# Aprende ASP:NET Core MVC 6 haciendo proyectos desde cero

# Sección 1: Introducción

## Video 1: Introducción

Nombrará las tecnologías y preparará el ambiente de desarrollo

Tecnologías: .NET, C# Y ASP.NET Core

Más adelante veremos proyectos con ASP:NET Core, que serán en Visual studio y usando DONETCLI

## Video 2: ¿Qué es .NET?

Es una plataforma de desarrollo que sirve para desarrollar distintos tipos de aplicaciones y funcionan en múltiples plataformas y para distintos dispositivos, NO es un lenguaje de programación, es un ambiente de trabajo.

Funciona con C#, EF# y Visual Basic, siento C# el más popular.

C# es un lenguaje multi paradigma con POO

.NET fue creado por Microsoft creado el 2002

.NET llegó hasta la versión 4.8, luego evolucionó para ser multi plataforma

.NET Core es la evolución de .NET y salió en el 2016

Como .NET llegó hasta la versión 4.8, cuando .NET Core empezó a subir de versiones, al momento de saltar de .NET Core 3 a 4, avanzó de la versión 3 a la 5, para evitar complicaciones.

Hasta la versión .NET 4.8, se le llamaba .NET Framework. Luego a los de marketing, a .NET Core 5 lo llamaron .NET 5, el cual no tiene nada que ver con .NET Framework 4.8 y desde ese entonces, a los .NET 5 hacia adelante se les llamra .NET Core 6, 7, 8, etc

## Video 3: ¿Qué es C#?

Es un lenguaje multi paradigmas creado por Microsoft en el 2000

Se pueden crear aplicaciones multiplataforma

Es un lenguaje fuertemente tipado, es decir, si escribes un código no válido, te darás cuenta

Está basado en clases, las cuales las puedes encapsular y puedes modular las clases

Permite trabajar con funciones de primera clase, es decir, puedes pasar funciones dentro de otras funciones, retornar funciones, asignar una función a una variable

## Video 4: ¿Qué es ASP.NET Core?

Es un framework para desarrollar aplicaciones web

Significa Active Server Pages, sirve para crear sitios web dinámicos

La misión es hacer más fácil la programación de las aplicaciones. Viene con un sistema para agregar usuarios a una plataforma, tiene un sistema de dependencias que ayuda a aplicar el principio de inversión de dependencias, es decir, nuestras clases dependan de abstracciones y no de tipos concretos.

A medida que evoluciona, hay mejoras en la velocidad

Al igual que la historia de .NET Framework y Core, para ASP.NET se repite la historia, la versión vieja (Framework) se utilizaba solo en Windows, mientras que la versión ASP.NET Core es multiplataforma.

## Video 5: Tipo de aplicaciones con ASP.NET Core

En el curso desarrollaremos aplicaciones con el patrón MVC

M: Modelo, se refiere a la data de la aplicación, la data puede ser un usuario, un profesor, etc, etc

V: Vista, es lo que el usuario va a utilizar

C: Controlador, se encarga de actualizar los modelos y le pasa los datos a la vista para que muestre a los usuarios el resultado final

Todo lo que hagas en MVC, puedes hacerlo en Razor Pages y viceversa

Puedes mezclar MVC con WEB APIs

**BLAZOR**

Lo que hace es enviar código de C# al navegador del usuario

Permite crear páginas interactivas

Existen 2 tipos, WebAssembly y Blazor Server

WebAssembly envía código de C# al navegador del usuario y lo ejecuta ahí mismo

Server lo que hacemos es ejecutar la aplicación remotamente des un servidor

**gRPC**

Es un framework que permite hacer llamadas de procedimiento remoto

La idea es gRPC es que permite comunicarnos de una manera muy rápida entre las distintas aplicaciones

Cuando tienes una aplicación con muchos microservicios, gRPC acelera bastante la velocidad de comunicación entre microservicios y se transforma en una buena opción para crear aplicaciones de esta estructura

## Video 6: Visual Studio y Visual Studio Code

En el curso usaremos un IDE, ya que traen integradas muchas opciones para facilitar la programación.

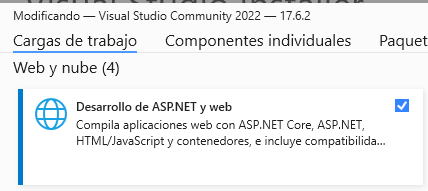
VS es un IDE de grandes ligas y se puede usar en Windows y MAC

No existe VS para Linux, pero si existe el VSC

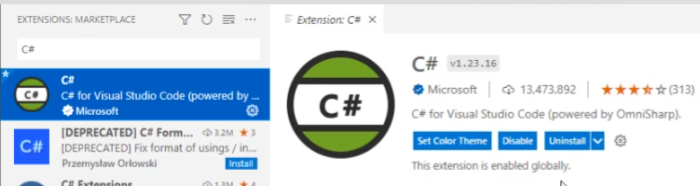
VSC es un editor de texto ligero y rápido

Famoso en ambientes de JS y usando framework como React, VUE o Angular

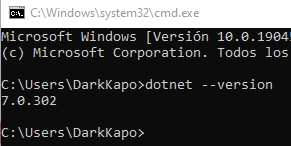
## Video 7: Instalando Visual Studio

Muestra dónde y cómo instalar VS y debemos agregar lo siguiente  


## Video 8: Instalando Visual Studio Code

Se puede usar para trabajar con .NET Core, para eso debes instalar la siguiente extensión  


## Video 9: Instalando .NET 6



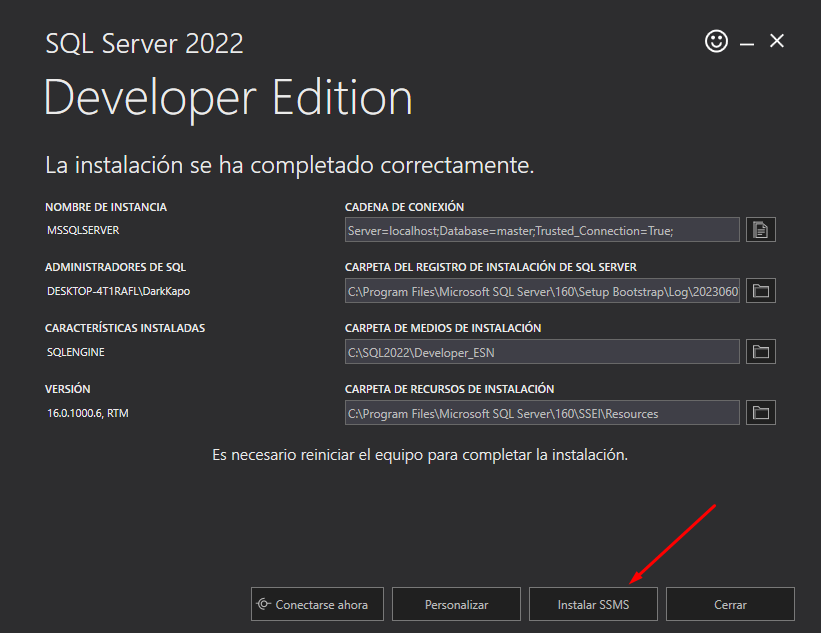
Según el curso, no hay problema en usar la versión 7.0

## Video 10: Instalando SQL server

Es el motor de base de datos que utilizaremos

<https://www.microsoft.com/es-cl/sql-server/sql-server-downloads>

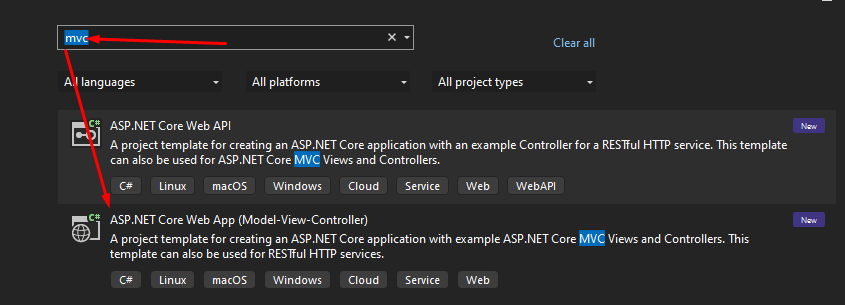
instalar el SSMS



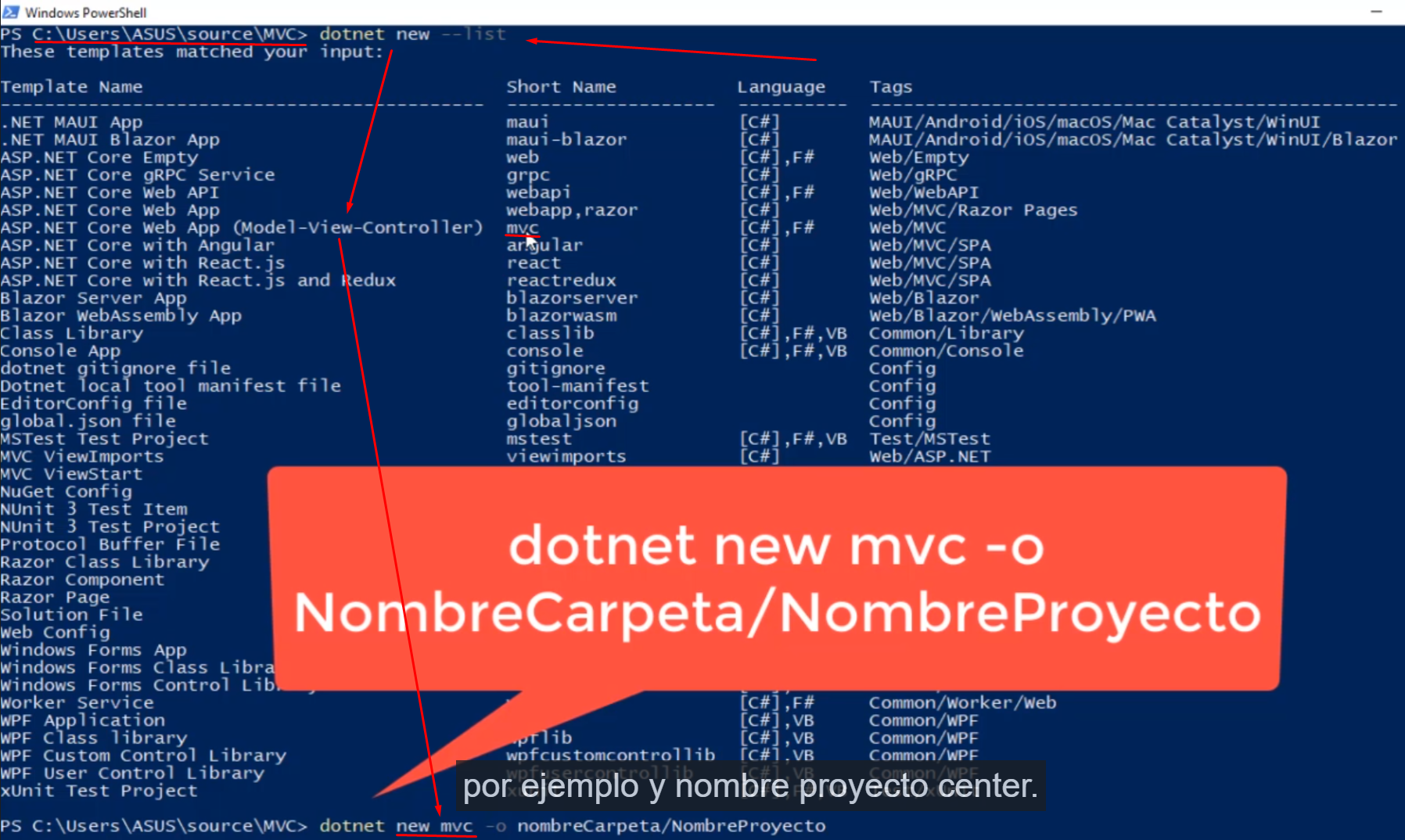
Cadena de conexión

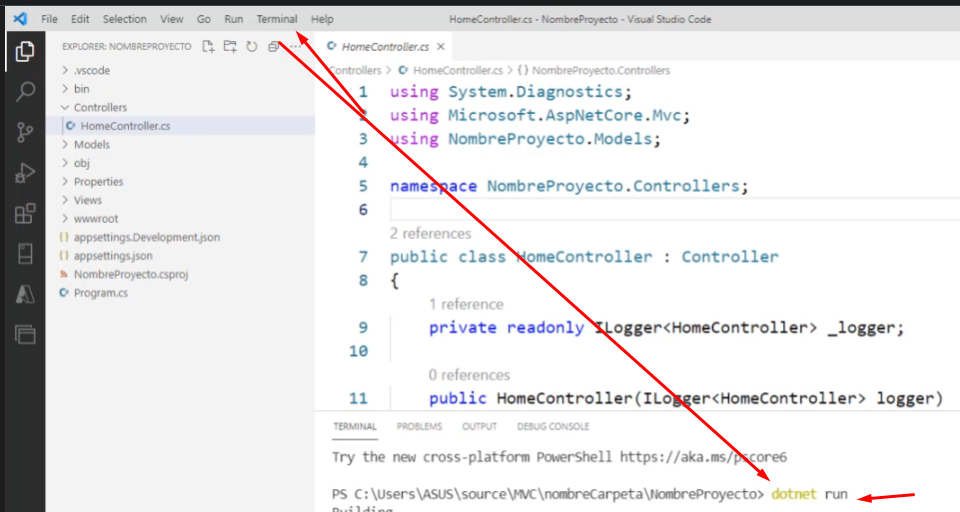
Server=localhost;Database=master;Trusted\_Connection=True;

## Video 11: Creando un Proyecto con Visual Studio



## Video 12: Creando un proyecto con Visual Studio Code



Para correr el proyecto  


## Video 13: Resumen

Comprobamos que .NET se puede usar para programar en cualquier plataforma usando C#

ASP.NET Core nos ayuda a hacer una aplicación mucho más fácil

Todo esto se hace en Visual Studio

## Video 14: Código fuente en Github

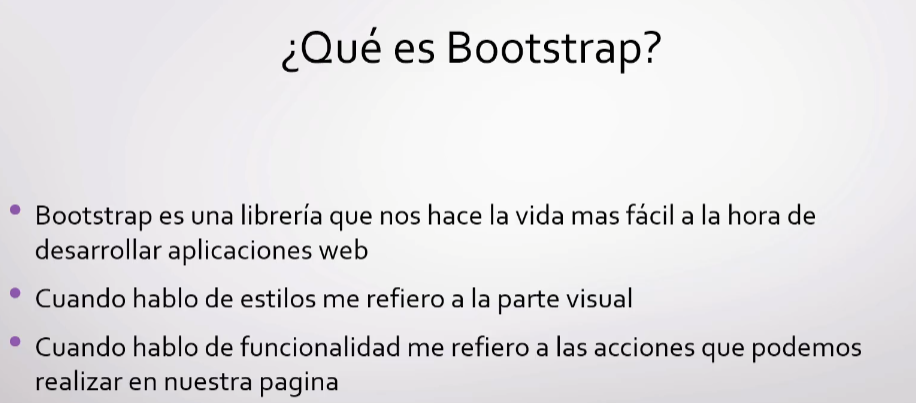
<https://github.com/gavilanch/Aprende-ASP-NET-Core-MVC-Haciendo-Proyectos/tree/main>

# Sección 2: Introducción a Bootstrap

## Video 15: Introducción

La idea es usar bottstrap para los proyectos, lka idea es que esta sección te enseñe a utilizarla

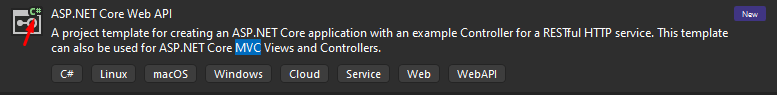
## Video 16: ¿Qué es bootstrap?

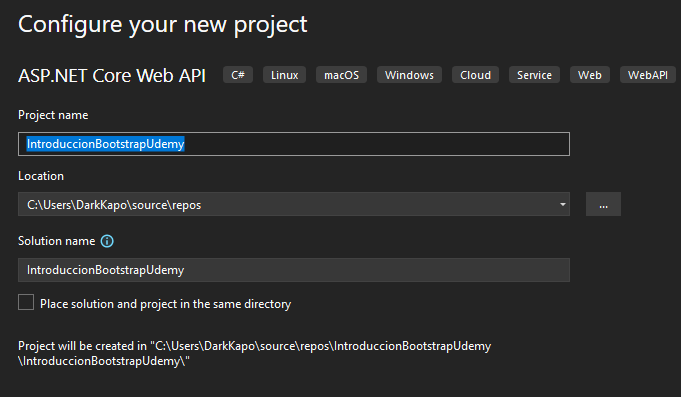


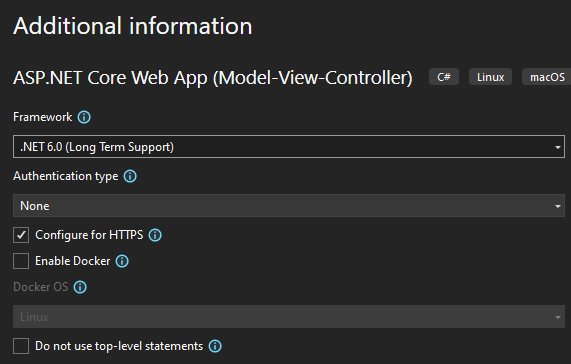
## Video 17: Contenedores

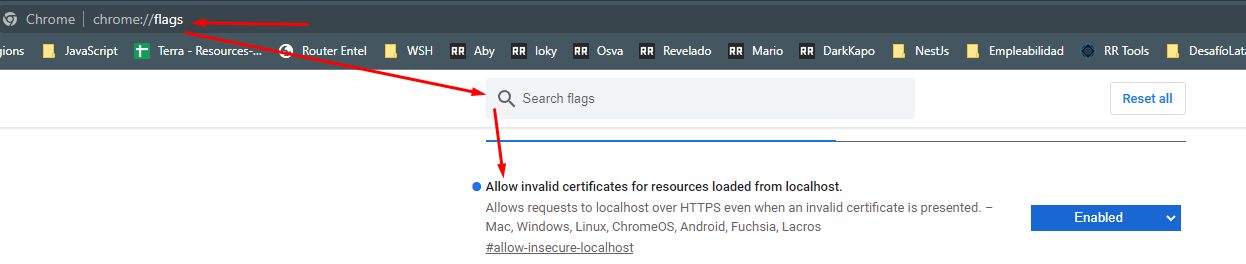
La documentación está en getbootstrap.com versión 5.1.3

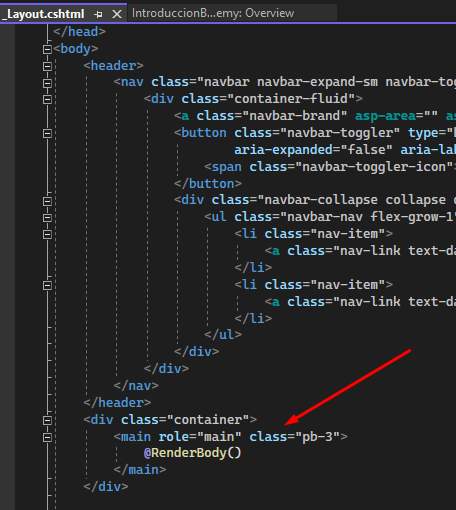
Cuando creas un proyecto MVC en visual studio este ya viene con bootstrap incluido



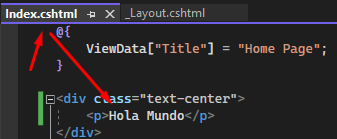




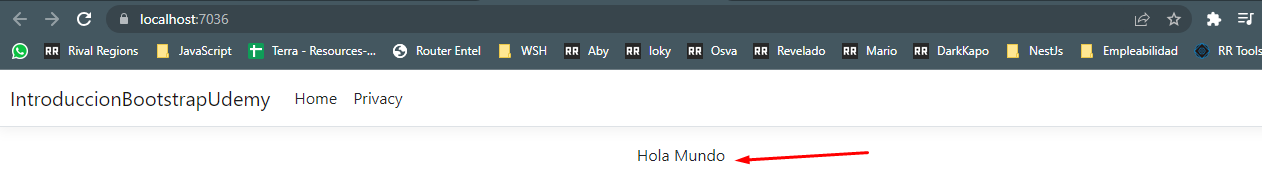
Error en la conexión por certificados incorrectos, para solucionar hacer esto:  


Los contenedores el bootstrap nos ayuda a centralizar la información, en el proyecto inicial, acá tenemos un contenedor  


Si cambiamos el RenderBody por cualquier otra cosa, veremos lo siguiente

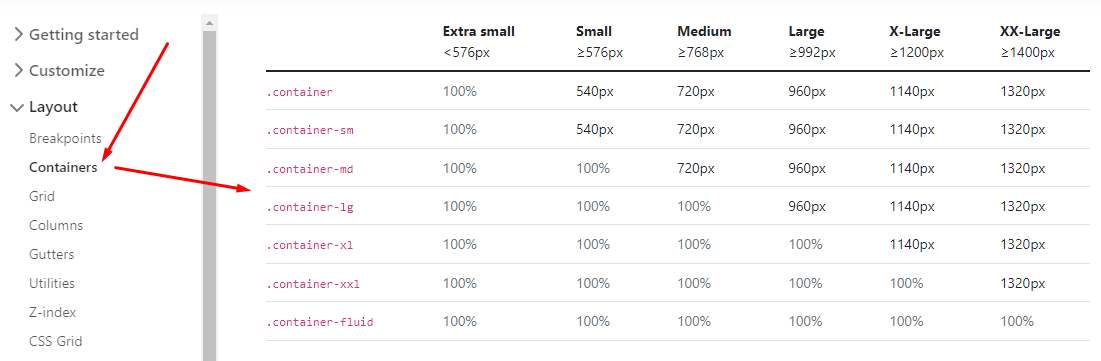


Ahora vamos a la página

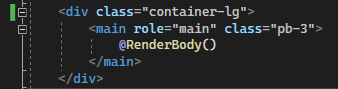


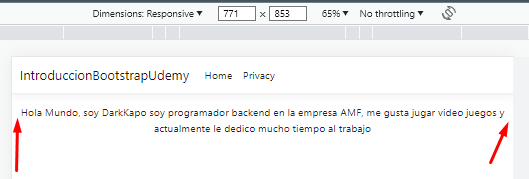
Lo que hay que entender, es que al usar la clase container, el contenido se centra, si quitas el container, el texto usará todo el ancho de la pantalla.

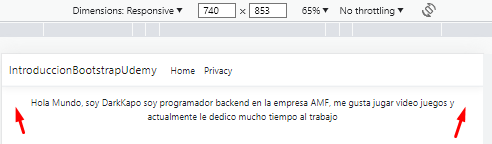
En la documentación de bootstrap tenemos más información para utilizar los márgenes



Vamos a fijarnos qué pasa con el botrde al usar el container-lg



Caso cuando es mayor a 768px (container-lg)  


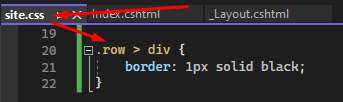
Caso cuando es menor a 768px (container-lg)  


Esto es lo que hay que saber de los contenedores, por ahora

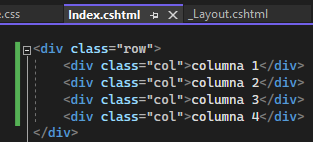
## Video 18: Grid system

Permite organizar los elementos en filas y columnas

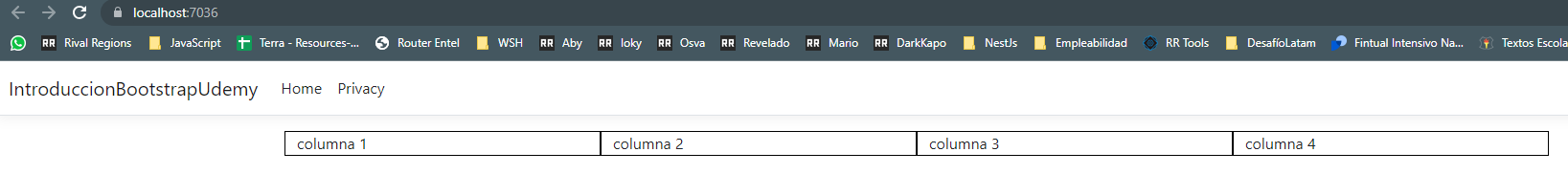
El primer ejercicio es agregar un borde a los div que están dentro de las filas



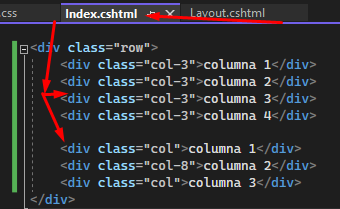
Agregar filas y columnas a la página principal

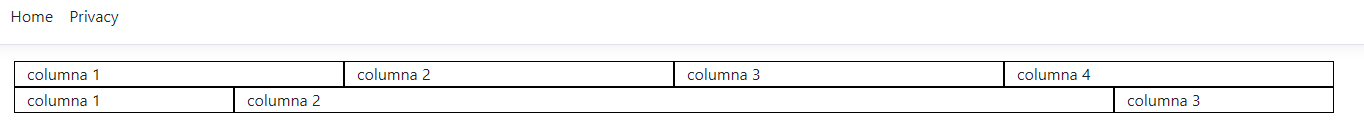


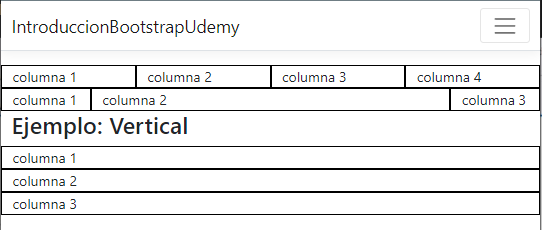
Ver el resultado



Por defecto, bootstrap funciona con columnas de 12, así que podemos jugar con esos valores

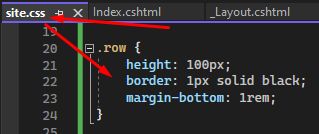


Lo importante es ver como el sistema distribuye las columnas  


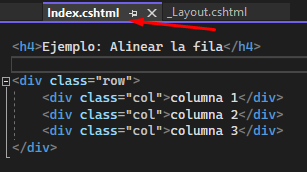
Vamos a probar jugando con los breakpoint, en este caso usaremos el sm  


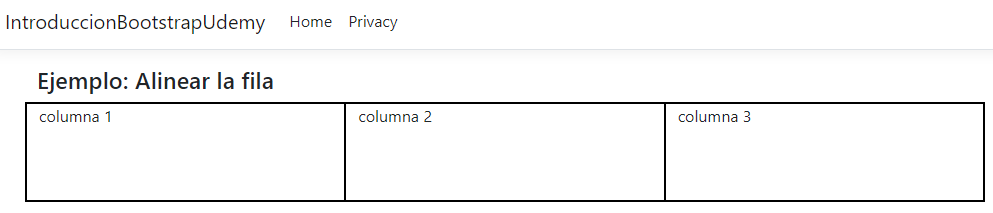
Para pantallas pequeñas, las columnas se ordenan de forma vertical

## Video 19: Columnas

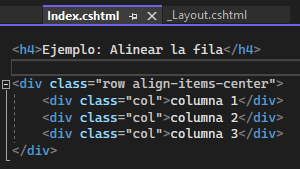
Antes, vamos a agregar un poco de css para luego jugar con las columnas  


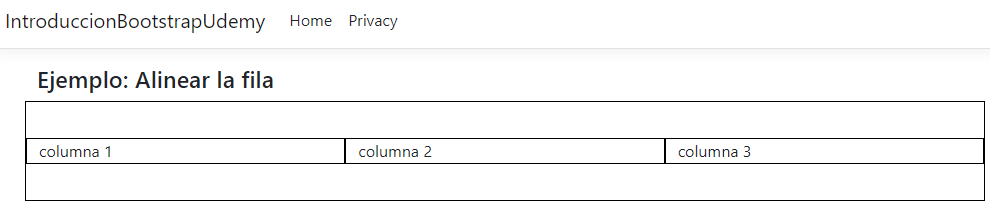
Agregar las columnas



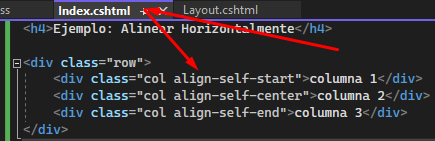


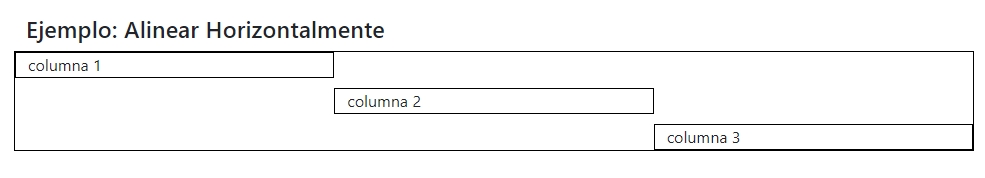
Alineando el texto de las columnas al centro



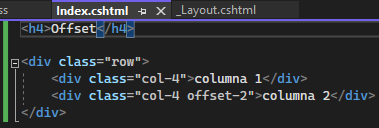


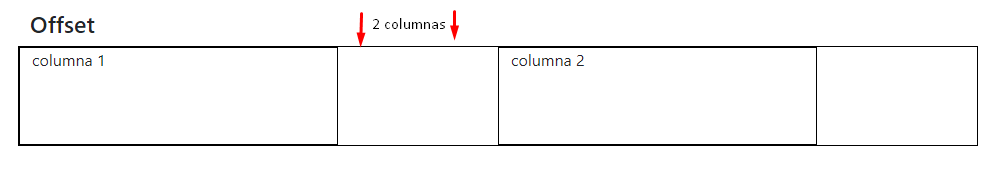
Recordar que previamente (CSS) definimos el row con una altura de 100px, lo importante es saber que, en este ejemplo, la fila está alineada al centro, puedes cambiar la palabra center por end o start y ver la nueva alineación



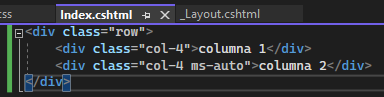


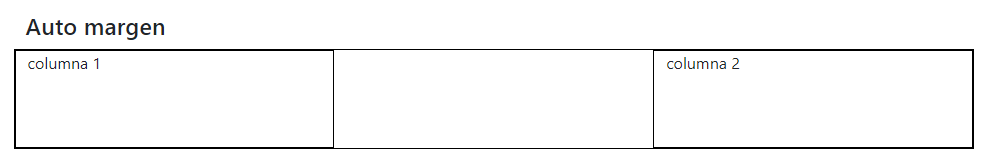
Ahora veremos para que sirve el offset de bootstrap (mueve las columnas de grid)





Ahora probaremos el ms auto, que alinea la lumna hacia la derecha

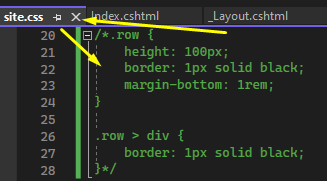




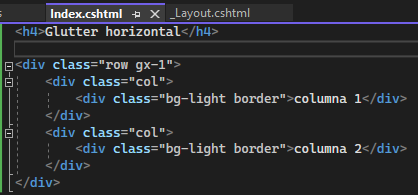
## Video 20: Gutters – Espacio entre columnas

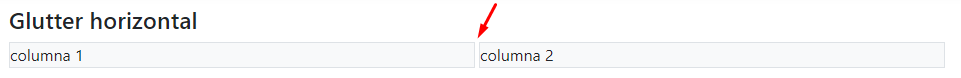
Puedes sumar espacio en las 4 direcciones

Comentar el código de CSS anteriormente agregado

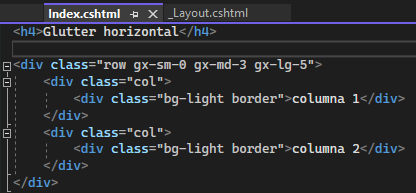


Usar glutter y ver resultado, puedes jugar con el número, el máximo es 5



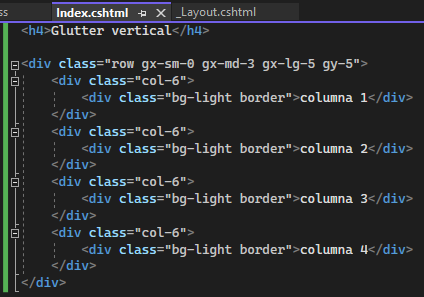


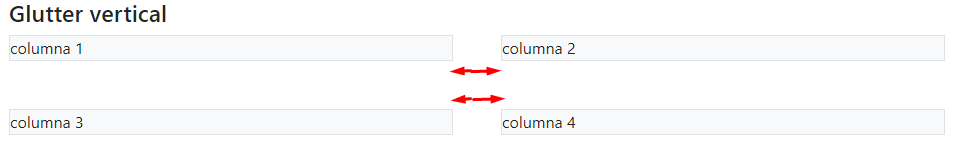
Vamos a probar los glutter en distintas resoluciones de pantalla



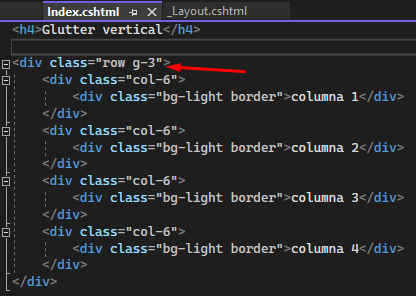
Ir al navegador y jugar con la pantalla

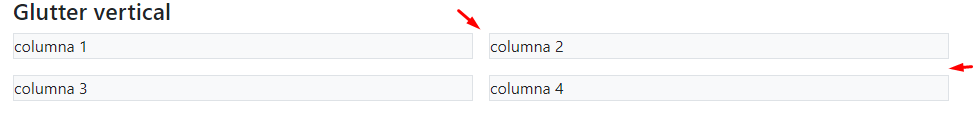
Probamos el glutter vertical





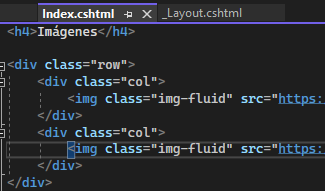
Probar glutter en todas direcciones

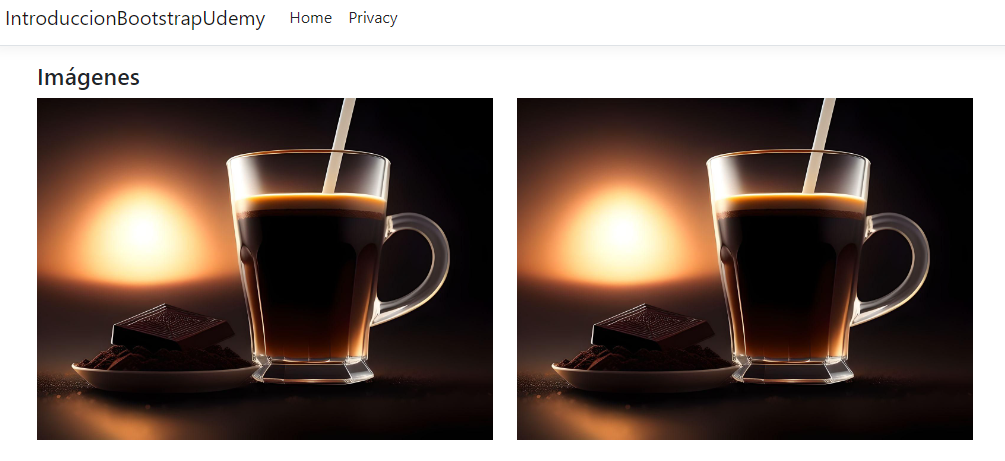




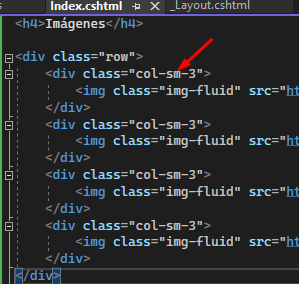
## Video 21: Manejando imágenes

La ida es manejar una imagen y que no se distorsione, para esto se usa la clase img-fluid



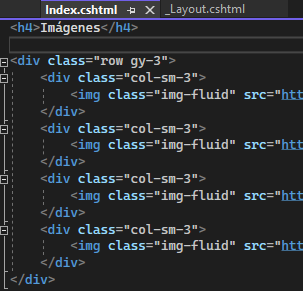


¿Cómo lo hacemos si para dispositivos pequeños queremos que se orden en vertical mientras que para dispositivos grandes lo queremos en horizontal?

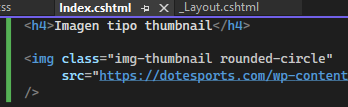


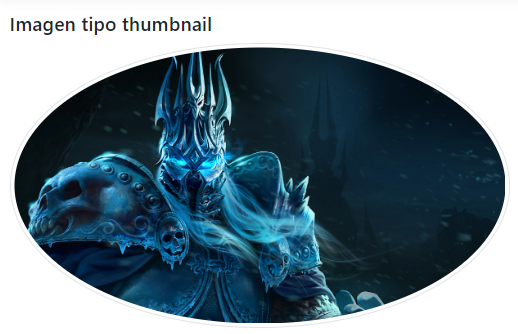


Ahora que sabemos usar glutter, podemos agregar una separación entre las imágenes

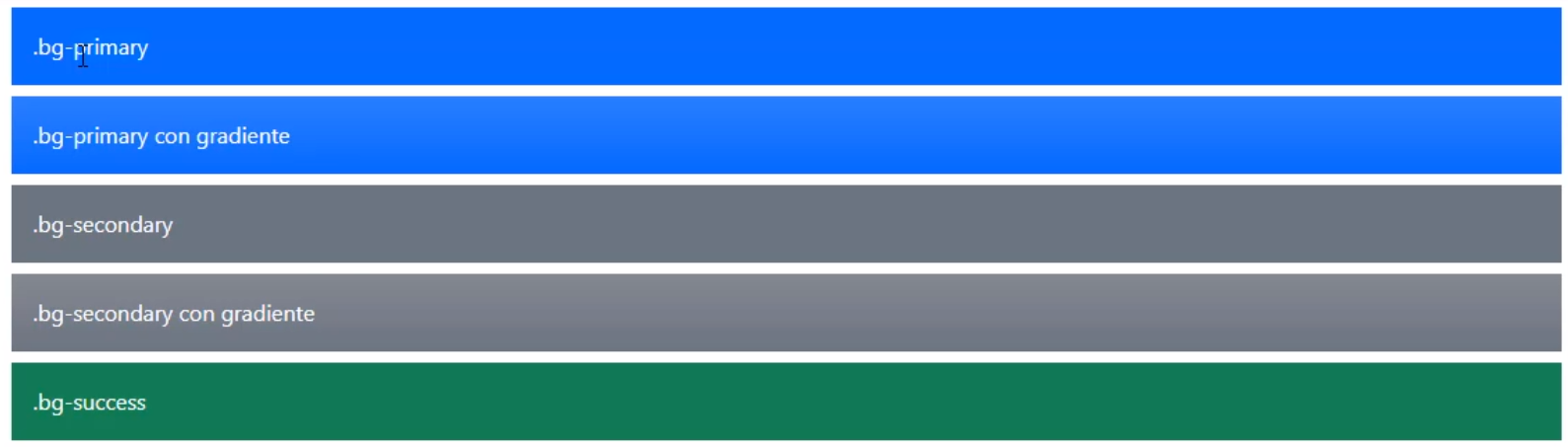


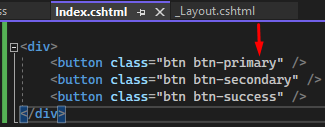
Crear una imagen tipo thumbnail





## Video 22: Colores

Se trata de los colores básicos que nos presenta bootstraps  






## Video 23: Tablas

### Código

1. <table **class**="table table-striped table-hover table-bordered table-sm">
2. <thead>
3. <tr>
4. <th>#</th>
5. <th>Nombre</th>
6. <th>Apellido</th>
7. </tr>
8. </thead>
9. <tbody>
10. <tr>
11. <td>1</td>
12. <td>Rodrigo</td>
13. <td>Quezada</td>
14. </tr>
15. <tr>
16. <td>2</td>
17. <td>Catalina</td>
18. <td>Quezada</td>
19. </tr>
20. <tr>
21. <td>3</td>
22. <td>Eduardo</td>
23. <td>Quezada</td>
24. </tr>
25. </tbody>
26. </table>

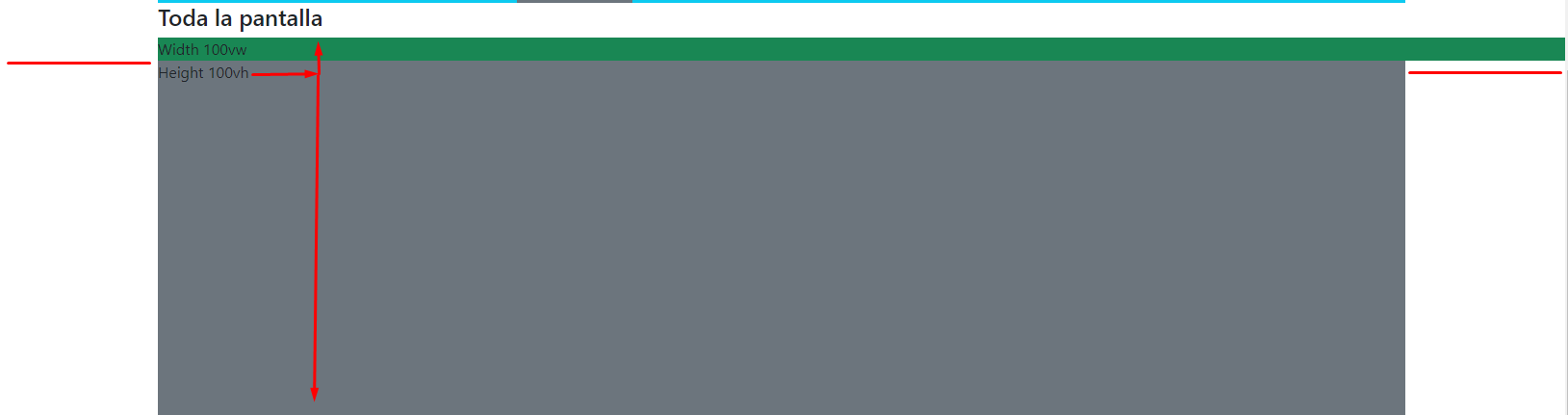
## Video 24: Manteniendo el ratio de un video

### Código

1. <div>
2. Con ratio:
3. </div>
5. <div **class**="ratio ratio-16x9">
6. <iframe width="560" height="315" src="https://www.youtube.com/embed/r5aIayl21CE" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
7. </div>
8. <hr />
10. <div **class**="row gy-3">
11. <div **class**="col-md-4">
12. <div **class**="ratio ratio-16x9">
13. <iframe width="560" height="315" src="https://www.youtube.com/embed/r5aIayl21CE" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
14. </div>
15. </div>
16. <div **class**="col-md-4">
17. <div **class**="ratio ratio-16x9">
18. <iframe width="560" height="315" src="https://www.youtube.com/embed/r5aIayl21CE" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
19. </div>
20. </div>
21. <div **class**="col-md-4">
22. <div **class**="ratio ratio-16x9">
23. <iframe width="560" height="315" src="https://www.youtube.com/embed/r5aIayl21CE" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
24. </div>
25. </div>
26. </div>

## Video 25: Tamaños

El width 100vw ocupa toda la pantalla en horizontal, la franja verde que está a la derecha es porque a la izquierda existe un marguen por el contenedor, así que el efecto que se ve es que la línea está corrida hacia la izquierda



### Código

1. <h4>Anchura</h4>
3. <div **class**="w-25 p-3 bg-primary">Width 25%</div>
4. <div **class**="w-50 p-3 bg-secondary">Width 50%</div>
5. <div **class**="w-75 p-3 bg-primary">Width 75%</div>
6. <div **class**="w-100 p-3 bg-secondary">Width 100%</div>
7. <div **class**="w-auto p-3 bg-primary">Width auto</div>
9. <h4>Altura</h4>
11. <div style="height: 100px;" **class**="bg-info">
12. <div **class**="h-25 d-inline-block bg-primary" style="width: 120px;">Height 25%</div>
13. <div **class**="h-50 d-inline-block bg-secondary" style="width: 120px;">Height 50%</div>
14. <div **class**="h-75 d-inline-block bg-primary" style="width: 120px;">Height 75%</div>
15. <div **class**="h-100 d-inline-block  bg-secondary" style="width: 120px;">Height 100%</div>
16. <div **class**="h-auto d-inline-block  bg-primary" style="width: 120px;">Height auto%</div>
17. </div>
19. <h4>Toda la pantalla</h4>
21. <div **class**="vw-100 bg-success">Width 100vw</div>
22. <div **class**="vh-100 bg-secondary">Height 100vh</div>

## Video 26: Espacios

Se trata de la separación entre elementos

### Código

1. <h4>Márgenes</h4>
2. @\*Marguen en eje y\*@
3. <div **class**="my-1">fila 1</div>
4. @\*marguen para todoss lados\*@
5. <div **class**="m-0">fila 2</div>
6. @\*Marguen en eje x\*@
7. <div **class**="mx-0">fila 3</div>
9. @\*p = padding, px = eje x, pe = end\*@
10. <button **class**="btn btn-primary p-5">btn 1</button>
11. @\*ms = margin Start, me = End\*@
12. <button **class**="btn btn-secondary ms-5">btn 2</button>

## Video 27: Utilidades de Texto

1. <h4>Alineando texto</h4>
3. <p **class**="text-start">texto a la izqueirda</p>
4. <p **class**="text-center">texto en el centro</p>
5. <p **class**="text-end">texto a la derecha</p>
7. <p **class**="text-sm-center">texto centralizado en pantallas </p>
8. <p **class**="text-md-center">texto centralizado en pantallas </p>
9. <p **class**="text-lg-center">texto centralizado en pantallas </p>
10. <p **class**="text-xl-center">texto centralizado en pantallas </p>
12. <h2>Texto dentro de su contenedor</h2>
13. @\*sirve para textos con espacios\*@
14. <div **class**="badge bg-primary text-wrap" style="width: 6rem">
15. Este texto no se sale de su contenedor
16. </div>
17. @\*sirve para textos sin espacios\*@
18. <div **class**="badge bg-primary text-wrap text-break" style="width: 6rem">
19. mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm
20. </div>
22. <h2>Transformando texto</h2>
24. <p **class**="text-lowercase">Minúsculas.</p>
25. <p **class**="text-uppercase">Mayúsculas texto..</p>
26. <p **class**="text-capitalize">cada palabra con mayúscula.</p>
28. <h2>Tamaño de texto</h2>
30. <p **class**="fs-1">.fs-(1 al 6)</p>
32. <h2>Estilo del texto</h2>
33. <p **class**="fw-bold">Negrita</p>
34. <p **class**="fw-bolder">Más negrita</p>
35. <p **class**="fw-normal">Normal</p>
37. <p **class**="fw-light">Texto light</p>
38. <p **class**="fw-lighter">Más light</p>
39. <p **class**="fst-italic">Cursiva</p>
41. <p **class** ="text-decoration-underline">Subrayado </p>
42. <p **class**="text-decoration-line-through">Línea en el texto </p>
44. <a href="#" **class**="">Enlace con decoración</a>
45. <br />
46. <a href="#" **class**="text-decoration-none">Enlace sin decoració</a>

## Video 28: Formularios

### Código

1. <h2>Formulario normal</h2>
2. <form>
3. <div **class**="mb-3">
4. <label **class**="form-label">Email</label>
5. <input type="email" **class**="form-control" />
6. </div>
7. <div **class**="mb-3">
8. <label **class**="form-label">Descripción</label>
9. <textarea type="text" **class**="form-control" rows="3" ></textarea>
10. </div>
11. <div **class**="mb-3">
12. <label **class**="form-label">Color favorito</label>
13. <**select** **class**="form-select">
14. <option **value**="1">Rojo</option>
15. <option **value**="2">Azul</option>
16. <option **value**="3">Verde</option>
17. </**select**>
18. </div>
19. <div **class**="form-check">
20. <input **class**="form-check-input" type="checkbox" [checked](http://www.google.com/search?q=checked+msdn.microsoft.com) />
21. <label **class**="form-label">Es empleado</label>
22. </div>
23. <div **class**="form-check">
24. <input **class**="form-check-input" type="radio" name="miradio" />
25. <label> Opción 1</label>
26. </div>
27. <div **class**="form-check">
28. <input **class**="form-check-input" type="radio" name="miradio" />
29. <label> Opción 2</label>
30. </div>
31. <div **class**="form-floating mb-3">
32. <input type="email" **class**="form-control" placeholder="email" />
33. <label>Email</label>
34. </div>
35. <div **class**="mb-3">
36. <label **class**="form-label">Email</label>
37. <input type="email" **class**="form-control" disabled />
38. </div>
39. <div **class**="mb-3">
40. <label>Color de la camisa</label>
41. <input type="color" **class**="form-control form-control-color" **value**="#563d7c" />
42. </div>
43. <div>
44. <button **class**="btn btn-primary">Enviar</button>
45. <button **class**="btn btn-secondary">Limpiar</button>
46. </div>
47. </form>
49. <h2>Formulario con columnas</h2>
51. <div **class**="row">
52. <div **class**="col">
53. <input type="text" **class**="form-control" placeholder="Primer nombre" />
54. </div>
55. <div **class**="col">
56. <input type="text" **class**="form-control" placeholder="Primer apellido" />
57. </div>
58. <div **class**="col">
59. <button **class**="btn btn-primary">Enviar</button>
60. </div>
61. </div>

## Video 29: Acordeón

### Código

1. <div **class**="accordion" id="ejemploAcordeon">
2. <div **class**="accordion-item">
3. <h2 **class**="accordion-header">
4. @\*Target apunta al contenido que va a mostrar\*@
5. <button data-bs-toggle="collapse" data-bs-target="#elemento1acordeon"
6. **class**="accordion-button">Elemento 1</button>
7. </h2>
8. @\*Parent dice que es hijo del elemento seleccionado\*@
9. <div id="elemento1acordeon" **class**="accordion-collapse collapse show"
10. data-bs-parent="#ejemploAcordeon">
11. <div **class**="accordion-body">
12. Este es el contenido del primer acordeón
13. </div>
14. </div>
15. </div>
16. <div **class**="accordion-item">
17. <h2 **class**="accordion-header">
18. <button data-bs-toggle="collapse" data-bs-target="#elemento2acordeon"
19. **class**="accordion-button">Elemento 2</button>
20. </h2>
21. <div id="elemento2acordeon" **class**="accordion-collapse collapse show"
22. data-bs-parent="#ejemploAcordeon">
23. <div **class**="accordion-body">
24. Este es el contenido del segundo acordeón
25. </div>
26. </div>
27. </div>
28. </div>

## Video 30: Alertas

### Código

1. <button onclick="mostrarAlertaExitosa()">Mostrar alerta exitosa</button>
2. <button onclick="mostrarAlertaError()">Mostrar alerta de error</button>
4. <div>
5. Las alertas se van a mostrar aquí
6. </div>
8. <div id="contenedor-alertas">
10. </div>
12. @section Scripts {
13. <script>
14. function mostrarAlertaExitosa(){
15. mostrarAlerta("Este es un mensaje exitoso", "primary");
16. }
17. function mostrarAlertaError() {
18. mostrarAlerta("Este es un mensaje de error", "danger");
19. }
21. function mostrarAlerta(mensaje, tipo) {
22. **const** alerta = document.createElement("div")
23. alerta.innerHTML = `<div **class**="alert alert-${tipo}" role="alert">
24. ${mensaje}
25. </div>`;
26. **const** contenedorAlertas = document.getElementById("contenedor-alertas");
27. contenedorAlertas.append(alerta)
28. }
29. </script>
30. }

## Video 31: Carrusel

### Código

1. <div id="carouselEjemplo" **class**="carousel slide" data-bs-ride="carousel">
2. <div **class**="carousel-inner">
3. <div **class**="carousel-item active">
4. <img src="https://denomades.s3.us-west-2.amazonaws.com/blog/wp-content/uploads/2015/10/05164133/salar-de-tara\_opt1.jpg" **class**="d-block w-100" />
5. </div>
6. <div **class**="carousel-item">
7. <img src="https://denomades.s3.us-west-2.amazonaws.com/blog/wp-content/uploads/2015/10/05164129/la-junta-cochamo\_opt1.jpg" **class**="d-block w-100" />
8. </div>
9. <div **class**="carousel-item">
10. <img src="https://denomades.s3.us-west-2.amazonaws.com/blog/wp-content/uploads/2015/10/05164144/ventisquero-colgante-queulat\_opt1.jpg" **class**="d-block w-100" />
11. </div>
13. <div **class**="carousel-indicators">
14. <button type="button" data-bs-target="#carouselEjemplo" data-bs-slide-to="0" **class**="active"></button>
15. <button type="button" data-bs-target="#carouselEjemplo" data-bs-slide-to="1"></button>
16. <button type="button" data-bs-target="#carouselEjemplo" data-bs-slide-to="2"></button>
17. </div>
19. <button **class**="carousel-control-prev" type="button" data-bs-target="#carouselEjemplo" data-bs-slide="prev">
20. <span **class**="carousel-control-prev-icon"></span>
21. </button>
23. <button **class**="carousel-control-next" type="button" data-bs-target="#carouselEjemplo" data-bs-slide="next">
24. <span **class**="carousel-control-next-icon"></span>
25. </button>
26. </div>
27. </div>

## Video 32: Ventanas modales

### Código

1. <button **class**="btn btn-primary" data-bs-toggle="modal" data-bs-target="#ejemploModal">
2. Abrir modal
3. </button>
5. <div **class**="modal fade" id="ejemploModal" tabindex="-1">
6. <div **class**="modal-dialog">
7. <div **class**="modal-content">
8. <div **class**="modal-header">
9. <h5 **class**="modal-title">Título modal</h5>
10. <button type="button" **class**="btn-close"></button>
11. </div>
12. <div **class**="modal-body">
13. <p>Este es un pequeño formulario</p>
14. <form>
15. <div **class**="mb-3">
16. <label **class**="form-label">Nombre</label>
17. <input type="text" **class**="form-control" />
18. </div>
19. </form>
20. <div **class**="modal-footer">
21. <button **class**="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>
22. <button **class**="btn btn-primary" data-bs-dismiss="modal">Guardar</button>
23. </div>
24. </div>
25. </div>
26. </div>
27. </div>

## Video 33: Tabs

### Código

1. <ul **class**="nav nav-tabs">
2. <li **class**="nav-item">
3. <button **class**="nav-link active" data-bs-toggle="tab" data-bs-target="#tab1">
4. Tab 1
5. </button>
6. </li>
8. <li **class**="nav-item">
9. <button **class**="nav-link" data-bs-toggle="tab" data-bs-target="#tab2">
10. Tab 2
11. </button>
12. </li>
13. </ul>
15. <div **class**="tab-content">
16. <div **class**="tab-pane fade show active" id="tab1">
17. Buenos dí**as**
18. </div>
19. <div **class**="tab-pane fade" id="tab2">
20. Buenas noches
21. </div>
22. </div>

## Video 34: Menú lateral con offCanvas

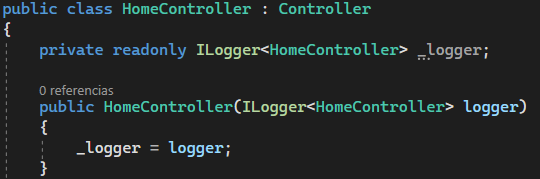
### Código

1. <button **class**="btn btn-primary" data-bs-toggle="offcanvas" data-bs-target="#ejemploOffcanvas">
2. Abrir offcanvas
3. </button>
5. <div **class**="offcanvas offcanvas-start" tabindex="-1" id="ejemploOffcanvas">
6. <div **class**="offcanvas-header">
7. <h5 **class**="offcanvas-title">Mi aplicación</h5>
8. <button type="button" **class**="btn-close text-reset" data-bs-dismiss="offcanvas"></button>
9. </div>
10. <div **class**="offcanvas-body">
11. <div>
12. Un texto de ejemplo
13. </div>
14. </div>
15. </div>

Agregar esto al Word principal

## Video 57: Inyección de dependencias

Tenemos un ejemplo de inyección de dependencia en el mismo controlador

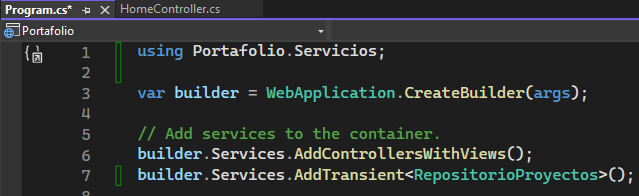


En vez de instanciar el objeto dentro del controlador, como lo hicimos con RepositorioProyectos, en controlador del Ilogger lo recibe como parámetro. La idea es hacer lo mismo con el RepositorioProyectos.

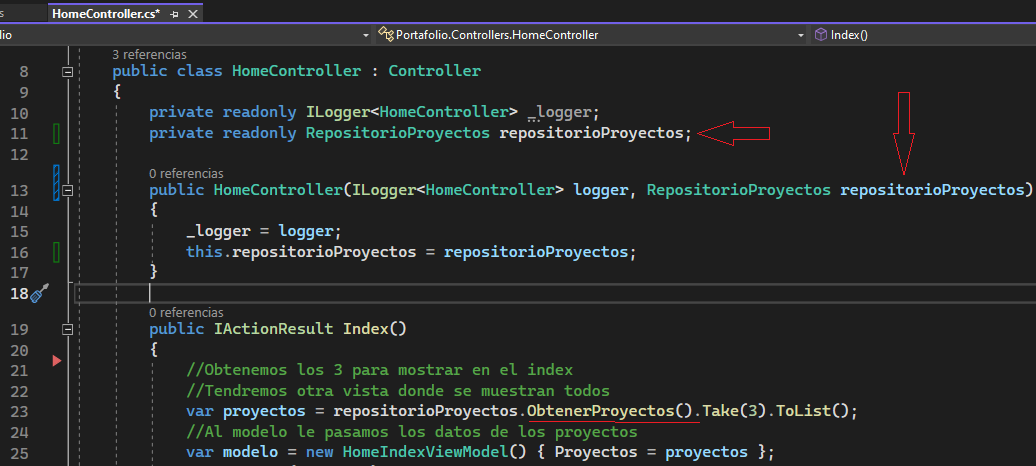
La ventaja es que logramos quitar la responsabilidad del controlador y centralizamos la configuración de las clases. Otra ventaja es que, al tener las instancias de forma independiente, es que nos ahorramos en tener que instanciar clases cuando tienes que pasarlas por parámetros. Ejemplo



Toda esta configuración de dependencia la haremos en un servicio y lo inyectamos en el archivo Program.cs



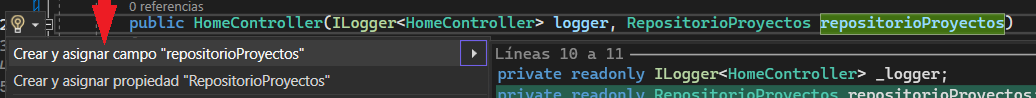
Con lo anterior logramos configurar la clase RepositorioProyectos en el sistema de control de dependencias, ahora puedes instanciar la clase RepositorioProyectos en el controlador de la siguiente forma



Ahora vemos en la línea 23 que podemos usar el método ObtenerProyectos sin necesidad de instanciar la clase dentro del controlador

**Nota**

Cuando instancias repositorioProyectos (línea 13) debes presionar ctrl + . y elegir la opción

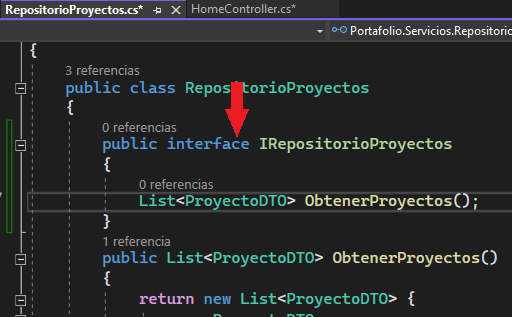


Con lo anterior logramos quitar la dependencia del controlador. Pero esto se puede mejorar un poco más utilizando una **interface**

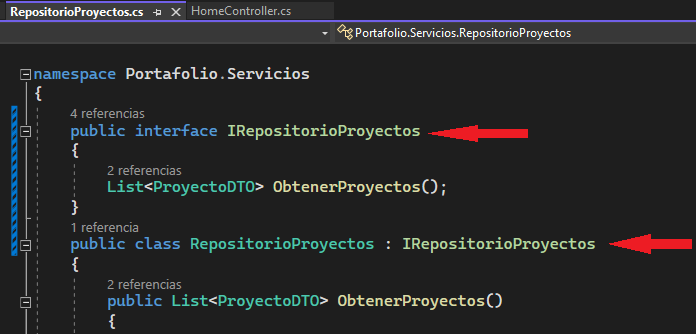
## Video 58: Inyección de dependencias con interfaces

La interface agrega flexibilidad a las clases.

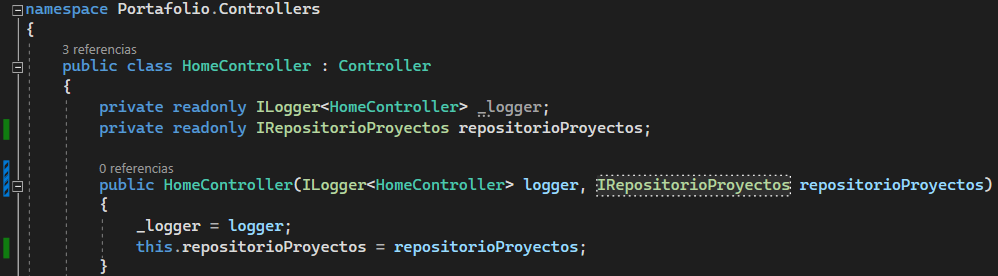
La interface es un “contrato” que nos obliga a implementar ciertos métodos de una clase, esta interfaz la creamos de la siguiente forma



Corregir la imagen y colocar esto



Y el controlador debe quedar de la siguiente forma



Correr el proyecto y verificar que aún se muestre la lista de proyectos

Ahora va a explicar que es el AddTransient que está en Program.cs

## Video 59: Servicios