

Lab3 - CSS avanzado (Bootstrap)

Pepe Calderón, Fernando Sola, Daniel Ayala, Inma Hernández, Margarita Cruz, Carlos Arévalo, David Ruiz



Escuela Técnica Superior de
Ingeniería Informática

Índice

1. Objetivo	1
2. Introducción	2
3. Importando Bootstrap en nuestro proyecto	3
4. Sistema de filas y columnas	4
5. Componentes Bootstrap relevantes	8
5.1. Navbar: la barra de navegación, revisada	8
5.2. Formularios	10
5.3. Botones	13
5.4. Etiquetas	14
5.5. Listas	14
5.6. Tarjetas	15
6. Ejemplos de vistas de la aplicación	17
6.1. Listado de fotos más recientes	17
6.2. Detalle de foto	20
6.3. Registro de usuario	23
7. Actualización en GitHub	27
8. Anexo I: Iconos Font Awesome	28

Objetivo

El objetivo de esta práctica es introducir al alumno al framework CSS Bootstrap, para diseñar aplicaciones Web responsivas y con un nivel de acabado visual satisfactorio. El alumno aprenderá a:

- Integrar el framework Bootstrap en un proyecto ya existente.
- Comprender y usar la organización básica de Bootstrap en filas y columnas.
- Conocer y emplear los elementos que provee Bootstrap para los componentes más habituales de una aplicación web.
- Leer y asimilar documentación técnica para poder emplear nuevos elementos Bootstrap no cubiertos en esta práctica.

Introducción

Bootstrap fue creado en 2011 por Mark Otto y Jacob Thornton, desarrolladores de interfaz de usuario en Twitter como un medio para acelerar la creación de interfaces web de calidad. Es de código abierto y su licencia permite un uso libre, tanto privado como con ánimo de lucro. El repositorio en GitHub de Bootstrap es el tercero más popular en número de estrellas relativo a tecnologías Web, sólo por detrás de los frameworks JavaScript Vue y React.

Gran parte del motivo de su popularidad reside en que permite hacer interfaces web de gran calidad visual con mucho menor esfuerzo que escribiendo manualmente todo el código CSS requerido. Además, está pensado para ser *mobile-first*, esto es, que su sistema de desarrollo de interfaces prima las pantallas pequeñas de los móviles, y los navegadores más grandes escalan a partir de éstas. Usando Bootstrap, se pueden hacer interfaces adaptables tanto a navegadores de escritorio como a pequeñas pantallas de móviles sin esfuerzo adicional.

Importando Bootstrap en nuestro proyecto

Para usar Bootstrap es necesario incorporar a nuestro proyecto una serie de archivos CSS y JavaScript. Estos pueden ser referenciados online, pero la plantilla de proyecto Silence descargada ya los incorpora. Añadiremos las siguientes líneas a nuestro `<head>` :

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="/css/bootstrap.min.css">
<link rel="stylesheet" href="/css/font-awesome.min.css">
<script src="/js/libs/bootstrap.min.js"></script>
```

Incorporaremos estas dependencias primero al archivo `index.html` , en el que trabajaremos en las siguientes secciones. Es de esperar que el estilo de la página cambie, ya que se han incorporado las hojas de estilo propias de Bootstrap. A continuación aprenderemos a usar los estilos que definen éstas en nuestros elementos.

Sistema de filas y columnas

Cuando usamos Bootstrap, el contenido de una vista se organizará en filas. Cada fila puede ser a su vez subdividida en un máximo de 12 columnas, que por defecto tendrán el mismo ancho. Por ejemplo, si definimos una fila con una sola columna, obtendremos un elemento con un ancho del 100% de la fila. Si definimos dos columnas en una fila, cada una tendrá el 50% del ancho de la fila, y así sucesivamente.

Para empezar a definir contenido, se debe crear un `<div>` padre que contendrá todo el contenido de la vista. Esta etiqueta debe tener clase `container` :

```
<div class="container">
  <!-- Cuerpo de la vista aqui -->
</div>
```

A continuación, definiremos las siguientes filas en las que organizar los elementos. Las filas son tan anchas como su elemento padre (en este caso, tan anchas como la ventana en sí) y su altura está determinada por los elementos que contienen. Para definir una fila, crearemos un `<div>` con clase `row` :

```
<div class="container">
  <div class="row">
    Contenido de la fila aqui
  </div>
</div>
```

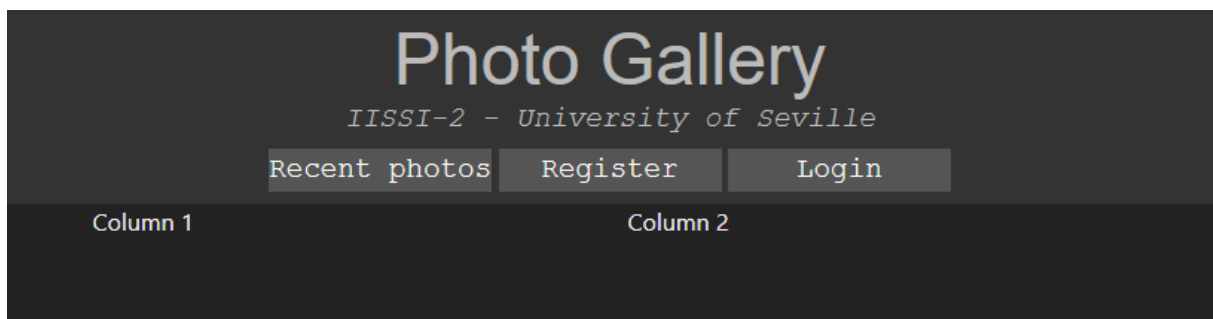
Finalmente, dentro de una fila podemos definir tantas columnas como deseemos, hasta 12. Por defecto, cada una de las columnas tendrá el mismo ancho, ocupando toda la fila. Las

4. Sistema de filas y columnas

columnas se definen mediante `<div>` s con clase `col-md` . Podemos probar introduciendo el siguiente `<div>` en nuestra vista, después de la cabecera y antes del resto del contenido:

```
<div class="container">
  <div class="row">
    <div class="col-md">
      Column 1
    </div>
    <div class="col-md">
      Column 2
    </div>
  </div>
</div>
```

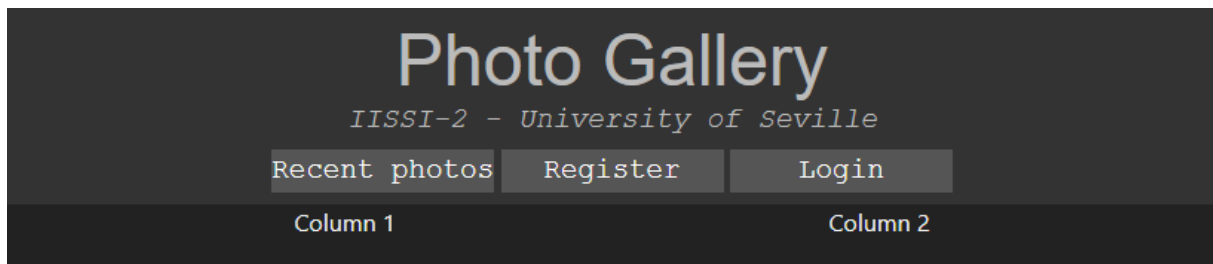
El resultado del código anterior es:



Como se aprecia, el contenido aparece en dos columnas. Además, Bootstrap añade márgenes automáticos entre el elemento raíz () y su contenido. Estos márgenes varían automáticamente según el tamaño de la ventana, por lo que se recomienda no modificarlos.

Sin embargo, dentro de cada columna, el texto aparece alineado a la izquierda. Si deseamos centrar el texto del interior de un elemento, podemos añadir a los elementos en cuestión la clase `text-center` , también provista por Bootstrap. Otras clases posibles para alinear texto dentro de un elemento son `text-start` (izquierda) y `text-end` (derecha). Así, resultaría:

```
<div class="container">
  <div class="row text-center">
    <div class="col-md">
      Column 1
    </div>
    <div class="col-md">
      Column 2
    </div>
  </div>
</div>
```

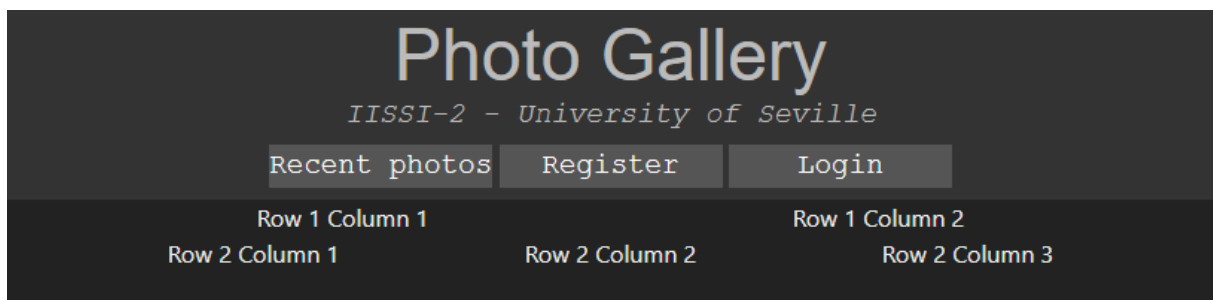


Observe como la clase `text-center`, aplicada a toda la fila, centra los elementos de todas las columnas de dicha fila. Si se desea un ajuste más fino, se puede aplicar la clase de alineamiento de texto deseada a cada columna.

Podemos añadir más filas, cada una con sus propias columnas. No es necesario que todas las filas tengan el mismo número de columnas: si añadiésemos una segunda fila con tres columnas, el resultado sería:

```
<div class="container">
  <div class="row text-center">
    <div class="col-md">
      Row 1 Column 1
    </div>
    <div class="col-md">
      Row 1 Column 2
    </div>
  </div>

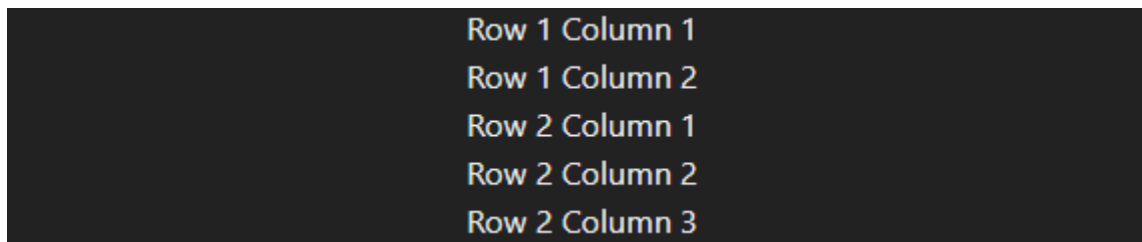
  <div class="row text-center">
    <div class="col-md">
      Row 2 Column 1
    </div>
    <div class="col-md">
      Row 2 Column 2
    </div>
    <div class="col-md">
      Row 2 Column 3
    </div>
  </div>
</div>
```



4. Sistema de filas y columnas

Como se aprecia, el número de columnas es diferente en las dos filas, aunque la separación entre filas puede ser insuficiente. Para ello, podemos aplicar código CSS personalizado para adaptar el margen inferior y superior de cada elemento `div.row` si así lo deseamos.

Como se explicó anteriormente, una de las principales ventajas de Bootstrap es que la adaptación a dispositivos móviles no requiere un esfuerzo adicional. Si se reduce el ancho de la ventana lo suficiente, las columnas pasan a apilarse verticalmente para adaptarse a pantallas más estrechas:



Las columnas definidas mediante la clase `col-md` se apilan verticalmente cuando la anchura del elemento padre es menor de 768px. Existen otras clases para definir columnas, como `col-sm` y `col-lg`, que requieren una anchura mínima menor y mayor respectivamente para pasar a apilar verticalmente los elementos.

También se pueden hacer columnas de tamaño asimétrico, para lo que se debe indicar su anchura relativa de 1 a 12 mediante la clase `col-md-X`, donde X es el valor de anchura. Una columna de anchura 6 ocupará la mitad de la fila, una de anchura 9 ocupará el 75%, etcétera. El resto de columnas de la misma fila se redimensionan automáticamente para completar la fila. Por ejemplo, si deseamos dedicar el 75% del ancho de una fila a un elemento y el restante a otro:

```
<div class="container">
  <div class="row text-center">
    <div class="col-md-9">
      Wide column (75%)
    </div>
    <div class="col-md">
      This column takes up the remaining space
    </div>
  </div>
</div>
```



Existe una gran cantidad de opciones adicionales que permiten mucha más flexibilidad y que, por brevedad, no se incluyen en este boletín, pero que pueden consultarse en la página relevante de la documentación de Bootstrap.

Componentes Bootstrap relevantes

Bootstrap nos provee una gran cantidad de componentes de uso muy común en interfaces: desde botones, listas y desplegables hasta carruseles de imágenes, índices de paginación y mensajes de confirmación. La lista completa de estos componentes, junto con instrucciones para su uso y ejemplos, puede encontrarse en la documentación. En las siguientes subsecciones se mostrarán algunos de los más relevantes para nuestra aplicación.

5.1. Navbar: la barra de navegación, revisada

En la práctica anterior, creamos una barra de navegación simple mediante CSS personalizado, pero Bootstrap nos provee un componente llamado Navbar destinado a cubrir específicamente este propósito.

La barra de navegación más sencilla posible incluye solamente los elementos relevantes de navegación, y se puede construir con el siguiente código HTML:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navigation index</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
```

```

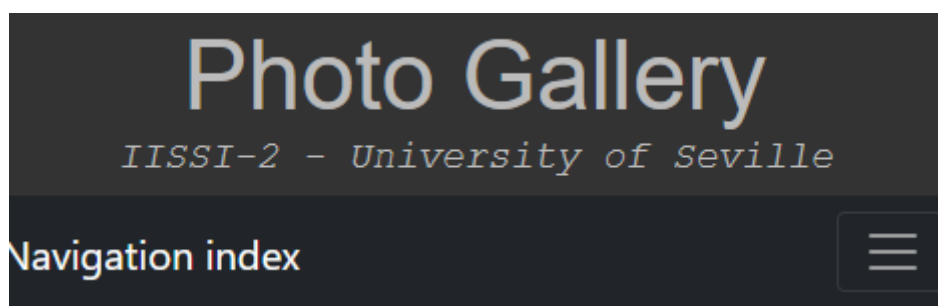
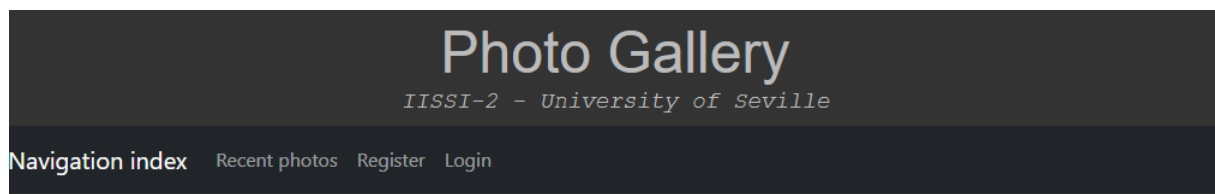
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
  <li class="nav-item">
    <a class="nav-link" href="index.html">Recent photos</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="register.html">Register</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="login.html">Login</a>
  </li>
</ul>
</div>
</div>
</nav>

```

Observe cómo se le ha dado un tono oscuro a la barra de navegación añadiéndole las clases `navbar-dark` y `bg-dark`. Si se deseara tener la misma en tonos claros, se pueden usar las clases `navbar-light` y `bg-light`.

Como suele ser habitual al usar este tipo de frameworks, la estructura del código HTML viene impuesta y es más compleja que hacerlo a mano. Esta es una de las principales desventajas de las librerías CSS como Bootstrap. En este caso, los elementos de la barra de navegación se encuentran dentro de un `<ul class="navbar-nav">`.

Sin embargo, este esfuerzo se ve recompensado con una barra de navegación de un estilo más uniforme y que, además, se colapsa automáticamente al hacer más estrecha la ventana, para adaptarse a móviles:



Además, se pueden añadir elementos colapsables, para agrupar varios elementos de la barra en uno sólo. Para ello, en lugar de añadir un `<li class="nav-item">` como los anteriores, añadiremos el siguiente elemento:

```

<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"

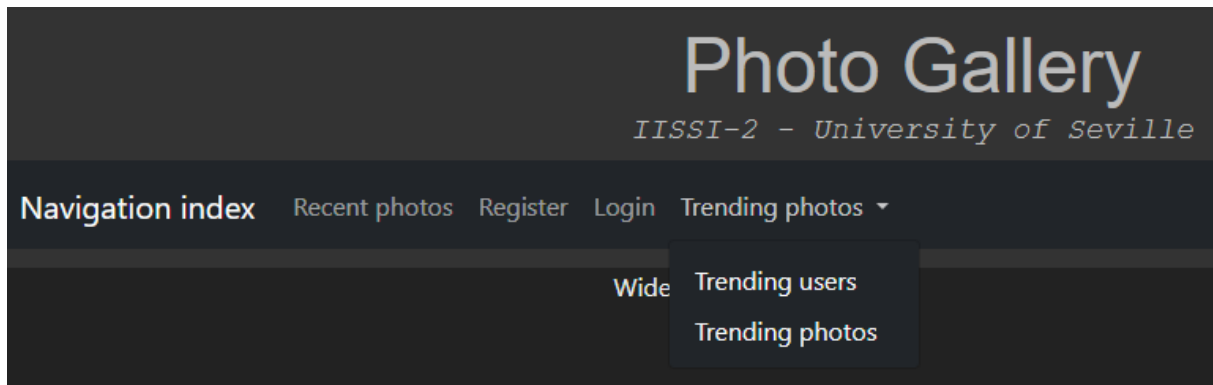
```

```

role="button" data-bs-toggle="dropdown"
  aria-expanded="false">
  Trending photos
</a>
<ul class="dropdown-menu bg-dark" aria-labelledby="navbarDropdown">
  <li><a class="dropdown-item text-light"
href="trending_users.html">Trending users</a></li>
  <li><a class="dropdown-item text-light"
href="trending_photos.html">Trending photos</a></li>
</ul>
</li>

```

En cada caso, deberemos adaptar los ID para que sean únicos, y modificar los enlaces de las etiquetas `<a>` para que referencien a los archivos HTML adecuados dentro de nuestra aplicación. Al igual que antes, la mayor complejidad en el código HTML se recompensa con un elemento de mayor calidad visual. Así, la nueva barra de navegación tras añadir este elemento resulta:



Existe una gran cantidad de opciones adicionales de personalización de la Navbar, que se pueden encontrar en la documentación correspondiente.

5.2. Formularios

```
{: .warning }
```

i Info

Recuerde: Cualquier contenido que use elementos Bootstrap en el cuerpo de la página (filas, columnas, componentes...) debe estar dentro de un Bootstrap también da formato a formularios siempre que éstos tengan un formato concreto. Un ejemplo de formulario simple en Bootstrap es el siguiente:

```

<form>
  <div class="form-group">
    <label for="username-input">Username:</label>
    <input type="text" class="form-control" id="username-input"
name="username" placeholder="Username">
  </div>
  <div class="form-group">
    <label for="password-input">Password:</label>
    <input type="password" class="form-control" id="password-input"
name="password" placeholder="Password">
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>

```

Username:

Password:

Los grupos de `<label>` e `<input>` deben ser agrupados mediante un `<div>` con clase `form-group`. Además, los `<input>` pueden contener un atributo `placeholder` para mostrar un texto por defecto a modo de ayuda cuando éstos estén vacíos. La clase `form-control` les da a los elementos de entrada el estilo propio de Bootstrap, mientras que el resto de atributos son los propios de cualquier elemento `<input>` que ya conocemos.

Observe que el formato de colores de los `<input>` no es oscuro, ya que Bootstrap define sus propios estilos para ellos que toman preferencia sobre los que definimos nosotros previamente. Podemos hacer que pasen a un tema oscuro añadiéndole las clases `bg-dark text-light` a los `<input>`:

El color del botón puede ser modificado mediante su clase (ver la documentación de Bootstrap respecto a colores). Además, los campos del formulario pueden mostrarse en cualquier

estructura deseada usando el sistema de filas y columnas de Bootstrap, siempre que todos los elementos se encuentren dentro de la etiqueta `<form>`.

Otros elementos de formulario relevantes se muestran a continuación:

Entradas de tipo `<select>` (por defecto son de opción única, si se añade a la etiqueta `<select>` el atributo `multiple` se convierte en un select de opción múltiple):

```
<div class="form-group">
  <label for="select-input">Select element from a list:</label>
  <select class="form-control" id="select-input" name="select">
    <option value="1">Option 1</option>
    <option value="2">Option 2</option>
    <option value="3">Option 3</option>
    <option value="4">Option 4</option>
    <option value="5">Option 5</option>
  </select>
</div>
```

```
{: .warning }
```

i Info

Recuerde que los atributos `for` de los `<label>` se asocian con los atributos `id` de los `<input>` para mejorar la accesibilidad, y que un `id` debe ser único en todo el documento.

Checkboxes (en este caso, el `<div>` de agrupación pasa de tener clase `form-group` a `form-check`):

```
<div class="form-check">
  <input class="form-check-input" type="checkbox" id="checkbox-id"
  name="checkbox">
  <label class="form-check-label" for="checkbox-id">
    Option with a checkbox
  </label>
</div>
```

Opciones con radio button (nótese cómo se indica que las opciones son mutuamente excluyentes dándoles el mismo atributo `name`):

```
<div class="form-check">
  <input class="form-check-input" type="radio" name="radio-button"
  value="1" checked>
  <label class="form-check-label">
    Option 1
  </label>
```

```

</div>
<div class="form-check">
  <input class="form-check-input" type="radio" name="radio-button"
value="2">
  <label class="form-check-label">
    Option 2
  </label>
</div>

```

Por defecto, los checkboxes y radios se muestran apilados verticalmente. Para mostrar varios uno al lado del otro, se puede usar la clase `form-check-inline` en lugar de `form-check`. Además, se puede indicar qué opción está seleccionada por defecto añadiendo el atributo `checked`.

La documentación detallada sobre formularios, junto con el resto de opciones de personalización disponibles, se encuentra disponible en la página de Bootstrap sobre formularios y elementos input.

5.3. Botones

Con Bootstrap se le puede dar formato de manera muy sencilla a cualquier botón, añadiéndole las clases `btn` y `btn-primary` (esta última determina su estilo de color):

```
<button type="button" class="btn btn-primary">Press me!</button>
```

Al igual que la mayoría de elementos Bootstrap, se puede cambiar el estilo de color del botón mediante las clases apropiadas. Los estilos disponibles son los siguientes:

```

<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

```

Se puede establecer un botón como desactivado añadiéndole el atributo `disabled`. Esto modificará su aspecto visual e impedirá que se pulse:

```
<button type="button" class="btn btn-primary">Enabled</button>
<button type="button" class="btn btn-primary" disabled>Disabled</button>
```



El resto de opciones relevantes se encuentran en la documentación del componente.

5.4. Etiquetas

Con el componente Badge de Bootstrap, se puede envolver un texto determinado en un elemento parecido a una etiqueta, para propósitos meramente estéticos:

```
This photo has the following categories:
<span class="badge bg-primary">Landscapes</span>
<span class="badge bg-primary">Nature</span>
```

This photo has the following categories: Landscapes Nature

Note que las etiquetas `<span>` sirven para contener texto, al igual que las etiquetas `<p>`. Sin embargo, a diferencia de las últimas, las primeras muestran el texto en la misma línea en lugar de separarlo en párrafos con saltos de línea.

Al igual que con los botones, se puede modificar su tema de color mediante las clases `primary`, `secondary`, `success`, etcétera. Además, se puede usar el mismo conjunto de clases `badge bg-*` con una etiqueta `<a href="...">` en lugar de `<span>`, para darle a un enlace aspecto de etiqueta. El resto de opciones relevantes se encuentran en la documentación del componente.

5.5. Listas

Las listas también se pueden formatear con Bootstrap añadiendo algunas clases definidas por el framework:

```
<ul class="list-group">
  <li class="list-group-item">Element 1</li>
  <li class="list-group-item">Element 2</li>
  <li class="list-group-item">Element 3</li>
  <li class="list-group-item">Element 4</li>
</ul>
```

Element 1
Element 2
Element 3
Element 4

Para darle un mayor grado de personalización, a los elementos de la lista se les puede añadir la clase `list-group-item-*` donde `*` es `primary`, `secondary`, `success`, etc.

```
<ul class="list-group">
  <li class="list-group-item list-group-item-primary">Element 1</li>
  <li class="list-group-item list-group-item-success">Element 2</li>
  <li class="list-group-item list-group-item-warning">Element 3</li>
  <li class="list-group-item list-group-item-danger">Element 4</li>
</ul>
```

Element 1
Element 2
Element 3
Element 4

El resto de opciones relevantes se encuentran en la documentación del componente.

5.6. Tarjetas

Las tarjetas o cards son una manera útil de agrupar varios elementos, como por ejemplo una imagen y texto variado. Una tarjeta sencilla se puede construir con Bootstrap mediante el siguiente código HTML:

```
<div class="card text-light bg-dark">
  

  <div class="card-body">
```

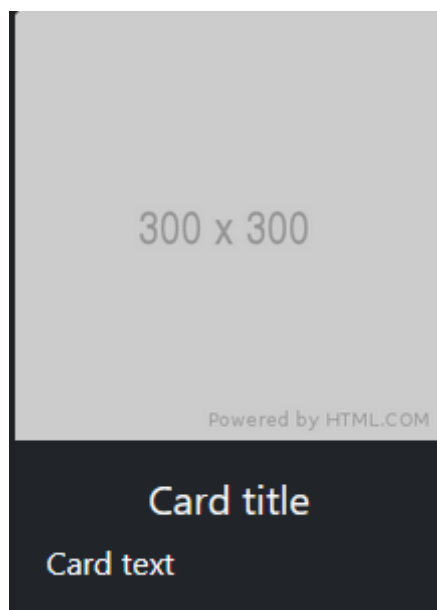
```
<h5 class="card-title text-center">Card title</h5>
<p class="card-text">Card text</p>
</div>
</div>
```

Como se aprecia, todo el código HTML de la tarjeta está contenido en un `<div class="card">`. La imagen en el interior de la tarjeta tendrá clase `card-img-top` para que se redimensione automáticamente, y el contenido de la tarjeta se encuentra en un `<div class="card-body">`, cada uno con las clases apropiadas para darle estilo al texto.

Por defecto, como casi todos los elementos HTML, la tarjeta tenderá a ocupar todo el ancho disponible. Para limitar esto, se puede establecer una anchura máxima por CSS o incluirla en una columna Bootstrap para que su ancho se adapte dinámicamente. El siguiente ejemplo incluye la tarjeta en una columna cuyo ancho es el 33% de la fila:

```
<div class="row">
  <div class="col-md-4">
    <!-- Aqui el codigo anterior de la tarjeta -->
  </div>
</div>
```

Así, resulta:



El estilo oscuro es, al igual que para los componentes anteriores, aportado mediante las clases `bg-dark text-light`. Si se quisiera usar un color específico diferente, se puede usar CSS personalizado en su lugar.

Los componentes pueden combinarse entre sí, por ejemplo, añadiendo botones, etiquetas o listas en el interior de una tarjeta. Las posibilidades de combinación sólo están limitadas por la imaginación y el buen gusto. El resto de opciones relevantes sobre las tarjetas Bootstrap están disponibles en la página correspondiente de la documentación.

Ejemplos de vistas de la aplicación

En las siguientes subsecciones se mostrarán sugerencias de implementación de algunas vistas del proyecto de curso. Conviene recordar que la implementación individual de cada vista es libre, y por tanto no tiene por qué seguir la implementación mostrada a continuación, siempre que en todo caso cumpla los requisitos del proyecto.

6.1. Listado de fotos más recientes

En todas las vistas, se debe partir de un elemento raíz que contendrá el resto del código HTML, ya que es necesario para que Bootstrap funcione correctamente. Comenzaremos, pues, por agregar una fila con una única columna con texto centrado, para mostrar el título de la vista:

```
<div class="container">
  <div class="row text-center">
    <div class="col-md">
      <h3>Recent pictures</h3>
    </div>
    <hr>
  </div>
</div>
```

Podremos comprobar que la combinación de una única columna y texto centrado hace que el título quede centrado en la ventana:

Recent pictures

A continuación, podemos añadir las fotos por filas. Es conveniente decidir de antemano cuantas fotos deseamos mostrar en cada fila, a fin de establecer la clase necesaria para que cada columna ocupe el porcentaje de ancho correspondiente (esto es especialmente útil en el caso de que una fila no quede completamente llena, ya que por defecto las columnas que contenga intentarán ocupar todo el ancho disponible).

En este caso mostraremos tres fotos por fila, por lo que cada `row` tendrá tres columnas de clase `col-md-4` ($12/3 = 4$)

```
<div class="row text-center">
  <div class="col-md-4">
    Photo 1 goes here
  </div>
  <div class="col-md-4">
    Photo 2 goes here
  </div>
  <div class="col-md-4">
    Photo 3 goes here
  </div>
</div>
```

Añadir esta fila al interior del `<div>`, tras el título, mostrará el contenido en tres columnas bajo el mismo:

Recent pictures

Photo 1 goes here

Photo 2 goes here

Photo 3 goes here

Una vez establecida la estructura, podemos trabajar en el código HTML que emplearemos para mostrar cada foto. Por ejemplo, para cada una de ellas podríamos mostrar su título, su descripción y su autor. Haciendo click en ella accederíamos a la vista de detalle de foto, donde se mostrarían el resto de atributos de la foto con más detalle.

Para mostrar cada una de las fotos de la galería podemos usar el componente tarjeta (ver Sección) de Bootstrap, usando los datos de las imágenes que introdujimos en la práctica anterior:

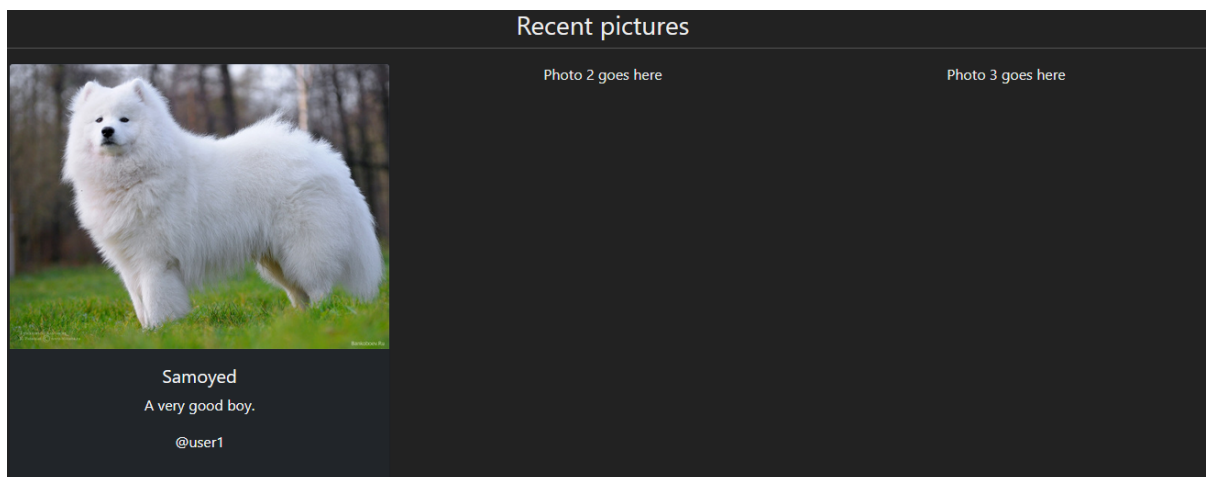
```
<div class="card bg-dark text-light">
  

  <div class="card-body">
    <h5 class="card-title text-center">Samoyed</h5>
    <p class="card-text">A very good boy.</p>
    <p class="text-end">@user1</p>
```

6. Ejemplos de vistas de la aplicación

```
</div>  
</div>
```

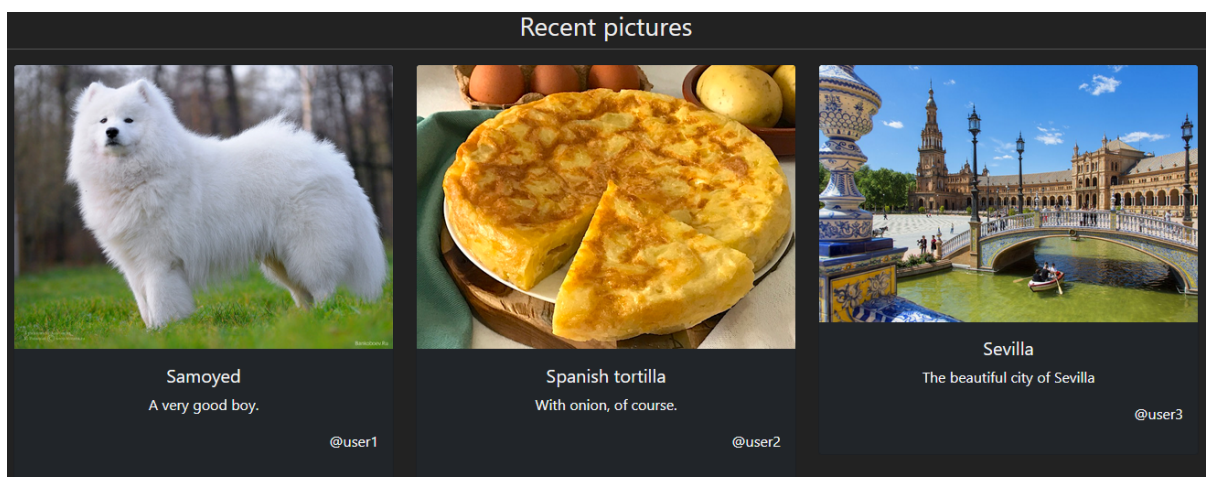
Si reemplazamos el texto que hemos puesto en las columnas como relleno con el código HTML anterior, podremos ver el resultado:



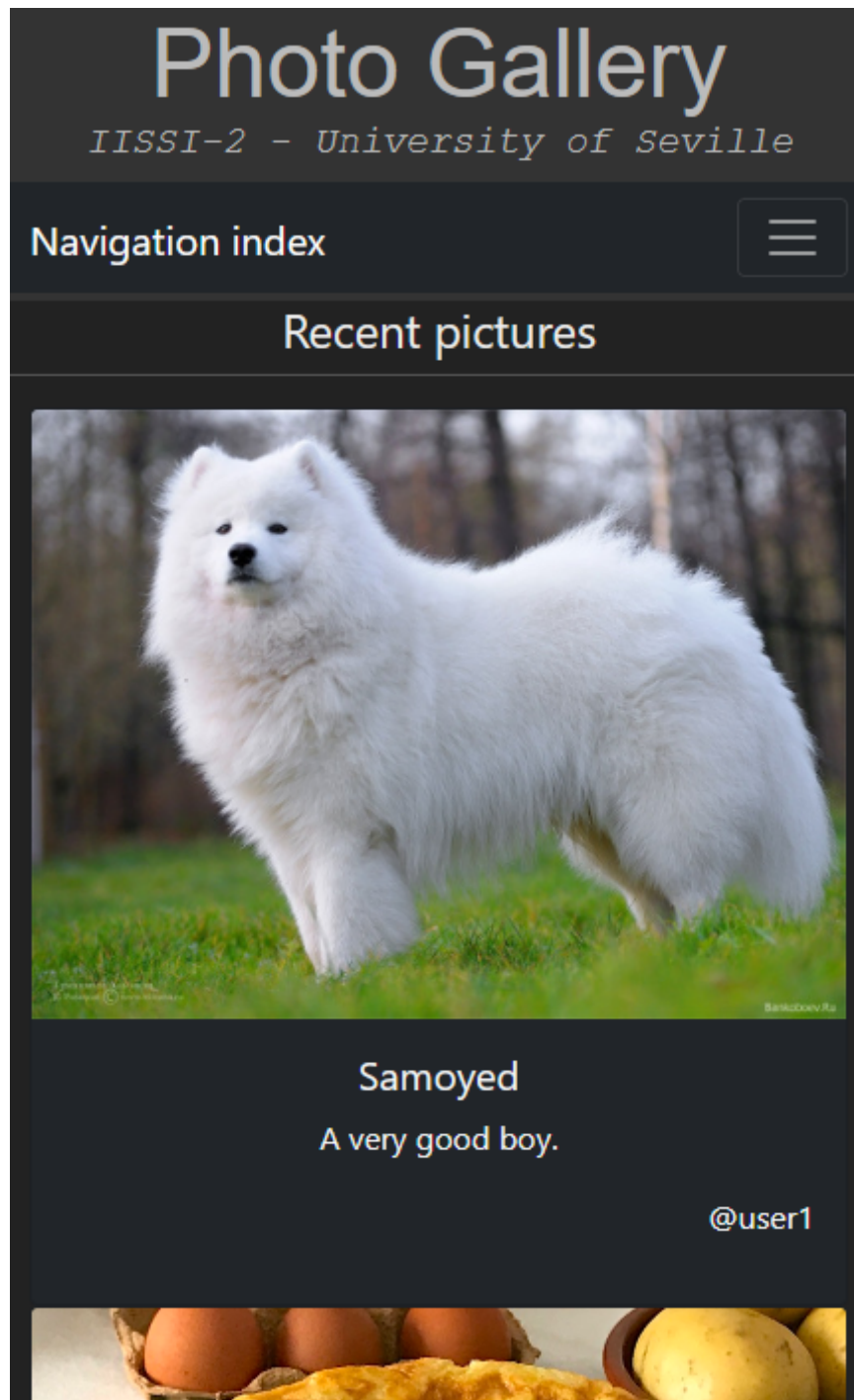
Para hacer que hacer click en la foto nos lleve a la página correspondiente de detalle de foto, podemos convertirla en un enlace envolviéndola en una etiqueta `<a>` :

```
<a href="photo_detail.html">  
    
</a>
```

En clases posteriores, se mostrará cómo hacer una única página de detalle de fotos que sea capaz de mostrar la información de cualquier foto de la galería. Finalmente, podemos repetir el proceso hasta completar la fila, modificando los datos necesarios en cada foto:



Por defecto, las tarjetas ocupan el ancho de cada columna. Combinado con la capacidad que nos da Bootstrap de organizar las columnas verticalmente cuando la ventana del navegador se estrecha, habremos conseguido que la galería se adapte automáticamente a móviles, mostrando en ese caso las fotos una encima de otra:



6.2. Detalle de foto

Una opción común es mostrar en la página principal de galería la información básica de cada foto, y luego tener una vista dedicada a mostrar todos los detalles de una foto concreta. La manera habitual de hacer esto es pasar el ID de la foto en cuestión a la vista de detalle usando un parámetro URL, y mediante JavaScript capturar este parámetro y cargar los datos de la foto en cuestión. Sin embargo, dado que JavaScript se introducirá en los próximos laboratorios, nos limitaremos a crear una vista con datos HTML estáticos. En el futuro, será fácil modificarla

6. Ejemplos de vistas de la aplicación

para que cargue de manera dinámica los datos de cualquier foto aprovechando la estructura de la vista ya creada.

Trabajaremos ahora sobre un archivo HTML llamado `photo_detail.html`, que será al que accedamos al pulsar sobre una imagen en la página principal de la galería, implementada en la sección anterior. Para ello, usaremos el mismo esquema de archivo HTML mostrado hasta ahora. Recuerde importar todas las dependencias necesarias en el `<head>`, e incluir la cabecera común con la barra de navegación.

Los elementos que mostraremos en esta vista serán el título y la descripción de la fotografía, la propia imagen a un tamaño más grande, la puntuación actual de la imagen, el usuario que subió la imagen y la fecha en que lo hizo. Además, añadiremos botones para editar y borrar la foto, y un pequeño formulario para valorarla.

En este caso separaremos la vista en dos columnas: en una columna más ancha mostraremos la información de la foto, mientras que en otra más estrecha a su derecha colocaremos los botones y demás elementos para interactuar con ella:

```
<div class="row text-center">
  <div class="col-md">
    Photo details here
  </div>

  <div class="col-md-3">
    Buttons here
  </div>
</div>
```

{: .warning }

i Info

Como en todas las vistas que usen Bootstrap, recuerde que todo el contenido debe definirse dentro de un

En la primera columna, incluiremos los datos relevantes que queramos mostrar de la foto:

```
<div class="col-md">
  <h3>Samoyed</h3>
  <h6>A very good boy</h6>
  <p>Uploaded by <a href="user_profile.html">User 1</a> on 12/01/2021</p>

  <hr>

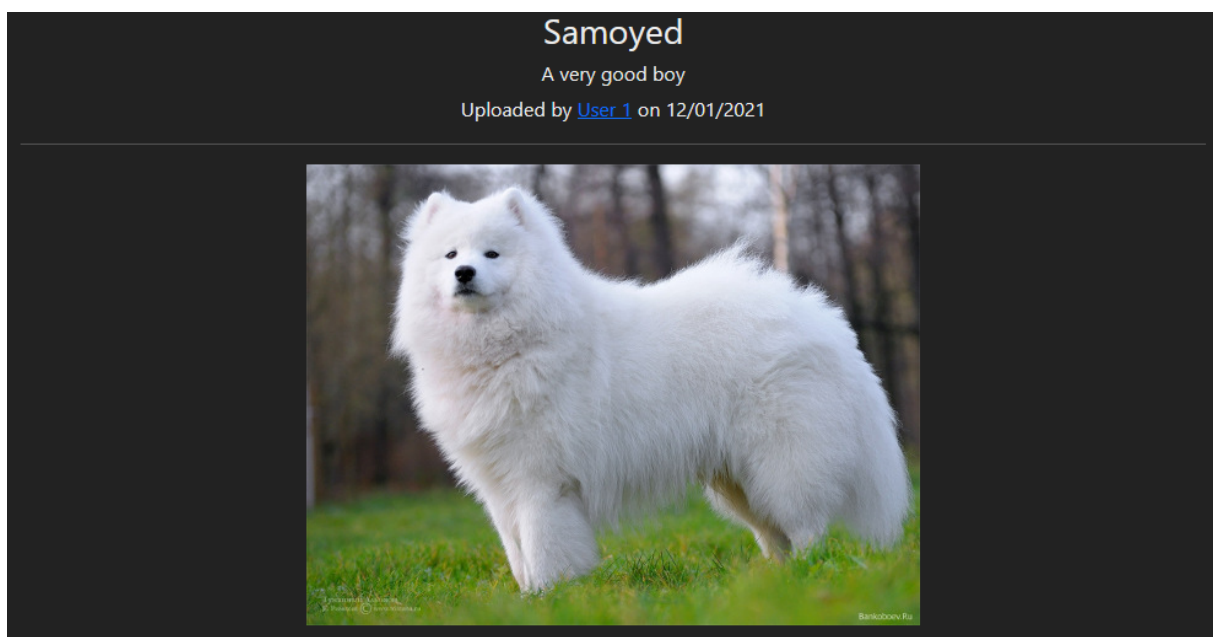
  
</div>
```

```
{: .warning }
```

i Info

Recuerde que no debe incluirse contenido directamente en el `<div>` de la fila, en su lugar, debe introducirse al menos una columna y poner el contenido dentro de ella para su correcto posicionamiento.

Nótese cómo se ha añadido a la imagen la clase `img-fluid`. Esto hace que Bootstrap le aplique automáticamente los estilos necesarios para que escale automáticamente con el ancho y alto de la página, haciéndola responsiva. El resultado es el siguiente:



En la columna de la derecha, añadiremos dos botones para eliminar y editar la foto. En prácticas posteriores daremos funcionalidad a estos botones mediante JavaScript:

```
<div>
  <button class="btn btn-primary">Edit this photo</button>
  <button class="btn btn-danger">Delete this photo</button>
</div>
```

Además, añadiremos un formulario con un `<select>` para que el usuario pueda valorar la foto con entre 1 y 5 estrellas. De nuevo, en esta práctica nos centramos únicamente en proporcionar la interfaz visual, no la implementación de funcionalidad:

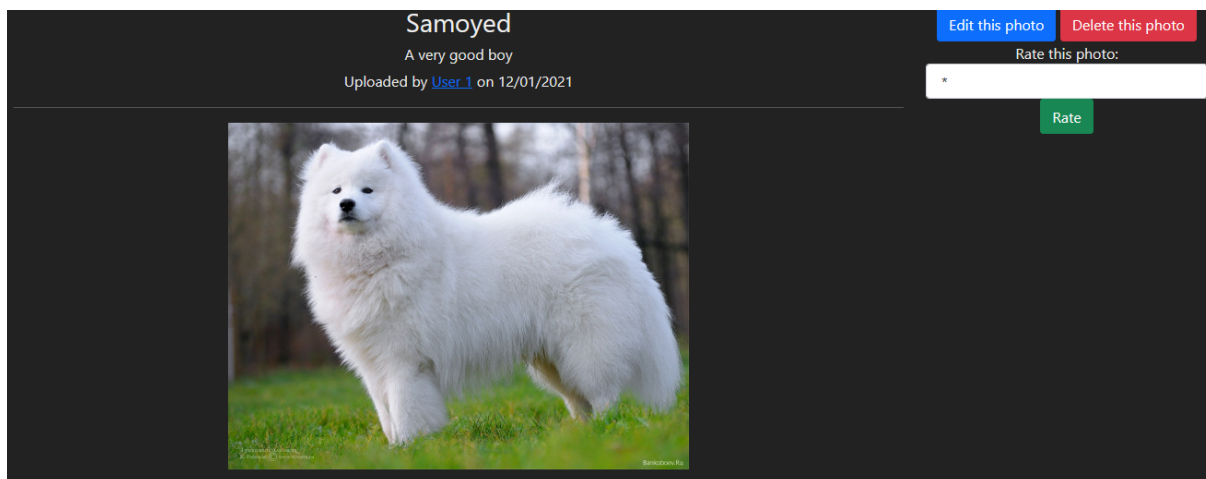
```
<form>
  <div class="form-group">
    <label for="rating-input">Rate this photo:</label>
    <select id="rating-input" name="rating" class="form-control">
      <option value="1">*</option>
```

```

<option value="2">**</option>
<option value="3">***</option>
<option value="4">****</option>
<option value="5">*****</option>
</select>
<button type="submit" class="btn btn-success">Rate</button>
</div>
</form>

```

El resultado es el siguiente:



Por último, podemos opcionalmente añadir un enlace con estilo de botón en una fila nueva al final de la vista, para facilitar la navegación y poder volver a la página principal:

```

<div class="row text-center">
  <div class="col-md">
    <a href="index.html" class="btn btn-primary">Return to the gallery</a>
  </div>
</div>

```

6.3. Registro de usuario

En la vista de registro de usuario podemos incluir un formulario Bootstrap para seguir unificando el estilo de la aplicación. Además, podemos aprovechar el sistema de filas y columnas para, por ejemplo, dividir el formulario en dos columnas diferentes y así aprovechar mejor el espacio en pantallas grandes. Para ello, comenzaremos por crear una etiqueta `<form>` y, dentro de ella, crear una fila con dos columnas:

```

<form>
  <div class="row">
    <div class="col-md">
      Element 1
    </div>
  </div>

```

```

</div>

<div class="col-md">
  Element 2
</div>
</div>
</form>

```

Register a new user

Element 1

Element 2

Podemos entonces introducir en cada columna un elemento input, tal y como se explicó en la Sección :

```

<div class="row">
  <div class="col-md">
    <div class="form-group">
      <label for="firstname-input">First name:</label>
      <input type="text" class="form-control text-light bg-dark"
id="firstname-input" name="firstName" placeholder="First name">
    </div>
  </div>

  <div class="col-md">
    <div class="form-group">
      <label for="lastname-input">Last name:</label>
      <input type="text" class="form-control text-light bg-dark"
id="lastname-input" name="lastName" placeholder="Last name">
    </div>
  </div>
</div>

```

Register a new user

First name:

Last name:

First name

Last name

Podemos repetir tantas filas como sean necesarias para implementar todos los campos del formulario. Además, podemos crear filas con un número diferente de columnas, y Bootstrap las adapta automáticamente:

Register a new user

First name:

Last name:

Email:

Phone number:

Username:

Password:

Repeat password:

Por supuesto, Bootstrap se encarga de que la adaptación a dispositivos móviles sea inmediata y no requiera mayor esfuerzo por nuestra parte:

Register a new user

First name:

Last name:

Email:

Phone number:

Username:

Finalmente, añadiremos una fila con una única columna, que contendrá el botón de enviar centrado horizontalmente:

```
<div class="row">
  <div class="col-md text-center">
    <button type="submit" class="btn btn-primary">Register</button>
  </div>
</div>
```

Register a new user

First name:	Last name:	
First name	Last name	
Email:	Phone number:	
Email	Phone number	
Username:	Password:	Repeat password:
Username	Password	Repeat password

Register

Actualización en GitHub

Actualice su proyecto en GitHub con los cambios hechos durante esta sesión. Recuerde los comandos relevantes:

- `git add .` añade los cambios efectuados en todos los archivos al próximo `commit` a efectuar.
- `git commit -m "mensaje"` crea un nuevo `commit` con los cambios efectuados a los archivos añadidos con el comando anterior, y con el mensaje indicado.
- `git push` actualiza el repositorio remoto con los cambios efectuados.

Anexo I: Iconos Font Awesome

Font Awesome, una librería diferente a Bootstrap, permite incorporar una gran cantidad de iconos a nuestra aplicación web de manera sencilla. **Pese a que la versión actual es la 5, se recomienda usar la 4** ya que tiene una mayor facilidad de instalación y uso. Font Awesome 4 se incluye en la plantilla de proyecto Silence, y si se han seguido los pasos mostrados en la Sección , se encontrará enlazado y listo para su uso. La línea relevante es la siguiente:

```
<link rel="stylesheet" href="/css/font-awesome.min.css">
```

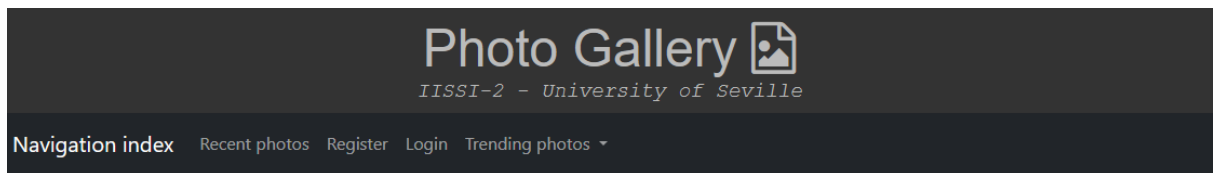
La lista de iconos disponibles está disponible en: (hay que cerrar el mensaje emergente sobre la versión 5 de Font Awesome).

Al hacer click en uno de los iconos mostrados, se muestra el código HTML relevante para incluirlo. En general, es de la forma

```
<i class="fa fa-*" aria-hidden="true"></i>
```

Donde * es el código del icono a incluir. Por ejemplo, si incluimos un icono en el título de la aplicación presente en la cabecera:

```
<h1 id="title">Photo Gallery <i class="fa fa-file-image-o" aria-hidden="true"></i></h1>
```



Los iconos Font Awesome son vectoriales, por lo que mantienen su calidad al hacer zoom y se pueden mostrar a cualquier tamaño.

i Info

Versión PDF disponible