">Columna</div>
    <div class="col">Columna</div>
  </div>
  <div class="row justify-content-around">
    <div class="col align-self-start">Columna</div>
    <div class="col align-self-center">Columna</div>
    <div class="col align-self-end">Columna</div>
  </div>
</div>

```

© JMA 2015. All rights reserved

Reordenando

- Usa las clases `.order-*` (del 0 al 5) para controlar el orden visual de tu contenido.
- Estas clases son responsive, por lo que puedes establecer y alterar el orden por breakpoint con `.order-{breakpoint}-*`.
- El valor del orden, del 0 al 5, establece el peso en la ordenación y no la posición, los valores inferiores aparecen antes que los superiores. Ha igualdad de peso se mantiene el orden de aparición. Por defecto todas las columnas tienen `.order-0` por lo que aparecen antes que las que tienen orden especificado.
- También están las clases responsive `.order-first` y `.order-{breakpoint}-first` para ordenar al principio, `.order-last` y `.order-{breakpoint}-last` para ordenar al final.

```
<div class="container">
  <div class="row align-items-center">
    <div class="col order-lg-last">Menu</div>
    <div class="col">Cuerpo</div>
  </div>
```

© JMA 2015. All rights reserved

Offset y Padding

- Las clases de compensación de columnas mueven las columnas a la derecha usando `.offset-{breakpoint}-*`. Estas clases aumentan el margen izquierdo de la columna en n columnas.
- Los gutters son el padding entre las columnas, que se utilizan para espaciar y alinear de manera responsive el contenido en el sistema de cuadrícula de Bootstrap. Las clases `.g-*`, `.gx-*` y `.gy-*` se pueden usar para controlar los anchos de los gutters horizontales y verticales. Los gutters se pueden ajustar de forma responsive con `.g-{breakpoint}-*`, `.gx-{breakpoint}-*` y `.gy-{breakpoint}-*`. Los gutters se pueden establecer para filas y columnas.
- Se pueden utilizar las utilidades gap en el contenedor de cuadrícula principal y evitar tener que agregar utilidades de margen a elementos de cuadrícula individuales.


```
.gap-{0 .. 5 | auto} .gap-{breakpoint}-{0 .. 5 | auto}
```
- Para el espacio vertical entre los elementos del contenedor:


```
.row-gap-{0 .. 5 | auto} .row-gap-{breakpoint}-{0 .. 5 | auto}
```

© JMA 2015. All rights reserved

Espaciado

- Las clases se nombran usando el formato {property}{sides}-{size} y {property}{sides}-{breakpoint}-{size}.
- Donde property es m (para margin) o p (para padding)
- Donde sides es uno de:
 - t: *-top
 - b: *-bottom
 - s - (start) *-left en LTR, *-right en RTL
 - e - (end) *-right en LTR, *-left en RTL
 - x: tanto *-left como *-right
 - y: tanto *-top como *-bottom
 - en blanco: los 4 lados del elemento
- Donde size es uno de:
 - 0: se eliminan el margin o padding al establecerlo en 0
 - 1: se calcula \$spacer * 0.25
 - 2: se calcula \$spacer * 0.5
 - 3: \$spacer
 - 4: se calcula \$spacer * 1.5
 - 5: se calcula \$spacer * 3
 - auto: en automático

© JMA 2015. All rights reserved

Flexbox

- Las clases de utilidad permiten administrar diseños flexbox:
 - .d-flex, .d-{breakpoint}-flex, .d-inline-flex, .d-{breakpoint}-inline-flex
- Direcciones:
 - .flex-{breakpoint}-{row | column | row-reverse | column-reverse}
- Justificar el contenido:
 - .justify-content-{start | end | center | between | around | evenly}
- Alinear elementos:
 - .align-items-{start | end | center | baseline | stretch}
- Alinearse uno mismo:
 - .align-self-{auto | start | end | center | baseline | stretch}
- Alinear contenido:
 - .align-content-{start | end | center | between | around | stretch}

© JMA 2015. All rights reserved

Flexbox

- Llenar:
.flex-fill
- Crecer y encogerse:
.flex-{grow|shrink}-{0|1}
- Envolturas:
.flex-nowrap, .flex-wrap, .flex-wrap-reverse
- Pilas: Las pilas ofrecen un atajo para aplicar una serie de propiedades de flexbox para crear diseños rápida y fácilmente en Bootstrap.
 - .vstack: para crear diseños verticales. Los elementos apilados ocupan el ancho completo de forma predeterminada.
 - .hstack: para diseños horizontales. Los elementos apilados están centrados verticalmente de forma predeterminada y solo ocupan el ancho necesario.

```
<div class="hstack gap-3">
  <div class="bg-light border">Primer elemento</div>
  <div class="bg-light border">Segundo elemento</div>
  <div class="bg-light border">Tercer elemento</div>
</div>
```

© JMA 2015. All rights reserved

CSS Grid

- El sistema de cuadrícula predeterminado de Bootstrap representa la culminación de más de una década de técnicas de diseño de CSS, probadas por millones de personas. Pero también se creó sin muchas de las funciones y técnicas modernas de CSS 3 disponibles en navegadores como el nuevo CSS Grid.
- Con Bootstrap 5, se agrego la opción **experimental** para habilitar un sistema de cuadrícula separado que se basa en CSS Grid, pero con un toque de Bootstrap. Todavía recibe clases que puedes aplicar por capricho para crear diseños responsive, pero con un enfoque diferente bajo el capó.
- Diferencias clave en comparación con el sistema de cuadrícula predeterminado:
 - Las utilidades de Flex no afectan las columnas de CSS Grid de la misma manera.
 - Los huecos (gaps) reemplazan a los gutters. La propiedad gap reemplaza el padding horizontal del sistema de cuadrícula predeterminado y funciona más como margin.
 - Como tal, a diferencia de .rows, .grid no tienen márgenes negativos y las utilidades de margen no se pueden usar para cambiar los gutters de la cuadrícula. Los espacios de cuadrícula (gaps) se aplican horizontal y verticalmente de forma predeterminada.
 - Los estilos en línea y personalizados deben verse como reemplazos de las clases de modificadores (por ejemplo, style="--bs-columns: 3;" vs class="row-cols-3").
 - El anidamiento funciona de manera similar, pero puede requerir que se restablezcan los recuentos de columnas en cada instancia de un .grid anidado.

© JMA 2015. All rights reserved

ESTILOS CSS

© JMA 2015. All rights reserved

Colores

- Bootstrap suministra variantes de color para agregar significado al proporcionar una indicación visual (influye en la accesibilidad).Variantes:
 - primary: Primario, utilizado para hipervínculos, estilos de enfoque y estados activos de componentes y formularios.
 - secondary: Secundario, texto más claro, para divisores y para indicar estados de componentes deshabilitados.
 - tertiary: Terciario, texto aún más claro, para diseñar fondos para estados de desplazamiento, acentos y fuentes.
 - success: Éxito, para acciones e información positivas o exitosas.
 - danger: Peligro, para errores y acciones peligrosas.
 - warning: Advertencia, para mensajes de advertencia no destructivos.
 - info: Información, para contenido neutral e informativo.
 - light: Claro, para colores con menos contrastantes.
 - dark: Oscuro, para colores con mayor contraste.
 - black, white, body: para referenciar colores.
 - Atributos -subtle (sutil, mas clarito) y -emphasis (énfasis, mayor contraste)
- Bootstrap admite modos de color: `data-bs-theme="light | dark"`

© JMA 2015. All rights reserved

Color de fondo

- De forma similar a las clases de color de texto contextual, establece el fondo de un elemento en cualquier clase contextual. Las utilidades de fondo no configuran color, por lo que en algunos casos tendrás que usar `.text-*`.
`.bg-{color}`
- Gradiente de fondo: Al agregar una clase `.bg-gradient`, se agrega un degradado lineal como imagen de fondo. Este degradado comienza con un blanco semitransparente que se desvanece hasta el fondo.
`.bg-{color}.bg-gradient`
- Opacidad: para la transparencia alfa (con un valor predeterminado 1).
`.bg-opacity-{10 | 25 | 50 | 75}`

© JMA 2015. All rights reserved

Tipografía

- Redefine los estilos de:
`h1 ... h6`, `p`, `.mark`, `ins`, `del`, `s`, `u`, `strong`, `.small`, `em`, `abbr`, `address`, `blockquote`, `cite`
- Crea estilos específicos para las etiquetas de código:
`code`, `kbd`, `var`, `pre`
- Clases para listas
`ul`, `ol`: `.list-unstyled`, `.list-inline`, `.list-inline-item`
`dl`: `.dl-horizontal`
- Clases para enlaces:
`.link-{color}`, `.link-opacity-{10|25|50|75|100}`, `link-opacity-*-hover`, `.link-underline-{color}`, `.link-underline-opacity-*`, `.link-offset-{1|2|3}`, `.link-offset-*-hover`, `.stretched-link` ("estira" el enlace para que se pueda pulsar en cualquier punto de su bloque contenedor)
- Clases de propias:
`.display-1display-6`: Títulos (los encabezados tradicionales están diseñados para funcionar mejor en el cuerpo)
`.lead`: Párrafo destacado
`.vr`: Reglas verticales están inspiradas en el elemento `<hr>`

© JMA 2015. All rights reserved

Tipografía

- Alineación, ajuste y desbordamiento del texto
.text-start, .text-end, .text-center, .text-wrap, .text-nowrap, .text-break, .text-truncate (...)
- Tamaño (font-size), peso (font-weight) y estilo (font-style)
.fs-{1 .. 6}, .fw-bold, .fw-bolder, .fw-normal, .fw-light, .fw-lighter, .fst-italic, .fst-normal
- Transformación de texto
.text-lowercase, .text-uppercase, .text-capitalize
- Decoración de texto
.text-decoration-none, .text-decoration-underline, .text-decoration-line-through
- Altura de la línea y fuente de espaciado de caracteres
.lh-1, .lh-sm, .lh-base, .lh-lg
.font-monospace
- Clases de colores y opacidad:
.text-{color}, .text-muted (.text-secondary sin opacidad), .text-reset (para que herede el color de su padre), .text-bg-{color}, .text-opacity-{10 | 25 | 50 | 75}

© JMA 2015. All rights reserved

Interacciones

- Selección de texto: Para cambiar la forma en que se selecciona el contenido cuando el usuario interactúa con él.
.user-select-all: el contenido se seleccionará por completo cuando el usuario haga clic.
.user-select-none: el contenido no se podrá seleccionar cuando el usuario haga clic en él.
.user-select-auto: el contenido tiene el comportamiento de selección predeterminado aunque este dentro de un user-select-none.
- Eventos de puntero: Para evitar o permitir interacciones (mouse, lápiz óptico, toque) con elementos.
.pe-none: deshabilita los eventos de mouse, lápiz óptico o toque.
.pe-auto: habilita los eventos de mouse, lápiz óptico o toque aunque este dentro de un pe-none.
- Anillo de enfoque: Modifica el estilo del anillo de enfoque, elemento que tiene el foco, por un box-shadow.
.focus-ring

© JMA 2015. All rights reserved

Display

- Incluye soporte para algunos de los valores display más comunes, así como algunos extras para controlar la visualización al imprimir.
`.d-{modo}` (para xs), `.d-{breakpoint}-{modo}` (para sm, md, lg, xl, y xxl)
- Donde {modo} es uno de: none, inline, inline-block, block, grid, table, table-cell, table-row, flex, inline-flex.
- En función al breakpoint habrá que mostrar y ocultar elementos por dispositivo.
`.d-block`, `.d-{breakpoint}-block`, `.d-none`, `.d-{breakpoint}-none`
- También habrá que indicar que elementos son imprimibles y como:
`.d-print-{modo}`

© JMA 2015. All rights reserved

Visibilidad y opacidad

- Como alternativa sin breakpoint, las clases `.visible` o `.invisible` controlan la visibilidad de los elementos, sin modificar su visualización y sin modificar el valor display.
- Los elementos con la clase `.invisible` se ocultarán tanto visualmente como para los lectores de pantalla/tecnología de asistencia.
- `.visually-hidden` oculta visualmente un elemento pero esta expuesto a tecnologías de asistencia.
- `.visually-hidden-focusable` oculta visualmente un elemento de forma predeterminada, pero se muestra cuando está enfocado (por ejemplo, cuando solo se usa el teclado).
- Como alternativa a la visibilidad está la opacidad (en %):
`.opacity-{0 | 25 | 50 | 75 | 100}`

© JMA 2015. All rights reserved

Alineación vertical, tamaño y desbordamiento

- Las clases de alineación vertical solo afectan a los elementos en línea (inline), bloque en línea (inline-block), tabla en línea (inline-table) y celda de tabla:
 - `.align-{baseline | top | middle | bottom | text-bottom | text-top}`
- Las utilidades de ancho y alto incluyen soporte para 25%, 50%, 75%, 100% y auto por defecto, tanto actual como máximo. También se pueden establecer al 100 % del viewport.
 - Ancho: `w-{25 | 50 | 75 | 100 | auto}`
 - Ancho máximo: `mw-{25 | 50 | 75 | 100 | auto}`
 - Ancho viewport: `min-vw-100, vw-100`
 - Alto : `h-{25 | 50 | 75 | 100 | auto}`
 - Alto máximo: `mh-{25 | 50 | 75 | 100 | auto}`
 - Alto viewport: `min-vh-100, vh-100`
- Una vez fijado el tamaño se podrá configurar cómo el contenido desborda un elemento.
 - `.overflow-{auto | hidden | visible | scroll}`

© JMA 2015. All rights reserved

Posicionamiento y flotación

- Las clases de posicionamiento rápido, aunque no son responsive, son:
 - `.position-static, .position-relative, .position-absolute, .position-fixed, .position-sticky`
- Se pueden organizar estos elementos fácilmente con el posicionamiento de bordes. El formato es `{property}-{position}`.


```
<div class="position-absolute top-50 start-50"></div>
```
- Donde property es uno de:
 - `top`: para la posición vertical top
 - `start`: para la posición horizontal left (en LTR) o right (en RTL)
 - `bottom`: para la posición vertical bottom
 - `end`: para la posición horizontal right (en LTR) o left (en RTL)
- Donde position es uno de:
 - `0`: para posición de borde 0
 - `50`: para posición de borde 50%
 - `100`: para posición de borde 100%
- Las clases que hacen flotante un elemento a la izquierda, a la derecha o deshabilitan la flotación son:
 - `.float-start, .float-end, .float-none, .float-{breakpoint}-start, .float-{breakpoint}-end, .float-{breakpoint}-none, .clearfix` (cuando un diseño utiliza muchos elementos flotantes, es común tener que limpiar un elemento para que no le afecten otros elementos flotantes)

© JMA 2015. All rights reserved

z-index

- Si bien no forman parte del sistema de cuadrícula de Bootstrap, los z-index juegan un papel importante en la forma en que los componentes se superponen e interactúan entre sí.
- Varios componentes de Bootstrap utilizan z-index, la propiedad CSS que ayuda a controlar el diseño al proporcionar un tercer eje para organizar el contenido. Se utiliza una escala de z-index predeterminada en Bootstrap que ha sido diseñada para capas de navegación, tooltips y popovers, modals y más.
- Se usan las utilidades z-index para apilar elementos uno encima del otro, donde n1 es el fondo y 3 el frente, solo si el valor position-* es distinto de position-static.
`.z-{n1 | 0 .. 3}`

© JMA 2015. All rights reserved

Bordes y sombras

- Bordes:
 - Añadir: `.border`, `.border-top`, `.border-end`, `.border-bottom`, `.border-start`
 - Quitar: `.border-0`, `.border-top-0`, `.border-end-0`, `.border-bottom-0`, `.border-start-0`
 - Color: `.border-{color}`
 - Ancho: `.border-{1 .. 5}`
 - Esquinas redondeadas: `.rounded`, `.rounded-top`, `.rounded-end`, `.rounded-bottom`, `.rounded-start`, `.rounded-circle`, `.rounded-pill`
 - Escala para esquinas redondeadas: `.rounded-{0 .. 3}`
- Sombras:
 - Normal: `.shadow`
 - Pequeña: `.shadow-sm`
 - Grande: `.shadow-lg`
 - Sin sombra: `.shadow-none`

© JMA 2015. All rights reserved

Imágenes, figuras y videos

- Bootstrap define varias clases CSS para que las imágenes tengan un comportamiento responsive (para que nunca se vuelvan más anchas que sus padres) y agregarles estilos livianos:
 - .img-fluid, para que la imagen escale con el ancho de su elemento padre (max-width: 100%; height: auto;).
 - .img-thumbnail, muestra la imagen con un relleno blanco y un borde fino simulando el aspecto de las fotografías de las antiguas cámaras instantáneas. Añade además una breve animación para hacer que la imagen aparezca al cargar la página.
 - .figure, .figure-img y .figure-caption proporcionan estilos para <figure> y <figcaption>. Las imágenes en las figuras no tienen un tamaño explícito, hay que agregar la clase .img-fluid a la para que sea responsive.
 - .ratio ratio-{1x1|4x3|16x9|21x9} permite administrar las proporciones de aspecto de contenido, el elemento secundario inmediato se dimensiona automáticamente.
 - .object-fit-{contain|cover|fill|scale|none} indica cómo el contenido de un elemento reemplazado debería redimensionarse para ajustarse a su contenedor.

© JMA 2015. All rights reserved

Tablas

- .table: Estilo básico con líneas de separación de filas
- .table-{color}: Estilo con variante de color
- .table-striped: Filas marcadas con franjas
- .table-striped-columns: Columnas marcadas con franjas
- .table-hover: Remarcado de la fila sobre la que está el ratón
- .table-active: Resalta una fila o celda de la tabla
- .table-bordered: Con todos los bordes en la tabla y las celdas
- .table-borderless: Para una tabla sin bordes.
- .table-group-divider: Para un borde más grueso y más oscuro entre <thead>, <tbody> y <tfoot>
- .table-sm: Tabla compacta, disminuye a la mitad el espaciado de celdas
- .table-responsive, .table-responsive-{breakpoint}: para tablas muy anchas, habilita el desplazamiento horizontal (scroll) en dispositivos pequeños hasta el breakpoint máximo.
- .caption-top: coloca el <caption> en la parte superior de la tabla

© JMA 2015. All rights reserved

GLIFOS

© JMA 2015. All rights reserved

Iconos

- Si bien Bootstrap ya no incluye un conjunto de iconos predeterminado, tiene su propia biblioteca de iconos integral llamada Bootstrap Icons.
 - Como alternativas ya probadas:
 - [Font Awesome](#)
 - [Feather](#)
 - [Octicons](#)
 - [Google Material icons](#)
 - [Ionicons](#)
 - Las bibliotecas de glifos o iconos se suelen presentar dos formatos:
 - Fuentes: Icon fonts es una técnica que representa los caracteres de texto como iconos en la pantalla.
 - SVG: Scalable Vector Graphics, es un formato de imágenes vectoriales basado en XML, con soporte para interactividad y animación.
-

© JMA 2015. All rights reserved

Iconos con Icon Fonts

- **Ventajas**
 - El color y el tamaño son más fáciles de manipular, ya que se manejan como cualquier texto a través de CSS.
 - El tamaño de archivo es menor que en SVG, y las fuentes son almacenadas en caché. Esto garantiza un mejor rendimiento.
 - Son más fáciles de integrar.
- **Desventajas**
 - Como son texto normal, el navegador les aplica antialias, generando contornos borrosos.
 - Como cada icono es un carácter, no es posible crear iconos multicolores.
 - Los navegadores presentan diferencias en la forma de mostrar las fuentes; por ejemplo, aplican más o menos grosor a cada carácter. Esto también aplica a los icon fonts y puede causar problemas para distinguir la forma de cada icono.
 - Son más difíciles de posicionar visualmente con respecto a otros elementos.
 - Suele suceder que el navegador no logre cargar el archivo de la fuente (a veces no lo hace por cuestiones de seguridad), o bien tiene problemas con el soporte de @font-face.
 - Pueden no mostrarse debido a aplicaciones para bloquear contenido que tenga instaladas el usuario.
 - Necesitas generar todos los formatos necesarios para que se muestren en todos los navegadores (aunque este problema desaparecerá pronto).
 - Son más difíciles de crear.

© JMA 2015. All rights reserved

Iconos con SVG

- **Ventajas**
 - Como son gráficos vectoriales, cambian de tamaño sin perder definición en los contornos.
 - Como están implementados en XML, podemos manipular sus partes por separado. Así, es posible crear iconos multicolores. También tenemos mayores posibilidades de animación.
 - Es más correcto en términos de semántica: mientras los icon fonts nos obligan a usar elementos vacíos o pseudoelementos de CSS, los iconos SVG se pueden representar con etiquetas como o <svg>, que fueron hechas especialmente para mostrar imágenes.
- **Desventajas**
 - Son el código fuente del gráfico que el navegador debe interpretar con el consiguiente consumo de memoria y proceso.
 - Están en formato texto cuyo tamaño ocupa mucho más que su correspondencia en binario.
 - La descarga de múltiples ficheros pequeños penaliza a los navegadores.
 - Para suplir los iconos SVG en navegadores que no soportan el formato hay que usar fallbacks (reemplazos) con varios inconvenientes.

© JMA 2015. All rights reserved

Bootstrap Icons

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

- Biblioteca de iconos de código abierto, de alta calidad y gratuita con más de 1800 iconos. Se pueden incluir como: SVG, sprite SVG o fuentes web, y usarlos con o sin Bootstrap en cualquier proyecto.
- Los iconos de Bootstrap se publican en npm, pero también se pueden descargar manualmente e individualmente o utilizar una CDN si es necesario.

```
npm i bootstrap-icons
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css">
@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css");
```
- Los iconos de Bootstrap son SVG, por lo que puede incluirlos en su HTML de varias maneras, dependiendo de cómo esté configurado el proyecto. Se recomienda usar un width: 1em y, opcionalmente, un height: 1em para cambiar el tamaño fácilmente a través de font-size.

© JMA 2015. All rights reserved

Biblioteca de iconos

- Bootstrap Icons incluye una Icon fonts (woff y woff2, unos 150kb) con todos los iconos y un CSS que importa la fuente y con clases para cada icono de la fuente. Las clases normalmente se aplican a etiquetas <i>, como abreviatura de icon en vez de italic. Para cambiar la apariencia del icono se utiliza el font-size y color.

```
<link rel="stylesheet" href="/node_modules/bootstrap-icons/font/bootstrap-icons.min.css">
<i class="bi bi-check2 text-primary fs-3"></i>
```
- Bootstrap Icons incluye un sprite SVG con symbols para todos los iconos, el fichero bootstrap-icons.svg de 1mb. Se utiliza la etiqueta <use> (SVG) con el sprite para insertar cualquier icono a través del nombre del icono como identificador del fragmento. Los sprites SVG permiten hacer referencia a un archivo externo similar a un elemento, pero con el poder decurrentColor para facilitar la creación de temas.

```
<svg class="bi" width="32" height="32" fill="currentColor">
  <use xlink:href="/bootstrap-icons/bootstrap-icons.svg#heart-fill"/>
</svg>
```

© JMA 2015. All rights reserved

Iconos individuales

- Los SVG son un origen valido para las etiquetas ``:
``
- Se puede incrustar directamente el código SVG de los íconos dentro del HTML de la página sin utilizar un archivo de imagen externo.

```

<a class="icon-link icon-link-hover" href="#">OK
  <svg xmlns="http://www.w3.org/2000/svg" width="16"
    height="16" fill="currentColor" class="bi bi-check2"
    viewBox="0 0 16 16">
    <path d="M13.854 3.646a.5.5 0 0 1 0 .708l-7 7a.5.5 0 0 1-.708
      0l-3.5-3.5a.5.5 0 1 1 .708-.708l6.5 10.293l6.646-
        6.647a.5.5 0 0 1 .708 0z" />
  </svg>
</a>

```

© JMA 2015. All rights reserved

Enlace con icono

- El componente auxiliar de enlace con icono modifica los estilos de enlace predeterminados para mejorar su apariencia y alinear rápidamente cualquier combinación de icono y texto.
- La alineación se establece a través del estilo flexbox en línea y un valor gap predeterminado.
- Estilizamos el subrayado con un desplazamiento y un color personalizados.
- Los iconos se dimensionan automáticamente para 1em que coincidan mejor con el texto asociado font-size.
- Las clases disponibles son:
 - .icon-link: en un elemento `<a>` normal, con un icono a la izquierda o a la derecha del texto de su enlace, el icono cambia de tamaño, se coloca y se colorea automáticamente.
 - .icon-link-hover: para mover el icono a la derecha al pasar el mouse.
 - .bi: en un elemento SVG le indica que es un Bootstrap Icons.

© JMA 2015. All rights reserved

CSS

- También puede usar una imagen SVG dentro de una declaración CSS (hay que asegurarse de escapar cualquier carácter, como # a %23 al especificar valores de color hexadecimales).


```
.bi::before {
  display: inline-block;
  content: "";
  vertical-align: -.125em;
  background-image: url("data:image/svg+xml,<svg viewBox='0 0 16 16' fill='%233333' xmlns='http://www.w3.org/2000/svg'><path fill-rule='evenodd' d='M8 9.5a1.5 1.5 0 1 0 0-3 1.5 1.5 0 0 0 3 0 0 0 3 0 0 0 3z' clip-rule='evenodd'/></svg>");
  background-repeat: no-repeat;
  background-size: 1rem 1rem;
}
```
- Cuando no se especifican dimensiones mediante width y height en <svg>, el icono llenará el espacio disponible. El atributo es viewBox obligatorio si desea cambiar el tamaño de los iconos con background-size. Tenga en cuenta que el atributo xmlns es obligatorio.

© JMA 2015. All rights reserved

Accesibilidad

- Para íconos puramente decorativos, para que sean ignorado por los lectores de pantalla se debería agregar el atributo aria-hidden="true".
- De lo contrario, hay proporcionar una alternativa de texto adecuada. Según el método que esté usando para agregar los íconos y dónde los esté usando (por ejemplo, como imágenes independientes o como el único contenido de un botón o control similar), hay varios enfoques posibles:


```

<svg class="bi" ... role="img" aria-label="check">
  <use xlink:href="bootstrap-icons.svg#check"/>
</svg>
<button ... aria-label="Mute">
  <svg class="bi bi-volume-mute-fill" aria-hidden="true" ...>...</svg>
</button>
```
- Al usar archivos SVG con , es posible que los lectores de pantalla no los anuncien como imágenes u omitan la imagen por completo. Agregar un role="img" adicional en el evita cualquier problema.

© JMA 2015. All rights reserved

FORMULARIOS

© JMA 2015. All rights reserved

Formularios

- Se han reiniciado varios elementos de formulario para estilos básicos más simples. Estos son algunos de los cambios más notables:
 - `<fieldset>` no tienen bordes, padding o margen, por lo que pueden usarse fácilmente como contenedores para inputs individuales o grupos de inputs.
 - Los `<legend>`, como los fieldsets, también se han rediseñado para que se muestren como una especie de encabezado.
 - `<label>` se establece en `display: inline-block` para permitir que se aplique margin.
 - Los `<input>`, `<select>`, `<textarea>` y `<button>` eliminan su margin y establecen `line-height: inherit` también.
 - `<textarea>` se modifica para que solo se pueda cambiar el tamaño verticalmente, ya que el cambio de tamaño horizontal a menudo “rompe” el diseño de la página.
- Los controles de formulario de Bootstrap se expanden con clases sobre los estilos reiniciados para diseñar pantallas personalizadas con una representación más consistente en todos los navegadores y dispositivos. Dado que se han reiniciado los estilos, asegúrate de utilizar las clases `.form-*` para recuperar el estilo apropiado.

© JMA 2015. All rights reserved

Formularios

```
<form class="border rounded-1 bg-info-subtle p-3 m-auto">
  <div class="mb-3">
    <label for="txtUsuario" class="form-label">Correo electrónico</label>
    <input type="email" class="form-control" id="txtUsuario" aria-
      describedby="userHelp">
    <div id="userHelp" class="form-text">Su usuario.</div>
  </div>
  <div class="mb-3">
    <label for="txtPassword" class="form-label">Contraseña</label>
    <input type="password" class="form-control" id="txtPassword">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="chkSesion" value="on">
    <label class="form-check-label" for="chkSesion">Recuérdame</label>
  </div>
  <button type="submit" class="btn btn-primary">Enviar</button>
</form>
```

© JMA 2015. All rights reserved

Layout

- Cada grupo de campos de formulario debe residir en un elemento `<form>`. Bootstrap no proporciona un estilo predeterminado para el elemento `<form>`, pero hay algunas características potentes del navegador que se proporcionan de forma predeterminada.
 - Dado que Bootstrap aplica `display: block` y `width: 100%` a casi todos los controles de formulario, los formularios se apilarán verticalmente de manera predeterminada. Se pueden usar clases adicionales para variar este diseño según el formulario.
 - Los formularios se comportan como cuadrículas por lo que no necesitan ser marcados como `.container` para contener filas y columnas.
 - Si `<button>` está dentro de un `<form>` por defecto es `type="submit"`, así que se debe ser específico e incluir siempre un `type`.
 - Se puede deshabilitar cada elemento de formulario dentro de un formulario con el atributo `disabled` en `<form>`.
- Las utilidades de margen son la forma más fácil de agregar estructura a los formularios, proporcionando una agrupación básica de etiquetas, controles, texto de formulario y mensajes de validación de formulario. Se recomienda apegar solo `.mb-*` (`margin-bottom`) y usar una sola dirección en todo el formulario para mantener la coherencia. Se pueden construir formularios más complejos utilizando las clases `.row` y `.col-*` para diseños de formularios que requieran varias columnas, anchos variados y opciones de alineación adicionales.

© JMA 2015. All rights reserved

Layout

- Para crear diseños más complejos con el sistema de cuadrícula:


```
<form class="row g-3">
  <div class="col-md-6">
    <label for="inputEmail4" class="form-label">Email</label>
    <input type="email" class="form-control" id="inputEmail4">
  </div>
  <div class="col-md-6">
    <label for="inputPassword4" class="form-label">Contraseña</label>
    <input type="password" class="form-control" id="inputPassword4">
  </div>
</form>
```
- Se pueden crear formularios horizontales con la cuadrícula agregando la clase `.row` para formar grupos y usando las clases `.col-*-*` para especificar el ancho de las etiquetas y controles. Hay que agregar `.col-form-label` a los `<label>` para que estén centrados verticalmente con los controles de formulario asociados.


```
<form>
  <div class="row mb-3">
    <label for="inputEmail3" class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10"><input type="email" class="form-control" id="inputEmail3"></div>
  </div>
```

© JMA 2015. All rights reserved

Layout

- El tamaño de las columnas en el formulario horizontal se establece con las unidades de `.col-*` o con `col-auto`.
- Para asegurar el tamaño vertical de las etiquetas en el formulario horizontal hay que usar `.col-form-label-sm` o `.col-form-label-lg` en los `<label>` o `<legend>` para adecuarse al tamaño de `.form-control-lg` y `.form-control-sm`.
- Los formularios en línea, para crear diseños horizontales adaptativos, utilizan las clases `.row-cols-*`, modificadores de gutters (horizontales y verticales), `.col-12` ayuda a apilar los controles de formulario (en viewports móviles estrechos), `.align-items-center` alinea los elementos del formulario en el medio (haciendo que `.form-checkbox` se alinee correctamente) y más.


```
<form class="row row-cols-sm-auto g-3 align-items-center">
  <div class="col-12">
    <label class="visually-hidden" for="txtUsuario">Nombre de usuario</label>
    <input type="text" class="form-control" id="txtUsuario"
      placeholder="Nombre de usuario">
  </div>
```

© JMA 2015. All rights reserved

Etiquetas

- Los `<label>` deben tener `.form-label` para establecer el estilo por defecto. Para adecuar su tamaño en los formularios horizontales, se sustituyen por `.col-form-label`, `.col-form-label-sm` (pequeño) o `.col-form-label-lg` (grande).

```
<div class="mb-3">
  <label for="txtUsuario" class="form-label">Correo electrónico</label>
  <input type="email" class="form-control" id="txtUsuario">
</div>
```
- Las etiquetas flotantes se muestran como un placeholder o flotando sobre el contenido de los campos de entrada cuando se rellena el contenido. Para crear una etiqueta flotante se marca con `.form-floating` al contenedor y, dentro, el control requiere un placeholder (se usa el `:placeholder-shown`) y debe preceder al `<label>` (se utiliza un selector de hermanos) que no requiere `.form-label`.

```
<div class="mb-3 form-floating">
  <input type="password" class="form-control" id="password" placeholder=" ">
  <label for="password">Contraseña</label>
</div>
```

© JMA 2015. All rights reserved

Controles de formulario

- Los `<input>` y `<textarea>` deben tener `.form-control` para establecer el estilo por defecto. Añadir `.form-control-sm` (pequeño) o `.form-control-lg` (grande) permite establecer la altura.
- El atributo `disabled` elimina los eventos de puntero y le da una apariencia atenuada con fondo. El atributo `readonly` impide la modificación de su valor y le da una apariencia con fondo, aunque con `.form-control-plaintext` le da una apariencia de texto sin formato (elimina el estilo de campo conservando el margen y padding correctos).

```
<input type="text" readonly class="form-control-plaintext" id="userId" value="email@example.com">
```
- En los `type="file"` la apariencia agrupa el botón con el campo de texto. En los `type="color"` hay que añadir `.form-control-color` para obtener la apariencia adecuada.

```
<input type="color" class="form-control form-control-color" id="color" value="#56d07c" title="Elige color">
```

© JMA 2015. All rights reserved

Controles de formulario

- Las casillas de verificación (`type="checkbox"`) y botones de radio (`type="radio"`) predeterminadas se reemplazan con contenedores `.form-check` y una serie de clases que mejoran el diseño y el comportamiento, brindando una mayor personalización y consistencia entre navegadores.
- El `<label>` de la casilla de verificación o del botón de radio se marca con `.form-check-label` y, por el localizador utilizado, debe ir a continuación del `<input>` marcado con `.form-check-input`. Para mostrar el `<label>` antes del `<input>` se agrega `.form-check-reverse` al contenedor.


```
<div class="mb-3 form-check">
  <input type="checkbox" class="form-check-input" id="chkSesion" value="on">
  <label class="form-check-label" for="chkSesion">Recuérdame</label>
</div>
```
- Cuando se configuran manualmente a través de JavaScript (no hay un atributo HTML disponible para especificarlo), las casillas de verificación pueden utilizar un tercer estado (`:indeterminate`).
- El atributo `disabled` elimina los eventos de puntero y le da una apariencia atenuada tanto al `<input>` como al `<label>` asociado.

© JMA 2015. All rights reserved

Controles de formulario

- De forma predeterminada, cualquier número de casillas de verificación y botones de radio que sean hermanos inmediatos se apilarán verticalmente y se espaciarán adecuadamente con `.form-check`. Agregando `.form-check-inline` a cualquier `.form-check` se agruparán en la misma fila horizontal.
- Un switch representa como un interruptor una casilla de verificación agregando `.form-switch` al `.form-check` (es conveniente usar `role="switch"` en el `<input>`). También admiten el atributo `disabled`.
- Las casillas de verificación y botones de radio se pueden representar como botones toggle (sin presionar/presionado). Para ello se sustituye `.form-check-input` por `.btn-check` en el `<input>` y, en el `<label>`, `.form-check-label` por `.btn` (se le puede agregar `btn-{color}` o `btn-outline-{color}`). Los botones toggle se pueden agrupar en un grupo de botones si es necesario. También admiten el atributo `disabled`.

© JMA 2015. All rights reserved

Controles de formulario

- En los `type="color"` hay que sustituir `.form-control` por `.form-range` que está diseñado para que la pista (el fondo) y el thumb (el valor) aparezcan iguales en todos los navegadores. También admite el atributo `disabled`.

```
<div class="col-12">
  <label for="nota" class="form-label">Calificación</label>
  <input type="range" class="form-range" min="0" max="10" step="0.5"
value="3.5" id="nota">
</div>
```
- Los `<select>` solo necesitan `.form-select` para activar los estilos personalizados. Los estilos personalizados están limitados a la apariencia inicial de `<select>` y no pueden modificar los `<option>` debido a las limitaciones del navegador. También se puede añadir `.form-select-sm` o `.form-select-lg` para que coincidan con el tamaño del resto de entradas de texto similares.
- El estilo personalizado es compatible con los atributos `size` (listbox) y `multiple` (selección múltiple). El atributo `disabled` elimina los eventos de puntero y le da una apariencia atenuada con fondo.

© JMA 2015. All rights reserved

Grupos de entradas

- Los grupos de entradas permiten colocar e integrar a cada lado de un `<input>` o `<select>` uno o varios complementos: texto (puede incluir casillas de verificación y botones de radio), botones (simples, grupos de botones, de selección, ...) y otros `<input>`.
- El contenedor se marca como `.input-group` (y con `.input-group-sm` y `.input-group-sm`) y el complemento con `.input-group-text` o `.btn`.

```
<div class="input-group mb-3">
  <div class="input-group-text">
    <input class="form-check-input mt-0" type="checkbox" value="on"
aria-label="Seleccionar importe">
  </div>
  <input type="text" class="form-control" aria-label="importe">
  <span class="input-group-text">€</span>
  <button class="btn btn-success" type="button">Enviar</button>
</div>
```

© JMA 2015. All rights reserved

Validación

- Todos los navegadores modernos admiten la API de validación de restricciones del HTML5, una serie de métodos de JavaScript para validar controles de formulario. Se puede proporcionar mensajes de validación personalizados en JavaScript con `setCustomValidity`. Los mensajes de feedback y su estilo de retroalimentación son diferentes para cada navegador y sistema operativo, sin permitir estilo a través de CSS aunque establece las pseudoclases de CSS `:invalid` y `:valid` en los elementos `<input>`, `<select>` y `<textarea>`.
- Bootstrap implementa una serie de clase con estilos personalizados para dar el feedback de la validación de los formularios de una forma unificada para todos los navegadores. Los estilos de feedbacks personalizados aplican colores, bordes, estilos de enfoque e iconos de fondo personalizados para comunicar mejor los feedbacks. Para que no interfiera el feedback predeterminados del navegador se debe desactivar con el atributo `novalidate` de `<form>` (deshabilita las notificaciones no las validaciones y sigue aplicando los estilos `:invalid` y `:valid`).
- Los estilos de validación están disponibles para los siguientes controles y componentes de formulario:
 - `<input>` y `<textarea>` con `.form-control` (incluyendo hasta un `.form-control` en `.input-group` pero hay que agregar `.has-validation` al `.input-group` para evitar problemas con los bordes)
 - `<select>` con `.form-select`
 - `.form-checks`

© JMA 2015. All rights reserved

Validación

- El feedback es una etiqueta con el texto y marcada con `.valid-feedback` (positivo) o `.invalid-feedback` (negativo) que se muestra u oculta en función al `:valid` o `:invalid` de la etiqueta anterior, por lo que el feedback debe ir a continuación del control validado que informa.
- Bootstrap aplica los estilos `:invalid` y `:valid` si un contenedor, generalmente el `<form>`, tiene la clase `.was-validated`. De lo contrario, cualquier campo obligatorio sin valor aparecerá como no válido al cargar la página y de esta forma se puede elegir cuándo activarlos (por lo general, después de intentar enviar el formulario) o desactivarlos (eliminando `.was-validated` del `<form>`).
- Para la validación del lado del servidor se puede indicar los campos del formulario válidos e inválidos con `.is-valid` e `.is-invalid` que también muestran u ocultan los `.valid-feedback` o `.invalid-feedback`. A estar ya establecidos, no necesitan `.was-validated` pero tampoco cambian dinámicamente en el cliente.

© JMA 2015. All rights reserved

Validación

- Se recomienda utilizar la validación del lado del cliente, en cuyo caso se puede dejar activado el `.was-validated`

```
<form class="was-validated">
  <div class="mb-3 form-floating">
    <input type="password" class="form-control" id="txtPassword"
      placeholder=" " required>
    <label class="form-label" for="txtPassword">Contraseña</label>
    <div class="valid-feedback">Correcto</div>
    <div class="invalid-feedback">Es obligatorio</div>
  </div>
```
- Si el diseño del formulario lo permite, se puede intercambiar las clases `.{valid|invalid}-feedback` por las clases `.{valid|invalid}-tooltip` para mostrar los comentarios de validación en un con estilo tooltip. Hay que asegurarse de tener un contenedor con `.position-relative` para el correcto posicionamiento de los tooltip.

© JMA 2015. All rights reserved

Accesibilidad

- Todos los controles de un formulario deben tener un nombre accesible adecuado para que su propósito pueda transmitirse a los usuarios de tecnologías de asistencia.
- La forma más sencilla de lograr esto es usar una etiqueta `<label>` con el atributo `for` con el id del control. En el caso de los `<button>` se debe incluir texto suficientemente descriptivo como parte del contenido.
- Para situaciones en las que no es posible incluir un `<label>` visible o contenido de texto apropiado, existen formas alternativas de proporcionar un nombre accesible, como:
 - Establecer explícitamente el nombre accesible usando el atributo `aria-label`
 - Apuntando con el atributo `aria-labelledby` al id un elemento existente que puede actuar como texto apropiado
 - Usar elementos `<label>` ocultos usando la clase `.visually-hidden`
 - Proporcionar un atributo `title` con el nombre
- Si ninguno de estos están presentes, las tecnologías de asistencia pueden recurrir al uso del atributo `placeholder` como respaldo para el nombre accesible en los elementos `<input>` y `<textarea>`.

© JMA 2015. All rights reserved

<http://getbootstrap.com/components/>

COMPONENTES

© JMA 2015. All rights reserved

Componentes

Alerts: Proporciona mensajes de retroalimentación contextuales para las acciones típicas de los usuarios con mensajes de alerta disponibles y flexibles.

Popovers: Notificación emergente como los que se encuentran en iOS.

Toasts: Notificaciones livianas diseñadas para imitar las notificaciones automáticas que se han popularizado en los sistemas operativos móviles.

Tooltips: Notificación emergente al pasar el ratón por encima.

Modal: Complemento modal del JavaScript de Bootstrap para agregar cuadros de diálogo con lightboxes, notificaciones de usuario o contenido completamente personalizado.

Cards: Las tarjetas brindan un contenedor de contenido flexible y extensible con múltiples variantes y opciones.

Carousel: Un componente de presentación de diapositivas para recorrer elementos (imágenes o diapositivas de texto) como un carrusel.

Accordion: Crea acordeones que despliegan u contraen verticalmente el contenido en combinación con el complemento JavaScript Collapse.

© JMA 2015. All rights reserved

Componentes

Collapse: Alterna la visibilidad del contenido con algunas clases y los complementos de JavaScript.

Dropdowns: Listas desplegables que alternna la visibilidad de superposiciones contextuales para mostrar listas de enlaces y más.

Badges: Pequeñas insignias de conteo y etiquetado.

List group: Los grupos de listas son un componente flexible y potente para mostrar una serie de contenidos.

Buttons: Estilos de botones personalizados para acciones en formularios, cuadros de diálogo y más con soporte para múltiples tamaños, estados y más.

Button group: Agrupa una serie de botones en una sola línea o apíalos en una columna vertical.

Close button: Botón de cierre genérico para descartar contenido modals y alerts.

Navs & tabs: Barra de navegación simple, horizontal o vertical, que se puede mostrar como pestañas, pastillas y otros.

© JMA 2015. All rights reserved

Componentes

Navbar: Barra de navegación principal para la creación de cabeceras adaptativas que incluye soporte para marca, navegación y más, incluido soporte para el complemento Collapse.

Offcanvas: Crea barras de navegación laterales ocultas en el proyecto, carritos de compras y más, con algunas clases y el complemento de JavaScript.

Breadcrumb: Miguitas de pan para indica la ubicación de la página actual dentro de una jerarquía de navegación.

Pagination: Barra de paginación para indicar que existe una serie de contenido relacionado en varias páginas.

Scrollspy: Sincronizador automático de la posición de desplazamiento con la navegación o un grupo los componentes.

Placeholders: Marcadores de posición para indicar que algo se esta cargando.

Spinners: Indicador animado del estado de carga de un componente o una página, creados completamente con HTML, CSS y sin JavaScript.

Progress: Barra de progreso con soporte para barras apiladas, fondos animados y etiquetas de texto.

© JMA 2015. All rights reserved

JAVASCRIPT

© JMA 2015. All rights reserved

Introducción

- El CSS, aun con las ampliaciones de las últimas versiones, se limita a controlar la estética. Bootstrap utiliza complementos opcionales en JavaScript para darle vida.
- Los complementos se pueden incluir individualmente (usando archivos `js/dist/*.js` individuales de Bootstrap), o todos a la vez usando `bootstrap.js` o `bootstrap.min.js`. Algunos complementos y componentes CSS dependen de otros complementos. Si se incluyen los plugins individualmente, asegúrate de comprobar si existen estas dependencias en los documentos.
- Los empaquetadores (Webpack, Rollup...) pueden usar los archivos `/js/dist/*.js` que están preparados para UMD. La versión de Bootstrap creada como ESM permite usar Bootstrap (`bootstrap.esm.js` y `bootstrap.esm.min.js`) como un módulo en el navegador, si los admite.
- Las versiones anteriores de Bootstrap tenían una dependencia de jQuery, ahora Bootstrap está diseñado para usarse sin jQuery, pero aún es posible usarlo con jQuery. Si Bootstrap detecta jQuery en el objeto `window` agregará todos sus componentes en el sistema de complementos de jQuery.

© JMA 2015. All rights reserved

Atributos y eventos

- Casi todos los complementos de Bootstrap se pueden habilitar y configurar solo a través de atributos de datos (data-) de HTML (forma preferida de usar la funcionalidad de JavaScript). Solo se debe usar un solo conjunto de atributos de datos en un elemento (no se puede activar un tooltip y un modal desde el mismo botón).
- Actualmente, para consultar elementos DOM, Bootstrap usa los métodos nativos `querySelector` y `querySelectorAll` por razones de rendimiento. Se deben usar selectores CSS válidos en los atributos de datos y constructores.
- Bootstrap proporciona eventos personalizados para la mayoría de las acciones únicas de los plugins. Por lo general, estos vienen en forma de infinitivo y participio pasado, donde el infinitivo (p. ej., `show`) se activa al comienzo de un evento, y su forma de participio pasado (p. ej., `shown`) se activa al finalizar un evento. Todos los eventos en infinitivo proporcionan la funcionalidad `preventDefault()`. Esto proporciona la capacidad de detener la ejecución de una acción antes de que comience. Devolver `false` desde un controlador de eventos también llamará automáticamente a `preventDefault()`.

© JMA 2015. All rights reserved

API programática

- Todos los constructores de los complementos aceptan un objeto de opciones o nada (lo cual inicia un complemento con su comportamiento predeterminado):

```
var modal = new bootstrap.Modal( document.getElementById('myModal'), {
  keyboard: false })
```
- Si se desea obtener una instancia de complemento en particular, cada complemento expone un método `getInstance` que devuelve `null` si no se inicia una instancia:

```
var pop = bootstrap.Popover.getInstance(myPopoverEl)
```
- Alternativamente, `getOrCreateInstance` se puede usar para asociar la instancia con un elemento DOM, o crear uno nuevo en caso de que no se haya inicializado:

```
bootstrap.Popover.getOrCreateInstance(myPopoverEl, configObject)
```
- También se puede usar un selector CSS como primer argumento en lugar de un elemento DOM para inicializar el complemento.

```
var modal = new bootstrap.Modal('#myModal')
var sel= new bootstrap.Dropdown('[data-bs-toggle="dropdown"]')
```

© JMA 2015. All rights reserved

API programática

- Todos los métodos programáticos de la API son asíncronos y regresan al de la llamada una vez que se inicia la transición, pero antes de que finalice.
- Para ejecutar una acción una vez completada una transición, se debe escuchar el evento correspondiente.


```
const myModal = document.querySelector('#myModal')
myModal.hide()
myModal.addEventListener('shown.bs.hidden', event => {
  myModal.dispose()
})
```
- Tooltips y Popovers utilizan un sanitizer (desinfectante) incorporado para desinfectar (posible código malicioso) las opciones que aceptan HTML.

© JMA 2015. All rights reserved

Métodos y propiedades

- Cada complemento de Bootstrap expone los siguientes métodos y propiedades estáticas.
- Métodos:
 - getInstance: Método estático que le permite obtener la instancia modal asociada con un elemento DOM.
 - getOrCreateInstance: Método estático que le permite obtener la instancia modal asociada con un elemento DOM, o crear una nueva en caso de que no se haya inicializado.
 - dispose: Destruye el modal de un elemento. (Elimina los datos almacenados en el elemento DOM)
- Propiedades estáticas:
 - NAME: Devuelve el nombre del complemento. (Ejemplo: bootstrap.Tooltip.NAME)
 - VERSION: Se puede acceder a la versión de cada uno de los complementos de Bootstrap a través de la propiedad VERSION del constructor del complemento (Ejemplo: bootstrap.Tooltip.VERSION)

© JMA 2015. All rights reserved

JavaScript deshabilitado

- Muchos complementos de Bootstrap utilizan JavaScript para darle vida o establecer su comportamiento.
- Los complementos de Bootstrap no tienen un respaldo especial cuando JavaScript está deshabilitado.
- Si le preocupa la experiencia del usuario cuando JavaScript está deshabilitado, se debe utilizar `<noscript>` para explicar la situación (y cómo volver a habilitar JavaScript) a sus usuarios.