Seccion 2: Introduccion a Razor

12. Introducción al Módulo 2

Antes de hablar de Blazor, debemos conocer Razor.

Sobre una sintaxis que nos permite combinar código de HTML y C Sharp.

Hola, Mi nombre es Felipe Gavilán.

Y en este módulo hablaremos de Razor.

Comenzaremos dando una introducción a Razor, veremos que es y cómo utilizarlo.

Más adelante aprenderemos a escribir código de Razor.

Específicamente veremos cómo declarar variables en Razor, el uso de condicionales y bucles en el código

y hablaremos de cómo utilizar código HTML arbitrario en un archivo de Razor.

13. Introducción a Razor

Vamos a ver una pequeña introducción a Razor.

Aquí estamos en Visual Studio, en nuestro proyecto de Play Store.

Películas que creamos en el módulo anterior.

Y lo que vamos a hacer es que vamos a venir acá al suelo.

Son las plotter, donde podemos visualizar los distintos archivos de nuestra solución.

Y yo voy a venir al Proyecto Client, al proyecto Blaze of Películas Client, voy a venir a pies y voy

a venir a Index, porque aquí en Index yo puedo escribir código de Razor.

Entonces yo lo que voy a hacer es que yo voy a borrar todo esto solamente dejando esto de Page Slash

Esto de Page Slash básicamente me permite indicar a través de que where l yo puedo acceder a este componente.

Vamos a ver que es un componente en el siguiente módulo, pero por ahora tú puedes imaginar que index

registro es un componente.

Fetch es un componente, Counter es otro componente.

Es decir, son como partes distintas de mi aplicación, como páginas distintas de mi aplicación.

Entonces yo estoy indicando que con esta directiva page, yo estoy indicando que en el root de mi aplicación

yo puedo entrar a este componente index que yo tengo acá y que yo voy a ver en ese componente.

Pues lo que yo ponga acá, cualquier cosa que yo ponga acá es lo que le va a salir al usuario en la

pantalla.

Por ejemplo, si yo pongo una P de párrafo, vamos a ver que yo puedo escribir código de HTML en esta

página, en este archivo de puntos razor y yo puedo decir, por ejemplo Hola y vamos a presionar control

F5 para ejecutar nuestra aplicación.

Vamos a ver que efectivamente estamos en el root de la aplicación y aquí tenemos nuestro Hola!

Pero obviamente esto está muy aburrido como un simple hola.

Así que vamos a hacer algo más interesante.

El chiste de Ray Short es que me permite combinar código de HTML y See Sharp de una manera muy sencilla.

Vamos a ver esto.

Vamos a decir que además de Hola, yo quiero saludar a una persona con su nombre, por ejemplo, yo

quiero decir hola Felipe, y eso está bien.

Si yo salvo yo puedo volver para acá y vamos a ver que tenemos.

Hola Felipe.

Sin embargo, quizás yo no quiero siempre mostrar Felipe porque yo puedo tener distintos usuarios con

distintos nombres.

Entonces yo lo que voy a hacer es que en vez de codear Felipe acá yo voy a colocar una variable para

yo colocar código de Sharp en un archivo de Razor.

Yo utilizo la directiva Code, lo cual me permite definir un bloque de código donde yo voy a poder colocar

código de Sharp.

Por ejemplo, aquí yo voy a decir String nombre igual a Felipe, punto y coma por acá y luego yo voy

a venir acá.

Y para yo colocar el valor esta variable acá, yo tengo que decir arroba con el arroba.

Yo lo estoy indicando a Roy Short que yo quiero colocar código de sí sharp.

Así, lo que sea que yo escriba luego de la arroba es una expresión de sharp.

En mi caso yo quiero colocar nombre que es esta variable que tenemos acá, y fíjate lo chévere de esto.

Y es que con yo decir arroba y nombre, en este caso yo estoy indicando que yo estoy tratando de colocar

el valor de la variable nombre.

Sin embargo, luego de nombre yo tengo este código de HTML y no hay ningún problema con eso.

Razor es lo bastante inteligente para entender que esto es código de HTML.

Esto es código de Sharp y esto es código de HTML.

Si yo tener que definir algún tipo de separador explícito entre mi código de HTML y mi código de sharp.

A esto que tenemos aquí, por cierto, le llamamos una expresión implícita de Roy Short.

Pues yo no he colocado nada que indique dónde empieza y donde termina la expresión de Sharp.

Una expresión explícita la veremos en un momento.

Por ahora vamos a salvar y vamos a venir acá y vamos a ver que se va a refrescar la página.

Y tenemos.

Hola Felipe, pero si yo mañana cambio esto por Claudia por ejemplo, o este nombre podría venir de

una base de datos o de algún otro lugar, no hay ningún problema, porque vamos a ver, que al yo cambiar

el valor de la variable, vamos a ver que aquí se cambia el nombre de la persona y mostramos.

Hola Claudia.

Claro está que yo puedo hacer más que simplemente mostrar el valor de una variable.

Por ejemplo, vamos a decir que yo tengo Felipe, pero yo quiero colocarlo en mayúscula, pues si yo

digo punto, acá vamos a ver que nos sale el intelligence con las distintas funciones que yo puedo utilizar

para trabajar con un string, y una de ellas es tu opera, que me permite convertir en mayúsculas un

string.

Entonces, si yo salgo y volvemos para acá, vamos a ver que luego de que refresque tenemos.

Hola Felipe, pero en mayúscula.

Lo que quería decir que yo tengo bastante versatilidad a la hora de manipular mis variables, lo cual

yo no podría hacer en el caso de que fuera un texto HTML normal como este.

Hola que yo tengo por acá también yo puedo definir funciones aquí.

Por ejemplo, vamos a decir que esto de colocar en mayúscula yo lo quiero tener.

Era una función.

Yo puedo hacer eso.

Yo puedo decir String.

Transformar.

Vamos a decir transformar.

Y esta va a ser una función que recibe un string como parámetro.

Y yo voy a decir aquí Valor, tú.

Punto y coma.

Y ahora yo lo que voy a hacer es que yo voy a venir acá y yo voy a decir transformar.

Fíjate que me sale transformar aquí, como es una función, yo abro paréntesis para invocarla y le paso

el parámetro que yo quiera, que en este caso es la variable nombre.

Y vamos a ver que si yo salgo y déjame colocar otro nombre solamente para que veamos alguna diferencia,

si yo salgo y volvemos hacia Google Chrome, vamos a ver que efectivamente tenemos.

Hola Guillermo en mayúscula te decía que podemos tener expresiones implícitas o explícitas de Razor.

Este es un ejemplo de una expresión implícita, porque no marcamos un inicio ni.

En fin, vamos a ver un ejemplo de una expresión explícita.

Por ejemplo, vamos a decir que yo quiero tener un párrafo y yo quiero decir vamos a sumar dos más dos.

Muy bien, pero yo quiero hacer eso con Sharp.

Entonces, qué yo hago?

Bueno, ya sabemos que yo tengo que utilizar un arroba para indicar que vamos a utilizar una expresión

de código.

Sin embargo, si yo digo dos más dos, de esta forma vamos a tener un error.

Porque si ella no sabe que yo quiero tener todo esto en una misma expresión de Sharp, entonces lo que

podemos hacer es utilizar paréntesis.

Utilizamos paréntesis para delimitar qué es una expresión de si sharp después de este arroba, que es

entonces este dos más dos que tenemos acá.

Por eso lo llamamos una expresión explícita, porque yo de manera explícita tengo que decir dónde inicia

y dónde termina mi expresión de Sharp.

Ahora, si yo salgo y volvemos hacia Google, vamos a ver que tenemos que dos más dos es igual a cuatro,

algo para lo cual utilizaremos bastante en este curso.

Las expresiones explícitas es para utilizar expresiones lambda para definir el evento de un control

de HTML.

Por ejemplo, vamos a decir aquí botón, yo voy a decir aquí botón, yo voy a crear un botón, presionarme,

presionarme y yo voy a decir aquí un clic o un clic.

Básicamente con esto yo estoy diciendo que vamos a definir un evento onClick, es decir, que vamos

a definir una acción que va a ocurrir cuando el usuario haga clic sobre el botón y como yo coloco el

arroba, pues se entiende que va a ser un código de Sharp el que se va a ejecutar.

Si yo quisiera que fuera código de JavaScript, vamos a ver más de JavaScript más adelante en el curso.

Pero si yo quisiera que fuera de JavaScript, tú sabes que podemos decir onclick sin el arroba, pero

como es una expresión de Sharp la que queremos utilizar, utilizamos el arroba.

Entonces yo voy a decir igual a y te decía que yo iba a hacer un ejemplo aquí de una expresión explícita.

Pues yo voy a decir arroba paréntesis y aquí dentro yo puedo colocar una expresión de Sharp.

En este caso sería una expresión Landau que básicamente me permite colocar una función anónima para

así yo decir que lo que sea que yo coloque aquí en esta parte que te estoy nombrando es lo que va a

ocurrir cuando el usuario presione el botón.

Por ejemplo, yo voy a decir con Sol right line para escribir en la consola Right line y yo simplemente

voy a colocar me han linkeado muy bien, entonces aquí yo estoy definiendo de nuevo una expresión lambda

en la cual yo estoy colocando una función anónima, lo cual me permite indicar que yo quiero que ocurra

cuando el usuario clique este botón y lo que va a ocurrir es que se va a ejecutar esto que tenemos acá.

Entonces déjame salvar, vamos a volver a Google Chrome y vamos a ver qué tenemos.

Presionamos, yo voy a presionar F12.

Antes de eso, déjame colocar esto así y vamos a ver que tenemos aquí la consola.

Yo le voy a dar impresióname y vamos a ver que tenemos.

Me han aplicado.

Es decir, cuando yo utilizo con storyline vemos que se coloca el texto en la consola.

No importa cuántas veces yo escriba, se va a repetir el mismo texto.

Tú no estás obligado a utilizar una expresión.

Landau Acá tú, pues igual que hicimos con Transformar, tú puedes definir una función.

Por ejemplo, yo voy a decir aquí Void porque no va a retornar nada a la función manejar click, botón

y ya.

Yo tengo esta función aquí que va a ser lo mismo que tenemos acá.

Déjame pegar este código acá me ha clickeando desde una función.

Voy a escribir para que se diferencie, punto y coma por acá.

Y ahora, en vez de yo utilizar una expresión explícita, yo puedo utilizar una expresión implícita

simplemente colocando el nombre de la función que yo quiero que se ejecute.

Entonces, ya con esto yo estoy diciendo que cuando el usuario haga clic en el botón Presionar, se

va a invocar la función manejar clic botón que va a escribir en la consola clickeando desde una función

vamos a salvar, vamos a venir para acá y vamos a ver qué podemos hacer.

Clic en Presionamos y de nuevo tenemos main clickeando desde una función.

14. Utilizando Clases

Algo cheveres que podemos hacer.

Como en todas las aplicaciones de Sharp, es que podemos definir clases para centralizar la lógica.

Entonces yo lo que voy a hacer es que yo voy a decir que yo quiero centralizar esta función de transformar

en una clase, porque quizás yo la quiero utilizar en distintos lugares.

Entonces yo lo que voy a hacer es que voy a venir al servicio en Explorer y aquí en el Proyecto Client

yo voy a hacer clic derecho porque quiero crear una carpeta, le vamos a llamar Utilidades y dentro

de utilidades yo voy a crear una nueva clase.

Le voy a llamar Utilidades String y dentro de acá yo me voy a llevar esto y lo voy a colocar acá, pero

la voy a hacer pública y también la voy a hacer estática para que se pueda utilizar sin necesitar una

instancia de la clase Utilidades String.

Entonces, ahora yo lo que puedo hacer es que puedo venir acá y vamos a ver que tenemos un error.

Claro, tenemos un error porque ya no tenemos acceso a la función transformar directamente desde acá,

sino que tenemos que utilizar utilidades String.

Entonces para eso yo voy a hacer aquí utilidades string punto.

Pero vamos a ver que tenemos un error.

Tenemos un error porque necesitamos utilizar la directiva y using para traer esta clase que tenemos

aquí que está en un namespace distinto.

Entonces yo puedo hacer el control punto acá y darle a Enter y vamos a ver cómo se coloca la directiva

using acá.

Y ahora, si yo tengo acceso a utilidades string y estoy utilizando la función transformar, y si yo

salvo, si yo salvo todo, vamos a ver que todo sigue funcionando en Google Chrome y venimos acá y vamos

a cerrar esto, vamos a ver que seguimos teniendo Guillermo en mayúscula.

Ahora yo quiero hacer algo y es que vamos a decir que realmente yo voy a utilizar esto en muchísimos

lugares, yo voy a utilizar este namespace muchísimos componentes de mi aplicación para ya no tener

que estar siempre repitiendo esto.

Yo puedo cortar acá y puedo venir al desarrollo en Explorer y puedo venir hacia Import Razor, que es

básicamente como un archivo general donde se colocan las directivas y using por defecto que van a aplicar

a los distintos componentes de mi aplicación.

Entonces aquí yo puedo colocar Leisure, películas, client, utilidades, salvamos.

Y ahora vamos a ver cómo aquí nuevamente esto sigue funcionando y sigue funcionando porque al utilizar

esta directiva using en este archivo de imports tenemos acceso a todos los tipos que se encuentran en

este namespace desde cualquier componente de mi aplicación.

Una clase que yo quiero que creemos para nuestro proyecto es la clase película.

Como hemos dicho, el nombre de nuestro proyecto es Razor Películas.

Por qué?

Porque vamos a realizar una aplicación que tiene que ver con películas.

Así que yo quiero que creemos una clase de película donde vamos a poder almacenar el título de una película,

la fecha de estreno, el trailer, entre otras cosas.

Entonces, dónde vamos a colocar esa clase de película?

Bueno, como la clase película es básicamente como una pieza central de nuestra aplicación, tenemos

que pensar muy bien dónde la vamos a colocar.

Eso me lleva a tener que explicar para qué sirven esos tres proyectos que se encuentran dentro de nuestra

solución.

Tenemos la solución blazer, películas y dentro tenemos tres proyectos el Proyecto Client, el Proyecto

Server y el Proyecto Start.

El Proyecto Client básicamente representa lo que el usuario va a ver con lo que el usuario va a interactuar.

Es decir, el proyecto Client es el que se va a descargar en el navegador del usuario y es el código

que se va a ejecutar en su navegador en el navegador del usuario.

Entonces, aquí yo voy a colocar, por ejemplo, cualquier cosa que tenga que ver con la capa de presentación

de mi proyecto Botones.

Esta función de transformar, que simplemente toma un string y lo vuelve mayúscula, pues no hay ningún

problema en colocarlo en el proyecto Client, porque ese es un código sencillo, por decirlo de algún

modo, que no me molesta que mi usuario lo vea.

No es un código sensible, es un código que básicamente trabaja con la parte de la presentación.

Ahora el proyecto Server es el que representa mi web API.

Aquí yo voy a colocar por ejemplo mi lógica para acceder a la base de datos.

Eso no es algo que yo quiero que se descargue en el navegador del usuario y por eso yo lo coloco en

el proyecto server o lo vamos a colocar en el futuro en el proyecto server, es decir, lo que se coloque

en el proyecto server no se va a descargar en el navegador del usuario, así que este es un lugar ideal

para colocar aquel código que tú quieres tener para ti en tu servidor y que el usuario no tenga forma

de acceder a este de manera directa o que no pueda visualizarlo en su navegador.

Finalmente, como tenemos dos proyectos, cliente y servidor, van a haber ocasiones en las cuales vamos

a querer compartir código entre ambos proyectos y para eso sirve este tercer proyecto que tenemos aquí,

que es el proyecto Surt, que significa compartido como nombre.

Lo indica es para compartir código tanto entre cliente y server.

Entonces, como la clase película para nosotros es tan fundamental en nuestra aplicación, la vamos

a colocar en Smart para así poder utilizarla tanto en client como en server.

Algunas personas no hacen esto, algunas personas toman esa clase de película y mejor la colocan en

server solamente y colocan una copia en client para que cada uno tenga una versión distinta y cada clase

pueda evolucionar de manera independiente una de la otra.

Ambas son opciones válidas.

En nuestro caso, vamos a utilizar la más sencilla, que es colocarla en el proyecto Start.

Entonces yo voy a venir a Search, entonces yo lo que voy a hacer es que voy a crear una carpeta acá,

agregar carpeta, le vamos a llamar entidades, aquí vamos a colocar las entidades de nuestro proyecto.

En el futuro hablaremos más de que es una entidad, pero básicamente una entidad es una clase que representa

una tabla de una base de datos que en nuestro caso va a ser SQL Server.

Entonces yo voy a hacer clic derecho, agregar clase película y yo necesito que la hagamos pública para

que podamos acceder a esta clase desde fuera de este proyecto.

Y yo por ahora le voy a colocar solamente dos propiedades que van a ser nombre o título.

Vamos a colocar el mejor título para el título de la película y también le vamos a colocar un DateTime

de fecha, lanzamiento, fecha, lanzamiento de la película.

Fíjate que tenemos aquí una línea verde.

Eso es porque tenemos los tipos de referencia no nulos en nuestro proyecto.

Entonces yo lo que voy a hacer es que voy a colocar aquí igual null, signo de exclamación.

Esto me sirve para perdonar el nulo para indicar que no hay ningún problema con que tengamos esto sin

inicializar en un valor que no sea nulo.

Otra opción hubiese sido, por cierto, otra opción hubiese sido que yo colocara aquí un no label y

aquí yo estoy diciendo que no hay ningún problema, que podemos colocar nulo en título y no hay ningún

problema con eso.

Pero si yo no quiero eso, si yo no quiero que el título pueda ser nulo, pues yo simplemente le perdono

el nulo para que no tengamos así ninguna advertencia.

Entonces ahora yo lo que quiero que hagamos es que mostremos una película aquí en nuestro componente

de index.

Para eso yo lo primero que voy a hacer es que voy a venir para acá, voy a tomar este namespace y lo

voy a llevar hacia mi componente de import, porque como te decía, yo voy a utilizar bastante eso de

entidades, así que yo lo voy a colocar aquí para mayor comodidad.

El punto y coma no es obligatorio, así que yo puedo eliminarlo.

Luego vamos a venir a Index y yo lo que voy a hacer es que voy a crear un pequeño método acá.

Por el momento vamos a codear a nuestra película en el futuro.

Eso va a venir de una base de datos, pero vamos a ir paso por paso.

Yo voy a decir aquí película, obtener película y yo voy a decir return new película y yo voy a colocar

título Wakanda Forever y vamos a colocar fecha de lanzamiento New DateTime.

Vamos a colocar 11 de noviembre del 2022.

Muy bien, por acá.

Y ahora yo voy a colocar aquí una variable película Wakanda forever.

Get return.

Obtener película.

Muy bien, ahora yo voy a venir acá arriba y voy a colocar un div.

Ya.

Yo puedo eliminar esto.

De hecho, vamos a colocar un div y vamos a decir h3 película y vamos a decir párrafo, título arroba,

Wakanda forever punto, título.

Y vamos a clonar esa línea.

Y vamos a decir fecha lanzamiento y vamos a decir por acá, fecha lanzamiento.

Y vamos a colocar aquí tu string y le vamos a poner este formato.

Hemos cometido un error con la fecha.

Yo coloqué esto mal.

Esto debió ser 2022 mes 11 día 11 es año, mes y día.

Muy bien, vamos a salvar y vamos a ver que tenemos y vamos a ver que tenemos película, el título de

la película y la fecha de lanzamiento.

Solamente para que destaque vamos a colocar el título y la fecha en negrita, así que vamos a venir

acá, vamos a venir acá.

Recordamos que podemos colocar código de HTML alrededor de nuestras expresiones de Sharp sin ningún

problema.

Vamos a hacer lo mismo por acá y vamos a colocar esto por aquí atrás.

Salvamos, volvemos y así está mejor.

15. Bucles

Ya vimos que podemos mostrar una película en pantalla.

Ahora vamos a intentar mostrar varias películas en pantalla.

Vamos a ver que para eso vamos a utilizar un bucle o un bucle.

Básicamente me permite iterar sobre una colección de valores.

Entonces, lo que vamos a hacer es que vamos a crear un nuevo método, el cual me va a permitir retornar

una colección de películas.

Entonces yo lo que voy a hacer es que va a venir para acá y voy a decir listado de película.

No tiene que ser un Estado poder hacer un arreglo, pero yo voy a trabajar con el listado, obtener

películas en plural y yo voy a decir retorno list, película.

Y vamos a decir yo voy a pegar tres películas y déjame colocar un punto y coma por acá.

Y por cierto, que yo tengo un error, debió ser forever.

Ahora que yo tengo este listado de películas, yo lo voy a utilizar.

Para eso yo voy a venir para acá.

Déjame dejar esto aquí por un momento y yo voy a utilizar un foreach.

Primero vamos a utilizar un foreach y un for.

Ambos me sirven para iterar, pero cada uno tiene sus particularidades.

Por ejemplo, vamos a decir foreach, yo voy a decir foreach y yo voy a decir película en obtener películas.

Y aquí adentro yo tengo acceso a la variable película a través de la cual yo puedo colocar los datos

de la película.

Entonces yo lo que voy a hacer es que yo me voy a llevar todo esto.

Sin embargo, este if lo voy a colocar aquí arriba y lo voy a colocar aquí abajo para que así tengamos

películas por acá.

Pero cada película va a estar dentro de su propio if también.

Entonces un diez por aquí y vamos a colocar esto por acá.

Entonces vamos a tener un div que va a contener todas las películas, un título de películas y luego

un dif para cada película.

Entonces por cada película yo voy a repetir este código de HTML, solo que en vez de yo decir Wakanda

forever, yo voy a decir película para acceder a esta variable película que yo tengo acá.

Y ya con eso yo estoy repitiendo este código tres veces, una por cada película que tengamos acá.

Y ahora déjame limpiar esto un poco y vamos a salvar y vamos a volver para Google Crown y vamos a ver

que tenemos.

Vamos a ver que tenemos películas y tenemos Wakanda Forever.

Tenemos Moana y tenemos Inception.

Ahora vamos a probar lo mismo pero con un for.

Algo que yo no puedo hacer con el foreach es, por ejemplo, colocar uno, dos y tres, es decir, colocarle

un número de secuencia a cada película con un for.

Esto es bastante sencillo.

Vamos a ver, déjame colocar aquí, déjame colocar aquí cuatro ejemplo for each solamente para separarlo

de lo que va a ser el ejemplo for ejemplo for.

Entonces yo lo que voy a hacer para mi comodidad es que yo voy a crear aquí una pequeña propiedad list

película solamente para que no tengamos que escribir tanto.

Yo voy a decir películas, le voy a decir Get return.

Obtener películas.

Maybe Entonces ahora vamos a venir acá arriba y vamos a utilizar un for arroba for tap tap y vamos a

decir aquí películas paint account.

Luego aquí adentro, déjame copiar esto y pegarlo acá.

Y vamos a ver que tenemos un error porque porque película era una construcción del for, es decir,

esta variable película la creaba el foreach.

Aquí yo estoy en un foro y no tengo acceso a una variable película.

Aquí.

Yo lo que puedo hacer entonces es en vez de utilizar película, yo voy a decir películas corchete y

voy a utilizar este índice para acceder a cada uno de los elementos de este listado de películas que

yo tengo acá.

Entonces yo voy a decir aquí y para, por ejemplo, acceder al primer elemento, luego al segundo elemento,

luego el tercer elemento, etc, vamos a hacer lo mismo con fecha de lanzamiento, vamos a decir esto

y vamos a ver que si yo salgo, por lo menos por ahora, vamos a tener todo bien.

Vamos a ver que efectivamente tenemos ejemplos foreach y ejemplo for.

Sin embargo, como te decía, lo chévere de utilizar el for es que yo tengo acceso a la variable de

índice.

Por tanto yo puedo venir acá y utilizar una expresión explícita de Say sharp y decir y más uno.

Y colocamos un paréntesis acá qué va a hacer esto?

Pues que me va a permitir colocar 123.

Así por ejemplo la película uno, la película dos la película tres.

Entonces vamos a salvar y vamos a ver que efectivamente tenemos eso.

Como podemos ver, aquí tenemos el uno, el dos y el tres.

Entonces, como podemos ver, tenemos el for que me permite tener más acceso a funcionalidades como

por ejemplo a través de esta variable de índice.

Sin embargo, el foreach, que es más simple.

Fíjate cómo me da incluso una variable que yo puedo utilizar en cada una de las iteraciones.

Si tú no necesitas acceder al índice, puedes utilizar el foreach.

Si tú necesitas acceder al índice, puedes utilizar el foro

16. Condicionales

En este vídeo vamos a hablar acerca de condicionales.

Los condicionales nos permiten ejecutar código según una condición.

Por ejemplo, vemos que tenemos varias películas acá.

Supongamos que yo quiero que si una película se estrenó hace poco tiempo, digamos hace menos de dos

semanas, pues yo quiero colocarle nuevo en el título para eso, entonces yo puedo utilizar un condicional.

Fíjate que tenemos la fecha de lanzamiento y yo lo que voy a hacer es que voy a comparar esa fecha de

lanzamiento con la fecha actual y si la diferencia en días es menor que 14, es decir, tiene menos

de dos semanas de haberse estrenado, pues le colocaremos nuevo aquí arriba.

Vamos a ver eso.

Vamos a venir acá y este ejemplo yo lo puedo hacer tanto con el for como con el for.

Yo lo voy a hacer con el for y sencillamente entonces yo lo que voy a hacer es que yo voy a venir acá

y voy a decir if datetime punto today, que es el día de hoy.

Le voy a sustraer la fecha de lanzamiento, punto, fecha, lanzamiento y voy a hacer punto días y vamos

a decir que tiene que ser menor o igual a 14.

Entonces, si la fecha actual, si le restamos la fecha de lanzamiento y hoy menos que 14 días de diferencia,

pues es nuevo.

Entonces aquí yo voy a colocar un span y le vamos a colocar un color style color red y vamos a decir

entre paréntesis nuevo.

Muy bien, entonces vamos a salvar y vamos a ver que si volvemos para acá, vamos a ver que efectivamente

tenemos nuevo colocado en esta película, porque yo estoy grabando este video un 14 de noviembre y por

tanto fue hace menos de dos semanas que se estrenó esta película y por lo tanto tenemos este mensaje

de nuevo por acá.

En cambio esta de Moana fue hace ya varios años, igualmente Inception y por eso no se les coloca ese

contenido de nuevo.

Pero fíjate como entonces utilizando un IF, utilizando un condicional, yo puedo expresar que si se

da una condición específica, entonces es que vamos a colocar una pieza de código, que en este caso

es este mensaje de nuevo que tenemos por acá.

Por otra parte, igual lo interesante sería que por ejemplo, coloquemos un condicional para indicar

que las películas no han cargado.

Cómo así?

Bueno, ahora mismo tenemos las películas.

En este obtener películas que tenemos acá, esto carga automáticamente, entramos en la aplicación y

es totalmente inmediato porque está en memoria.

Pero en la vida real esto va a venir de una base de datos que va a estar en un servidor distinto al

servidor de mi aplicación.

Así que esto se puede tomar unos segundos, se encargará.

Durante esos segundos yo quiero mostrarle un gif al usuario para indicarle que tiene que esperar.

Entonces vamos a simular eso.

Obviamente no tenemos una base de datos por ahora, pero podemos simular un tiempo de espera para verificar

que nuestra aplicación funciona correctamente.

Ante la situación de que las películas no han cargado para poder hacer este ejemplo, yo voy a tener

que utilizar un concepto que no hemos visto, que es un evento de ciclo de vida de un componente.

Básicamente, el evento que vamos a utilizar es el de inicializado, que básicamente me permite ejecutar

una función al momento de inicializar un componente.

Entonces yo lo que voy a hacer es que voy a venir para acá, voy a decir protected override y voy a

decir inicializar async y lo voy a colocar asíncrono.

Y lo que voy a hacer es que voy a quitar esto de acá y voy a decir Ah, what task delay.

Tres segundos.

Esto básicamente me permite forzar una espera de tres segundos antes de que se ejecute una función así

yo voy a venir acá y voy a decir películas, obtener películas.

Y ahora yo lo que voy a hacer es que yo voy a quitar esto de aquí y voy a colocar y voy a colocar get

y set de esta manera.

Al principio esto no va a tener ningún valor.

Déjame colocar incluso este no la bola aquí para indicar que esto puede ser nulo y no hay ningún problema

con eso, porque luego de tres segundos es que se va a cargar el listado de películas de nuevo.

Esto simulando la espera que nos toma buscar la data de una base de datos.

Ahora yo lo que quiero hacer es venir para acá.

Yo lo que quiero hacer es venir para acá y indicar con un condicional que va a pasar en caso de que

película sea igual a nulo.

Es decir, yo voy a decir if películas y null.

Si es nulo, entonces yo quiero colocar algún tipo de mensaje de espera como párrafo cargando las películas

else en el else.

Yo voy a colocar esto, es decir, para que solamente se.

Intente hacer el foreach de películas.

Que por cierto, esto debería ser películas.

Yo debería estar utilizando aquí películas muy bien.

Solamente en este caso es que vamos a utilizar las películas cuando veamos que películas no sea nulo.

Si es nulo, mostramos este mensaje.

Si no es nulo, entonces venimos acá y reiteramos el listado de películas.

Déjame comentar esto Control case para comentar para que no tengamos ningún problema con que esto sea

nulo y esto me de un error.

Entonces vamos a salvar, venimos para acá y vamos a esperar a que cargue esto de nuevo y vamos a ver

que tenemos cargando las películas.

Y luego de tres segundos fíjate cómo se coloca en las películas.

Te decía que yo quería colocar un gif, así que déjame ir a Google.

Vamos a buscar loading gif, vamos a buscar cualquiera loading gif que se vea más o menos bien.

Este creo que está bien.

Podemos decir copy image address y ahora vamos a venir para acá y en vez de esto que está un poquito

seco, yo voy a decir Image search esto y vamos a colocar esto así para que se visualice un poquito

mejor, para que se pueda visualizar todo en pantalla.

Cerramos y podemos colocar un alt cargando.

Muy bien, salvamos, venimos para acá de nuevo y vamos a ver que ahora tenemos un Loring gif.

Luego de tres segundos se muestran las películas.

Otra cosa que podemos hacer es decir else if para indicar una segunda condición.

Por ejemplo, qué pasa si las películas es un listado vacío?

Es decir, el listado de películas está vacío?

Entonces yo voy a decir el if películas punto count es igual a cero.

En ese caso, yo voy a decir párrafo No hay películas para mostrar.

Y ahora yo voy a venir acá, voy a comentar esto.

Déjame incluso decir en vez de esto yo lo voy a comentar.

Sí, lo voy a comentar y voy a decir películas igual a un arreglo vacío de películas.

Vamos a colocar esto así salvamos y ahora vamos a venir por acá y vamos a ver que luego de que cargue

tenemos el Loring gif por tres segundos y luego de que termine tenemos.

No hay películas para mostrar porque estamos colocando un listado vacío de películas.

Sin embargo, podemos quitar esto, podemos volver a colocar esto, vamos a bajar esto a un segundo

para no tener que esperar tanto y vamos a ver que todo funciona.

Y podemos continuar.

17. MarkupString

En ocasiones, ni siquiera el dinamismo de los condicionales es suficiente y vamos a querer renderizar

html, el cual viene de una variable o de una base de datos.

Para realizar esto podemos utilizar Markov String.

Sin embargo, es importante destacar que utilizar Markov Stream conlleva su riesgo, pues es posible

que seamos víctimas de código malicioso al exponernos a la vulnerabilidad de ejecutar código arbitrario

en nuestras aplicaciones, por lo que es importante proceder con cautela.

Vamos a decir que yo quiero cada título de la película formateado de una manera distinta.

Por ejemplo, puede ser que uno sea en negrita, otros en cursiva, otros sin ningún tipo de estilo

especial, etc Entonces lo que vamos a hacer es que vamos a venir acá y yo voy a quitar de aquí de título

esto de Bold, esta V de Bold o negrita para que no se coloque en negrita el título, por lo menos no

por defecto.

Y vamos a ver que efectivamente ahora los títulos no van a tener negrita.

Muy bien.

Ahora yo lo que voy a hacer es que yo voy a colocar eso de negrita o cursiva aquí en el nombre, por

ejemplo Wakanda forever.

Si va a venir en negrita.

Luego Moana va a venir en itálicas o cursiva e Inception lo vamos a dejar así como está.

Sin embargo, si yo salvo, vamos a ver que aquí lo que tenemos es el HTML como tal, Por qué?

Por una medida de seguridad de Razor?

Porque como te decía hace un momento, puede ser una vulnerabilidad estar ejecutando código arbitrario

en tu aplicación.

Por tanto, tenemos que indicarle a Roy Short que queremos realizar una excepción a eso que queremos.

De todos modos, evaluar el código HTML para que se le muestre evaluado al usuario.

Para eso, como te decía, vamos a utilizar Markup String.

Entonces yo lo que voy a hacer es que aquí detrás de título, yo voy a castear esto a Marco String,

cierro aquí y cierro acá y ya.

Eso es todo.

Ya con eso yo puedo salvar y yo puedo volver hacia Google Chrome.

Y vamos a ver que efectivamente tenemos Wakanda Forever.

En negrita tenemos Moana en cursiva e Inception sin ningún tipo de estilo en particular.

Principio del formulario

18. Resumen

Final del formulario

En este módulo vimos que la sintaxis de razor nos permite escribir código de HTML y si en un mismo archivo

esto es útil para tener en un mismo lugar la parte visual y lógica de una sección de la aplicación.

A esta sección le llamamos componente.

Dentro de un archivo de Ray podemos declarar variables métodos y utilizar estos miembros en nuestro

código.

Tenemos la posibilidad de utilizar bucles como el foreach y el for dentro de código de reusar.

Esto es útil para iterar colecciones y mostrar información en pantalla por cada elemento de la colección.

También podemos utilizar condicionales como if.

Esto es útil para ejecutar piezas de código según una condición establecida.

Podemos utilizar Markov String para renderizar dinámicamente código de HTML en archivos de Razor.