

# Diseño responsive

Diseño de Sistemas Software  
Curso 2020/2021

---

Carlos Pérez Sancho



Universitat d'Alacant  
Universidad de Alicante

Departament de Llenguatges i Sistemes Informàtics  
Departamento de Lenguajes y Sistemas Informáticos

1. Introducción
2. Diseño responsive
3. Bootstrap 4
4. Integración Bootstrap/Laravel

# Introducción

---

Cuando se diseña un interfaz hay que tener en cuenta

- El tipo de público al que va dirigido
- Los dispositivos en los que se va a visualizar la aplicación

El público objetivo determina el estilo visual y el vocabulario empleado en el interfaz

Los dispositivos objetivo condicionan la forma en que se muestra la información en pantalla

Los usuarios están acostumbrados a un aspecto visual y un comportamiento característicos (**patrones**) de cada tipo de aplicación

**Familias de aplicaciones:** aplicaciones de gestión, ofimática, redes sociales, ...

El aspecto de la aplicación debe ser consistente en todos los dispositivos, pero al mismo tiempo debe adaptarse a las características de cada uno

- Tamaño de pantalla
- Introducción de datos

Actualmente lo más común es crear aplicaciones multiplataforma

- Herramientas para generar aplicaciones nativas en distintos dispositivos: Xamarin, Universal Windows Platform (UWP)
- Aplicaciones web que se ejecutan en navegadores

Las tecnologías web son omnipresentes

- Navegadores
- Webviews en móviles (Apache Cordova, PhoneGap)
- Aplicaciones de escritorio (Electron)
- Aplicaciones Web Progresivas (PWA)

Independientemente de la tecnología de desarrollo (UWP, web, aplicaciones móviles nativas) no podemos saber en qué tipo de pantalla se va a mostrar la aplicación

Existe la necesidad de crear diseños “fluidos” que se adapten automáticamente al tamaño de la pantalla: Diseño Adaptable o **Responsive Design**

**Convergencia en la forma de diseñar interfaces**

# Diseño web adaptable

## ***Adaptive web design***

Técnicas para adaptar el contenido de una página web a algún dispositivo

## ***Responsive Web Design (RWD)***

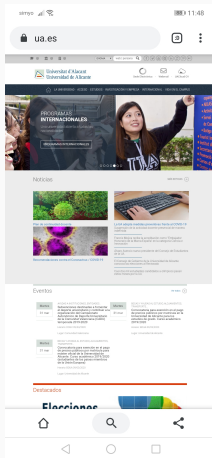
Técnicas para que el contenido de una página web se adapte **automáticamente** al tamaño de pantalla del dispositivo

En español normalmente se usa “Adaptable” como sinónimo de “Responsive”.

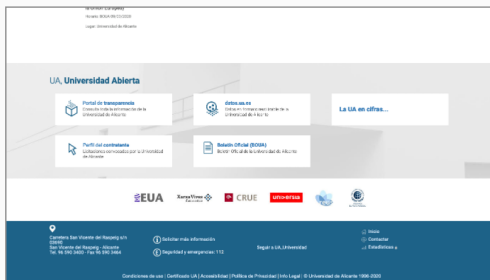


# ¿Por qué es necesario?

Si no hacemos nada, una página web se ve en un móvil igual que en un navegador de escritorio



# ¿Por qué es necesario?



Mejor experiencia de usuario (UX) → disminuye la tasa de rebote

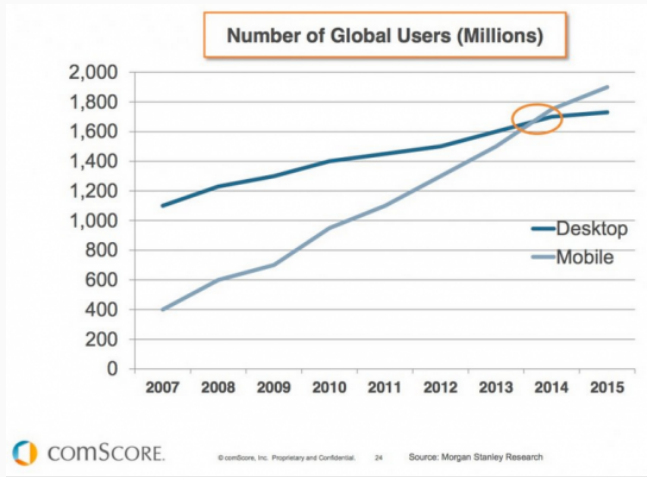
## ¿Por qué es necesario?

“Por primera vez en España hay más usuarios de internet (76,2 %) que de ordenador (73,3 %). El 77,1 % de los internautas accedieron a internet mediante el teléfono móvil.”

**Instituto Nacional de Estadística**, 2 de octubre de 2014

<http://www.ine.es/prensa/np864.pdf>

# ¿Por qué es necesario?



<http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics>

# ¿Por qué es necesario?

## Finding more mobile-friendly search results

Thursday, February 26, 2015

Webmaster level: all

When it comes to search on mobile devices, users should get the most relevant and timely results, no matter if the information lives on mobile-friendly web pages or apps. As more people use mobile devices to access the internet, our [algorithms](#) have to adapt to these usage patterns. In the past, we've made updates to ensure a site is [configured properly](#) and [viewable on modern devices](#). We've made it easier for users to [find mobile-friendly web pages](#) and we've introduced [App Indexing](#) to surface useful content from apps. Today, we're announcing two important changes to help users discover more mobile-friendly content:

### 1. More mobile-friendly websites in search results

Starting April 21, we will be expanding our use of mobile-friendliness as a ranking signal. This change will affect mobile searches in all languages worldwide and will have a significant impact in our search results. Consequently, users will find it easier to get relevant, high quality search results that are optimized for their devices.

<https://webmasters.googleblog.com/2015/02/finding-more-mobile-friendly-search.html>

# ¿Qué vamos a aprender?

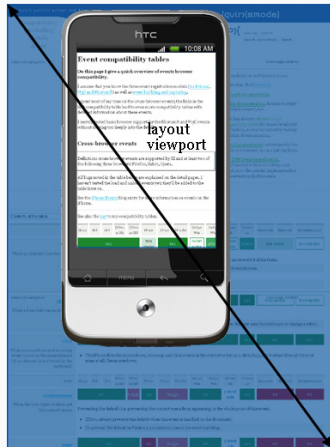
- Fundamentos del diseño responsive
- Funcionamiento básico de Bootstrap
- Creación de una plantilla y páginas usando Bootstrap en aplicaciones Laravel

## **Diseño responsive**

---

# Viewport

- Muchas páginas web están diseñadas para escritorio, con un tamaño fijo en píxeles
- Para poder mostrarlas correctamente, los navegadores móviles dibujan la página en un área (**viewport**) de tamaño similar al de una pantalla de escritorio: unos 980px, aunque puede variar

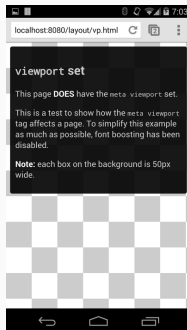


[http://www.quirksmode.org/blog/archives/2010/04/a\\_pixel\\_is\\_not.html](http://www.quirksmode.org/blog/archives/2010/04/a_pixel_is_not.html)



El tamaño del viewport se puede controlar con la etiqueta <meta>

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```



<https://developers.google.com/web/fundamentals/design-and-ux/responsive/>

# Densidades de pantalla

No todos los dispositivos tienen la misma resolución con el mismo tamaño de pantalla.

## Densidad de pantalla

Cantidad de píxeles que caben en la pantalla

Normalmente se mide en puntos por pulgada (dots per inch, **dpi**):

- ldpi ( $\approx 120\text{dpi}$ )
- mdpi ( $\approx 160\text{dpi}$ )
- hdpi ( $\approx 240\text{dpi}$ )
- xhdpi ( $\approx 320\text{dpi}$ ), xxhdpi ( $\approx 480\text{dpi}$ ), xxxhdpi ( $\approx 640\text{dpi}$ ), ...

# Un píxel no es un píxel

Usando píxeles reales, una página con diseño fijo se vería muy pequeña en dispositivos con densidades altas de pantalla.

Para que las páginas sean legibles, los navegadores móviles aumentan el tamaño de los píxeles CSS.



Viewport Sizes

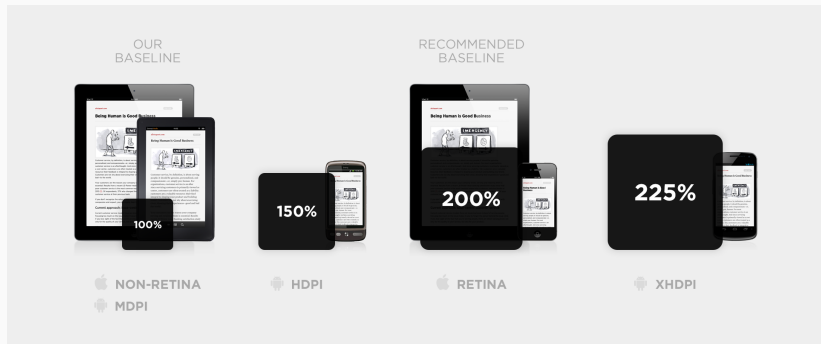
viewportsizes.com

galaxy s3

Device	Platform	Version	Portrait	Landscape
Samsung Galaxy S3 I9300	Android	4.0.4	360	640

# Un píxel no es un píxel

Se usa como referencia el tamaño real de un píxel en dispositivos con densidad mdpi (160dpi)



<http://web.archive.org/web/20161020193410/http://www.teehanlax.com/blog/density-converter/>

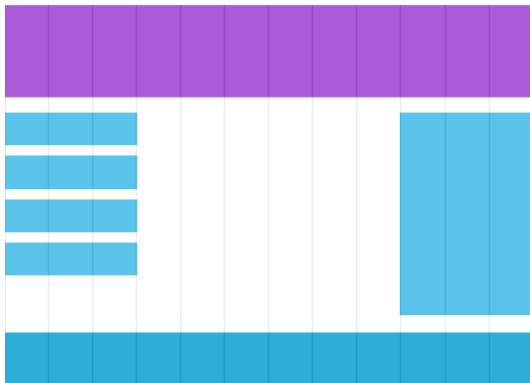
Muchas páginas estructuran el contenido mediante una rejilla (*grid*), que divide la página en columnas.

El uso de rejillas facilita el posicionamiento de los elementos en la pantalla colocando los elementos en celdas.

10%	80%	10%
-----	-----	-----

# Sistemas de rejillas

Normalmente se definen rejillas de 12 columnas, que se redimensionan automáticamente al cambiar el tamaño de la pantalla.



[http://www.w3schools.com/css/tryresponsive\\_grid.htm](http://www.w3schools.com/css/tryresponsive_grid.htm)

Una “rejilla fluida” reestructura el contenido apilando las columnas cuando el tamaño de la pantalla es demasiado pequeño.

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8									.col-md-4					
.col-md-4				.col-md-4				.col-md-4						
.col-md-6						.col-md-6								

Vista en pantallas grandes

.col-md-1
.col-md-1
.col-md-1
.col-md-8
.col-md-4
.col-md-4
.col-md-4
.col-md-4
.col-md-6
.col-md-6

Vista en pantallas pequeñas

<https://getbootstrap.com/docs/4.6/examples/grid/>

Para decidir cuándo debe cambiarse la disposición del contenido, las rejillas fluidas definen una serie de puntos de interrupción (**breakpoints**)

Se definen breakpoints para varios tamaños de pantalla

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
<code>.container</code>	100%	540px	720px	960px	1140px
<code>.container-sm</code>	100%	540px	720px	960px	1140px
<code>.container-md</code>	100%	100%	720px	960px	1140px
<code>.container-lg</code>	100%	100%	100%	960px	1140px
<code>.container-xl</code>	100%	100%	100%	100%	1140px
<code>.container-fluid</code>	100%	100%	100%	100%	100%

Breakpoints en Bootstrap 4



# Media queries

**Media queries:** introducidas en CSS3, permiten aplicar un conjunto de reglas sólo si se cumplen unas determinadas condiciones.

Permiten identificar el tipo de dispositivo, el tamaño y la orientación de la pantalla.

```
@media only screen and (max-width: 500px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

[http://www.w3schools.com/css/tryit.asp?filename=tryresponsive\\_mediaquery](http://www.w3schools.com/css/tryit.asp?filename=tryresponsive_mediaquery)

# Imágenes responsive

- Técnica básica: **max-width**

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

- Distintas imágenes para distintas densidades de pantalla:

```

```

Fixed size, here or in CSS

Image url

Pixel density of screen

This is used as the 1x src & by browsers that don't support srcset

- Adaptar el nivel de detalle al tipo de pantalla



<https://jakearchibald.com/2015/anatomy-of-responsive-images/>

# Bootstrap 4

---

# Bootstrap 4

Bootstrap es un framework para el diseño de interfaces con HTML y CSS



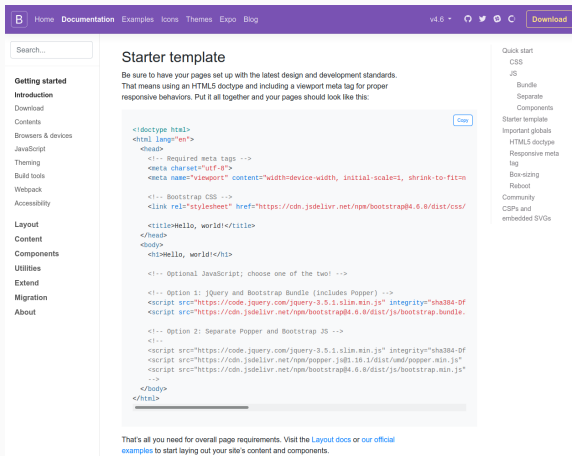
<https://getbootstrap.com/>

Bootstrap proporciona

- Una rejilla fluida para posicionar el contenido
- Herramientas para dar formato a distintos tipos de contenido: tablas, imágenes, código fuente, ...
- Una colección de componentes con estilos predefinidos: botones, menús, formularios, ...
- Utilidades para modificar el aspecto de los elementos anteriores: bordes, márgenes, alineación, ...

Todo esto mediante el uso de clases CSS para no tener que escribir nuestras propias reglas

## El primer paso es copiar la plantilla de la página de Bootstrap



The screenshot shows the Bootstrap 4.6 documentation website. The top navigation bar is purple with links for Home, Documentation, Examples, Icons, Themes, Expo, and Blog. A search bar is on the left, and a 'Download' button is on the right. The left sidebar contains a 'Getting started' section with links to Introduction, Download, Contents, Browsers & devices, JavaScript, Theming, Build tools, Webpack, Accessibility, Layout, Content, Components, Utilities, Extend, Migration, and About. The main content area is titled 'Starter template' and includes a paragraph explaining the purpose of the template. Below this is a code block containing the starter template HTML code. The right sidebar lists various resources like 'Quick start', 'CSS', 'JS', 'Bundle', 'Separate', 'Components', 'Starter template', 'Important globals', 'HTML5 doctype', 'Responsive meta tag', 'Box-sizing', 'Reboot', 'Community', 'CSPs and embedded SVGs'.

**Starter template**

Be sure to have your pages set up with the latest design and development standards. That means using an HTML5 doctype and including a viewport meta tag for proper responsive behaviors. Put it all together and your pages should look like this:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=1">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: jQuery and Bootstrap Bundle (includes Popper) -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-0f
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-0f
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.min.js"
    -->
  </body>
</html>
```

That's all you need for overall page requirements. Visit the [Layout docs](#) or [our official examples](#) to start laying out your site's content and components.

<https://getbootstrap.com/docs/4.6/getting-started/introduction/#starter-template>

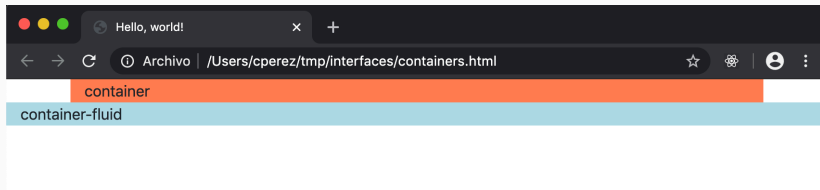
# Contenedores

Los contenedores son los elementos más básicos de Bootstrap

```
<body>
  <div class="container">
    <h1>Hello world!</h1>
  </div>
</body>
```

El contenedor determina el ancho máximo del contenido en la ventana

- **container** tiene el tamaño del breakpoint inferior más cercano
- **container-fluid** ocupa siempre todo el ancho disponible



## Las rejillas deben estar obligatoriamente dentro de un contenedor

El contenido se organiza en filas y columnas, usando etiquetas `div` y las clases proporcionadas por Bootstrap

Si no se especifica el tamaño de las columnas, se reparten el tamaño disponible a partes iguales

One of three columns

One of three columns

One of three columns

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

Copy



Hay 12 espacios disponibles en cada fila para repartir entre sus columnas

col-sm-8		col-sm-4
col-sm	col-sm	col-sm

```
<div class="container">
  <div class="row">
    <div class="col-sm-8">col-sm-8</div>
    <div class="col-sm-4">col-sm-4</div>
  </div>
  <div class="row">
    <div class="col-sm">col-sm</div>
    <div class="col-sm">col-sm</div>
    <div class="col-sm">col-sm</div>
  </div>
</div>
```

Copy

## Layout básico de una página

```
<div class="container">
  <div class="row">
    <div class="col-sm">Header</div>
  </div>
  <div class="row">
    <div class="col-sm-8">Body</div>
    <div class="col-sm-4">Sidebar</div>
  </div>
  <div class="row">
    <div class="col-sm">Footer</div>
  </div>
</div>
```

Se pueden anidar rejillas para crear estructuras más complejas

# Responsive grid

Las columnas pueden tener varias clases para definir su comportamiento en distintos tamaños de pantalla

	<b>Extra small</b> <576px	<b>Small</b> ≥576px	<b>Medium</b> ≥768px	<b>Large</b> ≥992px	<b>Extra large</b> ≥1200px
<b>Max container width</b>	None (auto)	540px	720px	960px	1140px
<b>Class prefix</b>	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>



## Para crear componentes podemos usar los ejemplos de Bootstrap como punto de partida

B

Home Documentation Examples Icons Themes Expo Blog

v4.4

Download

Search...

Getting started

Layout

Content

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Media object

Modal

Navs

### Examples

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.

PrimarySecondarySuccessDangerWarningInfoLightDarkLink

```
<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>

<button type="button" class="btn btn-link">Link</button>
```

#### Conveying meaning to assistive technologies

Using color to add meaning only provides a visual indication, which will not be conveyed to users of assistive technologies – such as screen readers. Ensure that information denoted by the color is either obvious from the content itself (e.g. the visible text), or is included through alternative means, such as additional text hidden with the `.sr-only` class.

Examples

Disable text wrapping

Button tags

Outline buttons

Sizes

Active state

Disabled state

Button plugin

Toggle states

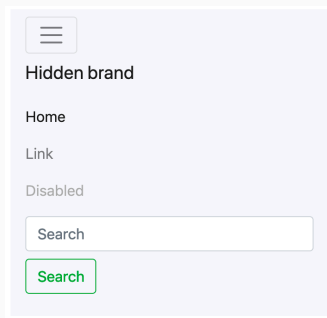
Checkbox and radio buttons

Methods

## Componentes más frecuentes

- Botones (buttons)
- Menús de navegación (NavBar)
- Formularios (Forms)
- Paginación (Pagination)

Permite crear menús que se contraen en dispositivos móviles



Los formularios se adaptan al tamaño de la pantalla y tienen un estilo más agradable

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

Submit



# Forms

Los campos de tipo input (donde el usuario escribe) se pueden complementar con 2 atributos de HTML5

- **placeholder:** texto que se muestra al usuario como pista
- **type:** text, password, email, number, ...

Especificar el tipo permite abrir un teclado especializado en dispositivos móviles

[https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)

```
<input type="email" placeholder="name@example.com" ...>
```

Email address

# **Integración Bootstrap/Laravel**

---

Repositorio GitHub

<https://github.com/cperezs/laravel-bootstrap>

routes/web.php

```
Route::get('/products', 'ProductController@list');  
Route::get('/products/{id}', 'ProductController@details');
```

- **/products** muestra todos los productos
- **/products/{id}** muestra los detalles de un producto, el valor del parámetro id se pasa al método del controlador

Las dos rutas enlazan con métodos del controlador

**ProductController**

app/Http/Controllers/ProductController.php

```
class ProductController extends Controller
{
    public function list() {
        $products = Product::all();
        return view('products.list')->with('products', $products);
    }

    public function details($id) {
        $product = Product::findOrFail($id);
        return view('products.details')->with('product', $product);
    }
}
```

Cada método realiza una consulta y pasa el resultado a la vista correspondiente

**Por simplificar, no se ha hecho uso de ningún patrón de lógica de negocio en el ejemplo**

## views/layout.blade.php

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.m

    <title>App name - @yield('title')</title>
  </head>
  <body>
    <div class="container">
      @yield('content')
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0ls
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.min.js" integrity=
  </body>
</html>
```

views/products/details.blade.php

```
@extends('layout')

@section('title', 'Product details')

@section('content')
<h1>{{ $product->name }}</h1>
<p><strong>Price:</strong> {{ $product->price }}</p>
<p>{{ $product->description }}</p>
<p><a href="{{ action('ProductController@list') }}">Go back</a></p>
@endsection
```

# Vista lista productos

views/products/list.blade.php

```
@extends('layout')

@section('title', 'Product list')

@section('content')
<h1>Products</h1>
<table class="table table-striped">
  <thead>
    <tr>
      <th scope="col">Id</th>
      <th scope="col">Name</th>
      <th scope="col">Price</th>
    </tr>
  </thead>
  <tbody>
    @foreach ($products as $product)
      <tr>
        <th scope="row">{{ $product->id }}</th>
        <td><a href="{{ action('ProductController@details', $product->id) }}">{{ $product->name }}</a></td>
        <td>{{ $product->price }}</td>
      </tr>
    @endforeach
  </tbody>
</table>
@endsection
```



# Vista lista productos

Crea una tabla usando los estilos de Bootstrap

En cada fila se crea un enlace para ver los detalles del producto

```
@foreach ($products as $product)
<tr>
  <th scope="row">{{ $product->id }}</th>
  <td><a href="{{ action('ProductController@details', $product->id) }}">{{ $product->name }}</a></td>
  <td>{{ $product->price }}</td>
</tr>
@endforeach
```

**¿Preguntas?**