Lo que voy a mostrar, si quieren ustedes no lo hagan todavía, porque esto va a consumir mucho recurso.

Voy a recargar el navegador web una vez más, voy a colocar en película servis, voy a colocar un consolador

que diga llamando esto no lo haga llamando, solo es una demostración llamando, llamando a Piei.

Grabo los cambios, voy a regresar.

La primera vez está bien que lo haga.

Trajo las películas, las mostró.

Eso está bien.

Pero cuando llego abajo aquí hizo cuatro peticiones http ven rápido.

Hizo eso.

Voy a bajar rápidamente.

Y aquí volvió a ser otras cuatro.

Voy a bajar.

Y volvió.

Aquí son nueve.

Voy a bajar rápido.

Voy a bajar rápido.

Aquí son seis más.

Entonces, estos son inconvenientes.

No se preocupen por estas imágenes.

Ya lo vamos a reemplazar.

Pero el problema que tenemos aquí es que se están disparando múltiples peticiones en el momento en que

sólo deberíamos de traer una teoría.

Aquí consumimos cuatro veces más de lo que ocupábamos.

Cuatro veces más, nueve veces más.

Y puede que se dispare muchas más veces dependiendo de la conexión Internet.

Cómo resolvemos esto?

Bueno, hay varias formas.

Uno sería saber cuando yo estoy cargando información y para eso podríamos quedarnos aquí.

Una propiedad la voy a poner public porque quiero acceder a ella.

También podríamos ser Garris y Seres.

Pero la verdad, eso lo voy a manejar si pública se va a llamar cargando.

Es un Vulcano que por defecto está en fols.

Ok.

En el momento en que yo empiezo a cargar, por ejemplo, yo estoy en este punto, en este preciso instante.

Voy a poner a cargar la información.

Entonces sería disputo cargando, cargando.

Va a ser igual a Truc?

Ok, entonces descargando va a ser igual a true.

Y dependiendo de esto yo voy a decidir si quiero mostrar que se yo, las hace.

Bueno, a permitir la carga o bloquearla.

Ahora, en el momento en que ya tengo una respuesta y yo incremento el Cathedral a Apache, aquí puedo

decir disputo cargando.

Va a ser igual a Fox porque ya terminé de cargar.

Y si se volviera a disparar, entonces tengo que dejarlo pasar.

Hay varias formas de trabajar con esto.

Esto es lo que voy a hacer.

Es una que todavía le falta un optimizacion que ya vamos a ver.

Tranquilos, déjenme terminar la clase antes de que me empiecen a linchar.

Ok, entonces vamos a regresar a el.

Aquí yo puedo preguntar si estoy cargando, por ejemplo y si el despuntó película servis punto cargando.

Si eso está en true, entonces voy a retornar o ser el ratón para salirme de una vez y que no haga nada

más.

Pero si esto estuviera en Fools, si esto estuviera en False, entonces si va a poder ejecutar este

código, voy a grabar los cambios.

Y antes de continuar, dejemos aquí el coso lo que diga cargando epie.

Traemos los cambios.

Voy a regresar a la aplicación.

Esto sí, la primera vez se hace.

Voy a empezar a bajar.

Ups!

Hice un zoom a la pantalla, empezó a bajar, empezó a bajar cargando y.

Y lo hice una vez.

Voy a bajar rápido.

Hay Luiso dos veces, voy a bajar rápido y lo hizo dos veces.

Y tal vez esto no se mira muy bien, pero ya vamos a arreglarlo.

Por lo menos estamos disparando máximo dos aquí sólo una, pero estoy bajando rápido.

Ok, entonces también podríamos darle tal vez un Dibam Stein, o sea, un espacio de tiempo a que la

información se renderiza en pantalla, porque también eso puede ser un inconveniente, pero ya redujimos

drásticamente los seis y las nueve llamadas que hacíamos.

Otra cosa que tengo que hacer para asegurarme de que efectivamente esto no se va a llamar varias veces,

es preguntar voy a preguntar acá, aquí arriba.

Puedo preguntar si el disputo cargando está en true, si está cargando igual a otro cargando películas,

entonces aquí no voy a ser ni siquiera la validación HTTP, o sea, ni siquiera la petición voy a hacerla.

Entonces, cómo puedo retornar algo si yo pongo aquí un return, punto y coma graven los cambios y eso

en teoría va a funcionar, pero no de la manera como nosotros queremos.

Bueno, si va a funcionar, pero no es del todo el resultado perfecto o no el resultado óptimo.

Yo empiezo a hacer scroll, yo seré scroll y pues en teoría funciona bien, bien, en teoría funciona

bien.

Estoy haciendo tantas peticiones.

Esos errores ya lo vamos a trabajar porque hay unas imágenes que no tiene poster y entonces por eso

se marca un error.

Ya vamos a quedarnos un pi para resolver eso.

El inconveniente que tengo es que esto realmente no está retornando un cartelera responses aquí cuando

se llame y bueno, se llame.

Y esto va a retornar.

Un null no es un observable y eso no es óptimo si regresa un oli%.

En fin, eso si quieren lo pueden dejar así.

Pero la verdad quiero mejorar este código un poco y es totalmente opcional.

Si ustedes no lo quieren hacer, no lo hagan así lo pueden dejar.

Pero la verdad yo lo que quiero hacer si se fijan aquí en el home component, aquí estamos usando los

resultados o respuesta a punto result 3.o results Realmente no estamos usando todo lo que este objeto

respuesta tiene.

Por qué?

La respuesta tiene varias cosas en resultados total de páginas o tal resultados y dates.

Realmente muchas cosas hay quien no estoy usando.

Entonces, para qué voy a retornar un objeto si siempre estoy extrayendo información que viene dentro

de él?

No sería más fácil de queeste que el cartel era realmente sólo regrese las películas, que es lo único

que me interesa.

Entonces podemos hacer esto mucho mejor.

Ok, este va a ser un obsesivo que va a regresar.

Un muy top es un arreglo de munis.

Asegúrense de importar muy también de las interfaces que teníamos ahí.

Ok, de mi cartelera response.

Es una un listado de películas con arreglo de películas.

Esto me va a marcar un error porque literalmente todo lo que tengo acá no retorna.

Eso está retornas.

Nunca telera response.

Entonces, si yo lo quiero transformar.

Puedo pasar esto?

Respuesta por el map tab paréntesis coma y el map también importarlo de los operadores de RX.

El MAP ya lo hemos utilizado antes.

Sirve para transformar una respuesta o una emisión de un observable.

Entonces aquí tendríamos la respuesta que la respuesta sigue de cartelera responde.

De hecho, si quieren dejar el cursor encima, lo van a ver.

Bueno, debería decir ahí que tipo es.

Déjeme terminar la función.

Entonces dejen el cursor encima.

Y ahora si he de Cathedral Responses.

Pero lo único que me interesa es la respuesta.

Punto.

Los resorts note que lo acepta perfectamente porque es del mismo tipo y ya se quitó el error.

Ahora estoy regresando únicamente un listado de películas.

Si ustedes graban los cambios, esto va a dejar de funcionar y me dice que no puede leer la propiedad.

Length de On define.

Estas son ciertas cosas que cambiaron.

Ok, ahora ya no regresamos un objeto.

Estamos regresando literalmente el arreglo.

Regresemos al Jom.

Note que me está marcando un error.

Realmente yo ya no necesito hacer nada de esto.

Aquí van a ser las movies directamente.

Puedo hacer la desestructuración directamente de las películas.

Voy a bajar.

Aquí tengo lo mismo.

Aquí voy a ponerle maids.

Y voy a cambiar esto por Moulins, por Meis y el código queda mucho más fácil.

Voy a grabar los cambios, voy a regresar.

La aplicación sigue funcionando genial.

Por último.

Por último, lo que acabamos de hacer no resuelve el problema que teníamos aquí.

Por qué?

Porque igual ya no estoy regresando un observable que emite móviles.

No está regresando, está regresando!

Undefined.

Entonces más de uno va a decir Bueno, simplemente pongámosle un arreglo.

El problema es que esto ahora retorna un arreglo en lugar de un observable que emite un arreglo.

Uy, Dios mío, Fernando, esto suena complicadísimo, pero la verdad es que no. En RX existe otro.

Bueno, una función para crear observables llamada off tap.

Paréntesis.

Y dentro de los paréntesis van a poner lo que ustedes quieran que emita como un observable y éste off

ustedes lo van a importar de rx js directamente de este paquete.

Y aunque ustedes aquí le dicen que está obsoleto algo, esto de que R.T.

No es todo lo que está obsoleto, sólo son ciertas firmas.

Si ustedes ven en la documentación, aquí también vamos a poder encontrar que hay muchas cosas que están

obsoletas, pero era la manera de trabajar antes.

Pero realmente el off sigue es bueno, sigue funcionando y sigue existiendo, y ese no es el que está

obsoleto.

Lo que están obsoletos son ciertas variantes del off ok.

Porque eventualmente ya no van a existir, pero ésta que estamos utilizando sí funciona.

Ok, entonces cualquier cosa me avisan.

En fin, en pocas palabras, lo que hace el off es transformar un observable que emite lo que sea que

está aquí adentro.

Es un observable que emite un arreglo vacío que perfectamente es válido para una película.

Voy a grabar los cambios y esta es una mejor solución, porque si estoy cargando, entonces regreso

un arreglo vacío y ya se acabó el asunto.

No hace nada más.

Entonces hemos hecho bastantes optimizaciones a nuestro código, a nuestro infinite scroll, para evitar

que eso dispare cientos de peticiones cuando estamos llegando al final, en la próxima clase.

Nos tenemos que enfocar ahora en arreglar estas imágenes que están rotas y la verdad es que lo hemos

dejado para el final.

Bueno, para final de esta parte miremos nuestro componente.

Nuestro película posee Greet, el HTML y nosotros estamos colocándolo de esta manera que no es muy bonita.

Funciona, pero no es muy bonita.

Entonces nosotros tendríamos que evaluar si este poster path viene y si viene entonces mostrarlo y si

no viene, colocar una imagen por defecto que ya se les facilite al inicio de la sección que era este

no?

Y esto es lo que vamos a mostrar cuando la imagen o el póster no exista, pero eso lo vemos en los próximos

videos.