

MODULO BOOTSTRAP

MOBILE FIRST

Este término hace referencia a la tendencia de desarrollar las aplicaciones web pensando primero que se utilizarán principalmente desde dispositivos móviles. Actualmente, es normal que los browsers de los dispositivos móviles estén optimizados y brinden un mejor soporte para la tecnología HTML5 y CSS3, por lo cual resultaría evidente pensar el sistema



web considerando que será principalmente utilizado por usuarios en entornos reducidos (el tamaño reducido de la pantalla del dispositivo móvil). Es decir, es una filosofía en la cual se piensa el diseño y construcción de la aplicación para que la experiencia sea óptima en los dispositivos móviles.

QUE ES BOOTSTRAP

Bootstrap es un potente conjunto de herramientas que ofrecen una gran cantidad de funciones para el front-end que permiten acelerar el desarrollo de prototipos y/o de la aplicación. Fue desarrollado inicialmente en 2011 por ingenieros de Twitter. La versión actual, Bootstrap 5, fue liberada en mayo de 2021.

Incluye plantillas HTML y CSS con tipografías, formas, botones, cuadros, tablas, barras de navegación, carruseles de imágenes y muchas otras. Para muchos de los efectos que se generan a través de estas herramientas se utiliza javascript. Por este motivo se debe considerar que la versión 5.x emplea vanilla JavaScript en lugar de jQuery (utilizado en versiones anteriores)

Aunque su preferencia es mobile first, permite crear diseños que se ven bien en múltiples dispositivos por lo que en realidad ofrece un responsive design (diseño responsivo)

Características Principales

Ventajas

- Orientado a programadores, no a diseñadores gráficos
- Es posiblemente la herramienta más popular para este fin, aunque hay alternativas como Foundation
- Resulta sencillo y rápido escribir páginas con muy buen aspecto
- Se adapta a distintos dispositivos (responsive design)
- Proporciona un diseño consistente
- Es compatible con los browsers modernos
- Es software libre

Desventajas

- Al ser una herramienta muy popular, las páginas web que no estén personalizadas quedan iguales que las de todo el mundo

- No es especialmente fácil personalizar los estilos (Foundation puede ser más adecuado para esto)

Hola Mundo con Bootstrap

Existen diferentes maneras de integrar Bootstrap a un proyecto de aplicación web, pero de forma genérica estos son los pasos por realizar:

- Definir el viewport
- Incluir un elemento link apuntando al CSS de Bootstrap
- Incluir un elemento script apuntando al código JavaScript de Bootstrap

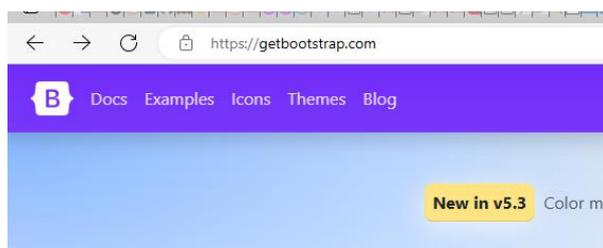
Un aspecto interesante es que la inclusión de enlaces a CSS y Javascript se puede realizar de diferentes maneras.

Uso de enlaces CDN

Una manera muy sencilla de utilizar Bootstrap es ofrecida en la propia página de la herramienta.

Paso 1: Ir a la página de Bootstrap: <https://getbootstrap.com/>

Paso 2: Seleccionar la opción Docs del menú superior de la página



Paso 3: Buscar la sección en donde se visualice CDN Links. Allí tendrá disponible los enlaces que deberá copiar para colocarlos en la página web.

CDN links

As reference, here are our primary CDN links.

Description	URL
CSS	https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css
JS	https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js

You can also use the CDN to fetch any of our [additional builds listed in the Contents page](#).

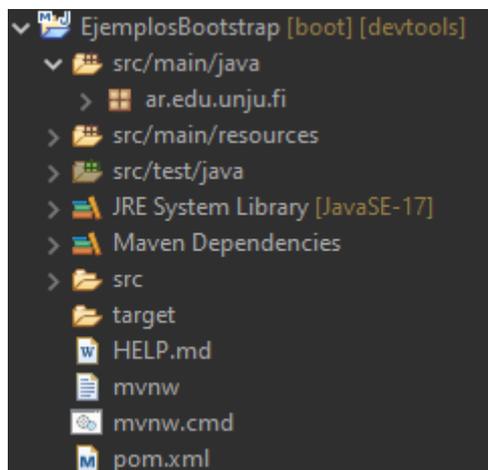
Una CDN es una red de servidores que distribuye contenido de un servidor de "origen" a todo el mundo, almacenando ese contenido en caché en un lugar más cercano al de cada usuario final que accede a Internet con un dispositivo habilitado para la Web. El contenido que solicitan se almacena primero en el servidor de origen y, a continuación, se replica y almacena en otros lugares según sea necesario. Al almacenar contenido físicamente cerca del lugar donde está el usuario y reducir la distancia que debe recorrer, se disminuye la latencia y la presión ejercida en los servidores de origen, al distribuir geográficamente la carga entre varios servidores.

Un CDN (Content Delivery Network) representa entonces, un grupo de servidores distribuidos en diferentes ubicaciones geográficas de todo el mundo para permitir la entrega rápida del contenido de un sitio web. También se conoce como red de distribución de contenidos.

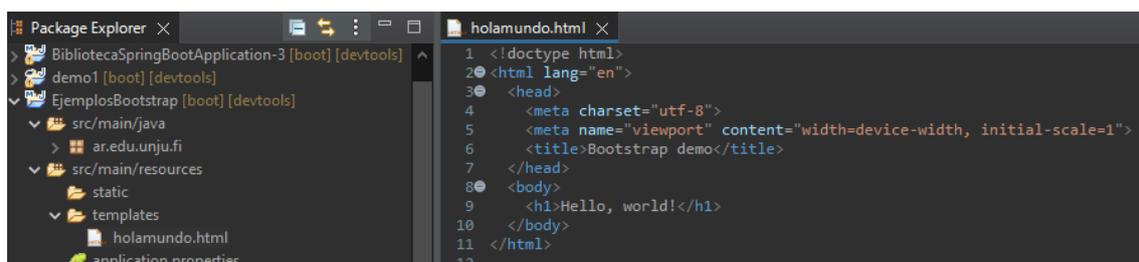
Cuando un usuario se conecta a un sitio web, la distancia física entre él y el servidor de alojamiento web afecta al tiempo que tarda el contenido en aparecer en la pantalla. Además de reducir el consumo de ancho de banda, la red de centros de datos distribuida globalmente mejora la velocidad de entrega de los contenidos para los usuarios, independientemente de su ubicación en relación con el servidor principal.

Ejemplo de aplicación en el proyecto Spring Boot

Observe la arquitectura de una aplicación con Spring Boot



Sobre este proyecto se creará una página web con el siguiente contenido



Si ejecutamos la página desde el browser local, obtendremos lo siguiente

Hello, world!

Puede observar que la línea 5 contiene una etiqueta meta con el atributo name="viewport".

Ya desde su diseño original, un requisito importante es que las páginas se puedan representar en pantallas de cualquier tamaño. Con la aparición de los smartphones, esto es aún más necesario y complejo de llevar a cabo. A lo largo de los años se han usado diferentes técnicas para conseguirlo, obteniendo cada vez resultados mejores:

1. Técnica inicial: Viewport. Barras de desplazamiento horizontal y vertical, recomposición de los elementos sobre el viewport.
2. Primeros smartphones: Viewport virtual
3. Teléfonos móviles actuales. Diseño responsivo basado en grid

Viewport

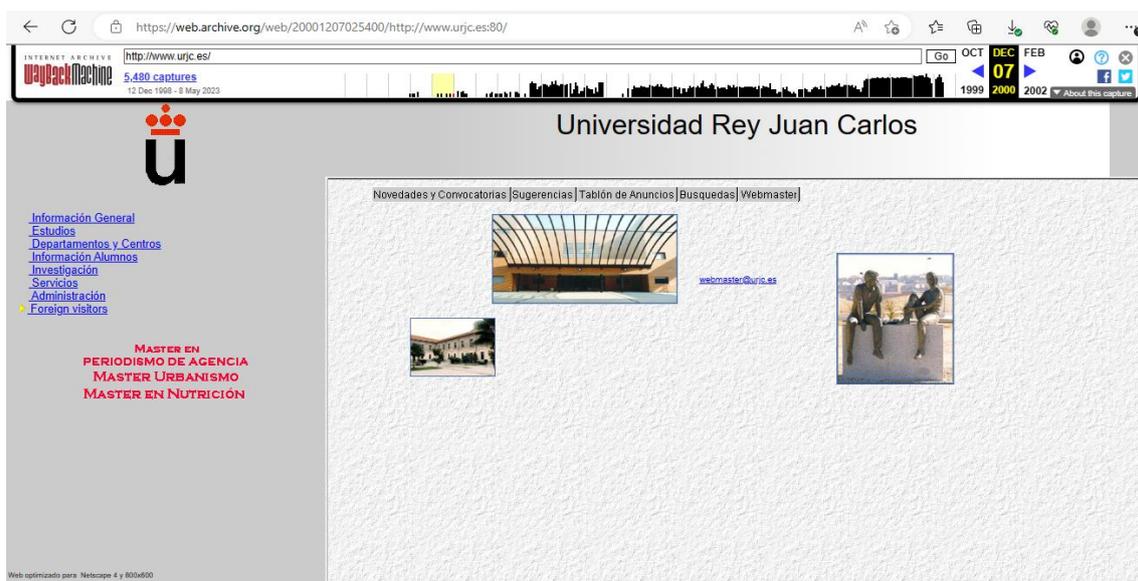
Para diseñar webs en dispositivos móviles, es importante tener claro qué es el viewport y cómo se comporta:

- Viewport es la zona visible de una página web. En los browsers tradicionales de escritorio, coincide con la ventana del browser.
- Supongamos una página web grande y compleja, como la portada de un periódico. Es muy probable que la página sea más grande que la ventana del browser, el usuario usará las barras de scroll para mover el viewport sobre el documento. Al redimensionar la ventana, cambiará el tamaño del viewport
- Cambiar el tamaño del viewport reposiciona el texto y todos los elementos: las líneas se truncan, las imágenes se recolocan, etc.

Viewport es un rectángulo donde se compone un fragmento (tal vez completo) de la página web para presentarla al usuario

Con la aparición de los browsers en teléfonos móviles, los cambios del tamaño de la pantalla son mucho más drásticos. Las técnicas tradicionales siguen funcionando, pero proporcionan una experiencia de uso muy poco satisfactoria. El área visible de un móvil es demasiado pequeña, componer una página web tradicional en ese viewport queda mal. Observe lo que sucede en esta página antigua cuando la ventana es muy grande o pequeña:

<https://tinyurl.com/y7e77lvw>



La imagen anterior representa la visualización de la página de manera estándar. Ahora observe la misma página al modificar el tamaño de la ventana (Fig. 1); los efectos visuales son muy malos, aparecen las barras de scroll, y aún así se ve mucho menos estética.

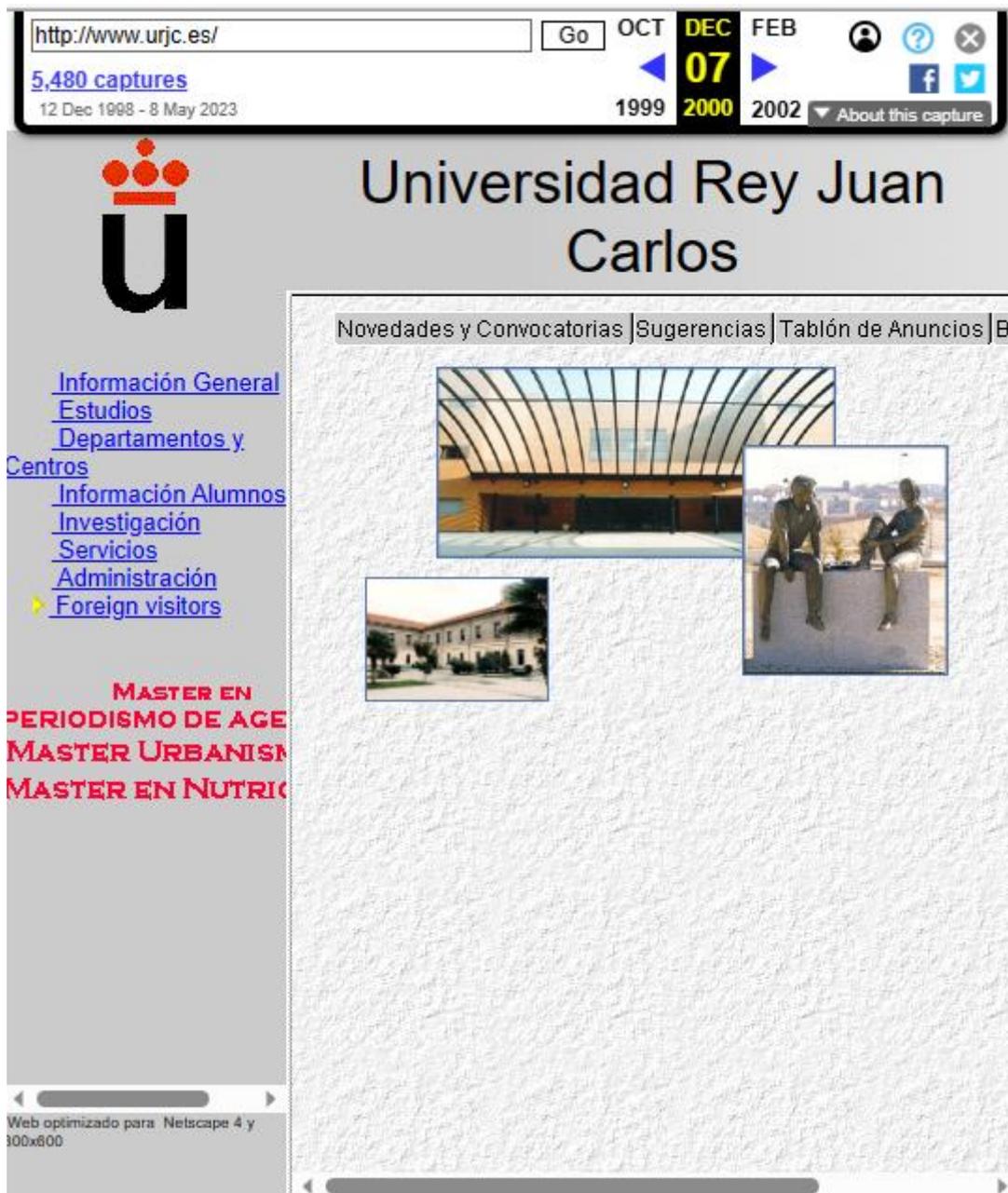


Figura 1. Efectos del uso de viewport

Además, en un navegador para móvil no hay barras de scroll, ocuparían un espacio demasiado valioso; ni ventanas, serían demasiado pequeñas.

Viewport virtual

Para solucionar los problemas del viewport, aparece el viewport virtual, mayor que el viewport ordinario (la pantalla). Lo introduce Apple para Safari en iOS, luego pasa a ser estándar:

- El ancho del viewport virtual es razonablemente grande, por ejemplo 980 píxeles en el navegador Safari para iPhone.
- El navegador compone la página sobre este viewport virtual, ya no hacen falta barras de desplazamiento horizontal.

- El usuario arrastra el viewport (la pantalla, más pequeña) sobre el viewport virtual, para que le muestre una zona u otra del documento. También se le puede permitir hacer zoom. Redimensionar este viewport ya no provoca la recomposición de la página.

Páginas responsivas

Una página web moderna con un mínimo de calidad se entiende que tiene que ser responsive.

- La página se adapta al tamaño de la pantalla (escritorio, tablet, móvil), sin usar la barra de desplazamiento horizontal, que es muy incómoda.
- La barra de desplazamiento vertical se sigue usando, no es molesta
- El diseño responsive tal y como lo conocemos en la actualidad se basa en el uso de un grid. En español se traduce por cuadrícula, rejilla o casilla. Sobre este grid trabaja Bootstrap, hay otros, pero siempre son similares.
- En estas páginas ya no hace falta un viewport virtual, porque la página está diseñada para adaptarse al viewport ordinario (la pantalla pequeña)

Sobre este aspecto profundizaremos más adelante.

Conclusión Parcial:

La propiedad <meta name="viewport"...> permite indicar la escala inicial del viewport.

Como las páginas con Bootstrap son responsive, especificaremos que el viewport virtual coincida con el ancho de la pantalla, esto es con el viewport ordinario. En otras palabras: no se trabajará con el viewport virtual.

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

También se puede inhabilitar el zoom en dispositivos móviles con user-scalable=no. Además los usuarios solo podrán hacer scroll y tendrá una apariencia nativa.

```
<meta name="viewport" content="width=device-width, initial-scale=1,maximum-scale=1, user-scalable=no">
```

Ahora ingresamos los enlaces CDN, de la siguiente manera:

Coloque la etiqueta <link> en <head> para referenciar CSS y la etiqueta <script> para referenciar el paquete de JavaScript (incluido Popper para colocar menús desplegables, ventanas emergentes e información sobre herramientas) antes del cierre </body>.

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1,maximum-scale=1, user-scalable=no">
6 <title>Bootstrap demo</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
8 rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJ0Z"
9 crossorigin="anonymous">
10 </head>
11 <body>
12 <h1>Hello, world!</h1>
13 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"
14 integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1Hicje39Wm4jDKdf19U8gI4ddQ3GYNs7NTKfAdVQSZ"
15 crossorigin="anonymous"></script>
16 </body>
17 </html>
```

Observe que con esto la apariencia del formato del texto cambia, lo cual muestra que está todo listo para empezar a trabajar.

Hello, world!

Uso de Maven

Podemos agregar a nuestro proyecto la dependencia de Bootstrap. Es más podemos hacer que los webjars se encarguen de buscar las dependencias correspondientes a CSS y JS de Bootstrap.

La mayoría de los desarrolladores de Java prefieren usar Maven y Gradle como herramientas de gestión de compilación y dependencia.

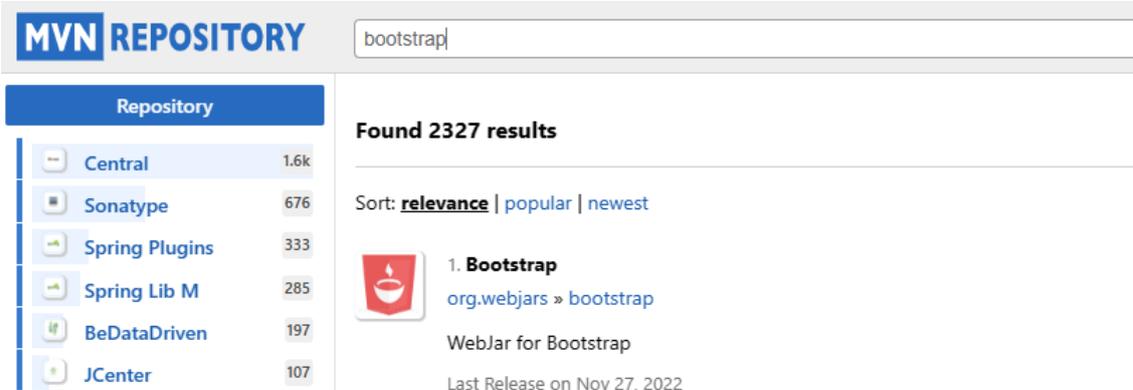
El problema principal que resuelve WebJars es hacer que las dependencias del lado del cliente estén disponibles en Maven Central y se puedan usar en cualquier proyecto estándar de Maven.

Aquí se mencionan algunas ventajas interesantes de WebJars:

- Podemos administrar explícita y fácilmente las dependencias del lado del cliente en aplicaciones web basadas en JVM.
- Podemos usarlos con cualquier herramienta de compilación de uso común, por ejemplo: Maven, Gradle, etc.
- Los WebJars se comportan como cualquier otra dependencia de Maven, lo que significa que también obtenemos dependencias transitivas.

Paso 1: Ir a la página de Maven: <https://mvnrepository.com>

Paso 2: Buscar Bootstrap y elegir la opción que devuelva org.webjars



The screenshot shows the Maven Repository search interface. The search bar contains 'bootstrap'. The results section shows 'Found 2327 results' and is sorted by 'relevance'. The first result is '1. Bootstrap' from 'org.webjars » bootstrap', which is a 'WebJar for Bootstrap' with a 'Last Release on Nov 27, 2022'. On the left, a sidebar lists various repositories: Central (1.6k), Sonatype (676), Spring Plugins (333), Spring Lib M (285), BeDataDriven (197), and JCenter (107).

Paso 3: Copie la dependencia más actual. Tenga en cuenta que al momento de realizar este tutorial la versión 5.3 aún está en etapa Alpha, por lo que se descargará la versión 5.2.x más estable.

 **Bootstrap » 5.2.3**
WebJar for Bootstrap

License	Apache 2.0
Categories	Web Assets
Tags	web bootstrap assets
HomePage	http://webjars.org
Date	Nov 27, 2022
Files	pom (8 KB) jar (2.3 MB) View All
Repositories	Central
Ranking	#1124 in MvnRepository (See Top Artifacts) #9 in Web Assets
Used By	398 artifacts

[Maven](#)
[Gradle](#)
[Gradle \(Short\)](#)
[Gradle \(Kotlin\)](#)
[SBT](#)
[Ivy](#)
[Grape](#)
[Leiningen](#)
[Buildr](#)

```

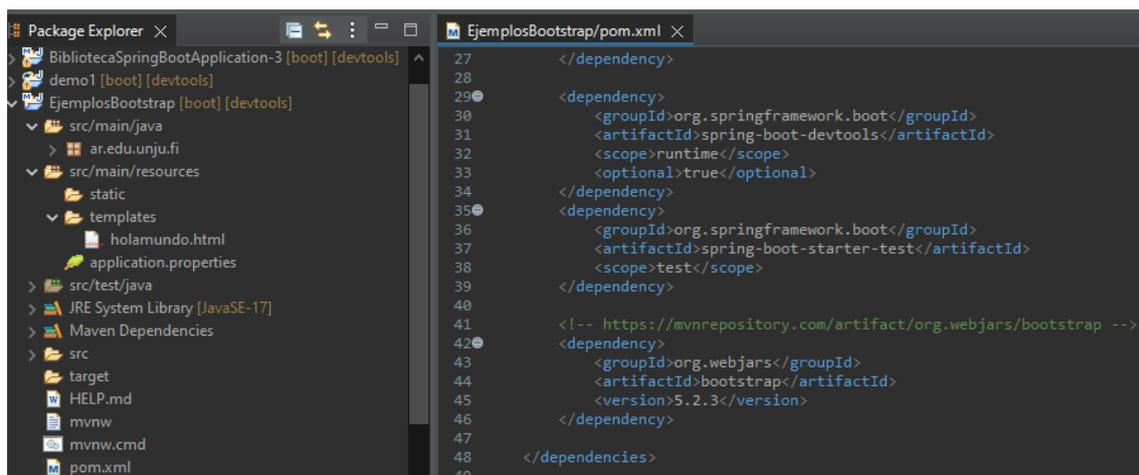
<!-- https://mvnrepository.com/artifact/org.webjars/bootstrap -->
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>bootstrap</artifactId>
  <version>5.2.3</version>
</dependency>

```

Include comment with link to declaration

Copied to clipboard!

Paso 4: Agregar la dependencia al archivo pom.xml



Si observamos las dependencias de Maven podremos ubicar que se hallan las bibliotecas CSS y JS de Bootstrap.

Además, vamos a incluir webjars locator. Esta librería nos va a ser útil a la hora de indicar la versión de Bootstrap que vamos a referenciar. Esta biblioteca se encargará de detectar la versión

que correspondería referenciar, lo cual ayudará a no tener que actualizar esto en los enlaces de Bootstrap en las páginas.



License	MIT
Categories	Web Assets
Tags	web locator assets
Ranking	#2498 in MvnRepository (See Top Artifacts) #29 in Web Assets
Used By	164 artifacts

Central (48)

Version	Vulnerabilities	Repository	Usages	Date
0.46.x 0.46		Central	16	Oct 31, 2022

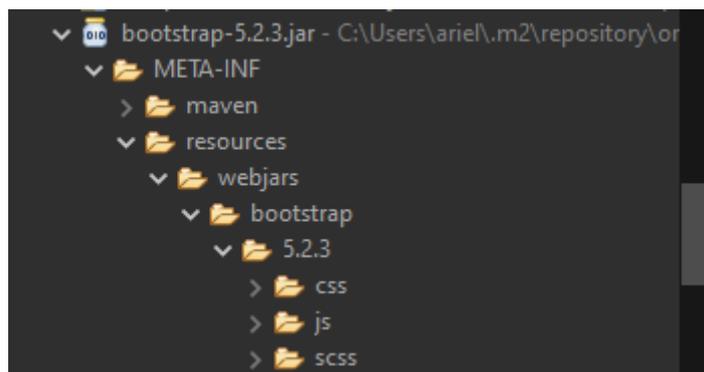
Buscaremos la versión más nueva y copiaremos su descripción de dependencia, para luego colocarla en el archivo pom.xml

```
<!-- https://mvnrepository.com/artifact/org.webjars/webjars-locator -->
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>webjars-locator</artifactId>
  <version>0.46</version>
</dependency>
```

Ahora lo que nos queda es que los enlaces de la página hagan referencia a los elementos de la biblioteca de dependencias. Aquí nos puede ayudar Thymeleaf. Observe como queda el archivo de la página web

```
1 <!doctype html>
2 <html lang="en"
3   xmlns:th="http://www.thymeleaf.org">
4   <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
7     <title>Bootstrap demo</title>
8     <link th:rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}"/>
9   </head>
10  <body>
11    <h1>Hello, world!</h1>
12    <script type="text/javascript" th:src="@{/webjars/bootstrap/js/bootstrap.min.js}"></script>
13  </body>
14 </html>
```

El path relativo de la ubicación de las bibliotecas en las dependencias de Maven de nuestro proyecto se observan en la siguiente imagen



Observe que se debería haber realizado la referencia a la carpeta que indica la versión. Pero no se lo hizo debido a que se agregó la dependencia webjar locator, que se encarga de agregarla en tiempo de ejecución.

Para probar que esto funciona vamos a tener que crear un controlador, ya que estamos usando thymeleaf

Aquí el controlador

```
@Controller
public class HolaMundoController {

    @GetMapping("/hola")
    public String getHolaMundoMavenPage() {
        return "holamundomaven";
    }
}
```

Podemos ver que la página muestra el contenido formateado por Bootstrap

Hello, world!

Y también, si inspeccionamos el contenido del html generado podremos observar lo siguiente

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
6     <title>Bootstrap demo</title>
7     <link rel="stylesheet" type="text/css" href="/webjars/bootstrap/5.2.3/css/bootstrap.min.css"/>
8   </head>
9   <body>
10    <h1>Hello, world!</h1>
11    <script type="text/javascript" src="/webjars/bootstrap/5.2.3/js/bootstrap.min.js"></script>
12  </body>
13 </html>
```

Donde se ha reemplazado en las etiquetas link y script la versión correspondiente al CSS y JS correspondientes de Bootstrap de la aplicación. Como se ha comentado previamente esto lo realiza webjars locator.