En la clase pasada nosotros nos quedamos en el punto donde creamos simplemente el servicio y bueno eso

era todo lo que hicimos realmente.

Ahora toca configurarlo para que nos podamos conectar a Sibö sigamos los pasos bajamos un poco y ahora

hay que abrir ese archivo 6.3 en el que nosotros tenemos acá al que creamos en otra dirección pero es

el mismo archivo.

Entonces ahora nos piden que copiemos todo este código.

Por favor copien el suyo porque hay configuraciones de su Heydi su Kleine ahí el cual van a necesitar

para poder conectarse a este dades en particular.

Bueno no el mío obviamente el de ustedes.

Bueno si ustedes copiaron este código se conectarían al mío pero porque estamos haciendo la configuración

y a mí no me gusta ocultar todo este código porque a veces nos podemos perder.

Entonces usen el suyo toquen este botón para copiar todo el código.

Empezamos acá seleccionen todo el código que tenemos acá con controlã o Cumana o simplemente borran

todo y lo pegan ok vamos a subir un poco.

Aquí hay un pequeño error de este outcome fue que no lo estábamos usando y no se usa en este archivo

o al menos no por los momentos.

Lo voy a comentar y grabar los cambios.

El archivo no debería de tener ningún error.

Acuérdense que estos paquetes ya los instalamos así que debería de funcionar sin ningún problema.

Básicamente lo que nos sirve o lo que nos está haciendo acá este archivo es crearnos métodos y observables

para detectar cuando la persona se conecta cuando se desconecta y cuando hace todo lo demás.

Si nosotros presionamos en Visual Studio control p o cómo Anspe seguido de una arroba nosotros vamos

a poder ver todos los métodos y obviamente también sus propiedades.

Aquí tenemos el outline y ese signo de dólar usualmente significa que es un observable es como una sintaxis

que no es obligatoria no es obligatorio poner el signo de dólar pero les va a decir a ustedes que eso

es un observable y nos podemos suscribir a él.

Luego tenemos el auténtico hired que básicamente se explica por sí solo y aquí tenemos uno que es una

propiedad y también tenemos el Yussef profanó que también es otro observable y esto es lo que nosotros

realmente estamos creando con todo este código no hace falta que ustedes lo comprendan al 100 por ciento

esto es puro JavaScript pero en su versión de tags entonces eso es todo lo que nosotros vamos a ocupate

el servicio clave los cambios y los cierran.

Ok aquí voy a pedirles que regresemos acá descarguemos el navegador web y aquí no me pareció ningún

error.

Entonces también les voy a pedir que bajemos la consola con control y volvamos a levantar la aplicación

con el XR.

Nuevamente lo que yo quiero ver es si tenemos algún error con esa librería por qué.

Porque a mí cuando yo estuve haciendo este ejercicio aquí yo tenía un error que me decía algo como que

la promesa o las promesas ya habían sido reemplazadas en el son.

JS Estoy esperando todavía puede ser que me dé el error pero si ustedes lo tuvieran en este preciso

instante entonces la solución que yo les voy a dar por momentos mientras la gente de óxido actualiza

su librería es que vayamos al paquete Jayson bajamos un poco y acá en el archivo han sido.

J.S en lugar de usar esta versión o la última versión.

Yo probé con la versión 1.0 punto cero esa es la que me funcionó sin ningún inconveniente y seguíamos

el Código tal cual está en la documentación.

Entonces nada más para que tengan esos presente.

Ojalá me dé el error para que lo podamos apreciar bien.

Sigamos viendo la documentación ya hicimos todo este código y ahora tenemos en nuestro servicio y ahora

básicamente ya trabajar con él si no estoy mal.

Aquí tendremos para obtener el usuario hacer login manejar la autenticación y cómo vamos a hacer eso.

Ok ahora nos toca ser restaurar el orden del Estado de Longhi cuando la aplicación recarga básicamente

lo que nos está pidiendo es que el app component tenemos que establecer simplemente este método hacer

la inyección de nuestro servicio y realizar el local.

Eso es lo que nos están pidiendo que hagamos en este instante.

Entonces vamos a ver algunos a nuestro component punto 13.

Hay que hacer la inyección del servicio o extraer el cajero porque no me sirve.

Entonces vamos a El constructor arrolladas prives de tipo de tipo Service TAP.

Asegúrese que lo importe y grabamos los cambios puede ser que esto me dispare ya el error.

Vamos a ver y no me disparadora parece que ya no está aquí ya lo arreglaron la gente de Chilo.

Genial sigamos bajando ya tenemos eso que restaurar el Lovin vamos a ver qué más hay aquí nos están

diciendo para manejarlo en el mapa.

Nosotros ya tenemos un bar simplemete y vamos a tener que usarlo.

Noten que aquí el servicio está público y van a decir por qué aquí está privado y aquí está público

porque el hecho de que sea público me va a permitir a mí utilizarlo en el html del componente.

Si fuera sólo Priv en el componente en este nivel yo no tendría acceso a propiedades apuntó.

Yo no tengo esas propiedades pero si yo lo pusiera como público acá Public gravó los cambios en este

archivo regreso entonces en teoría yo debería ser capaz de ver acá las propiedades de alto.

Igual no aparecen acá pero esa sería la diferencia tal vez el mismo estudio no me lo reconoce pero esa

es la principal diferencia entre que sea público y privado en la inyección de un servicio en el constructor

de un componente.

Ok voy a cerrar esto y hagamos esta implementación.

Ups.

Esta implementación del naba hay que hacer esto y hacer esta conexión para poder observar a los cambios.

Con eso vamos a terminar este video pero todavía quedan cosas por hacer.

Vámonos a nuestro Nachbar hagamos el HTML lo que primero tenemos que hacer es colocar dos botones ok

para poder mostrarlo cuando estamos bloqueados y cuando estamos sin autenticación Boron cerramos botón

con la clase suponerle MTN MTN oponerle outlet Primary outline perdón a Hotline Miami que estoy acá

en The Massé ingresar a copiarme este botón.

Luego no va a ser problema y este va a ser Ringer para que sea rojo.

Y aquí va a decir salir.

Veremos los cambios.

Deberíamos de ver ambos botones.

Perfecto y ahora utilicemos para poder mostrar cuando estamos indicados y cuando no estamos auténticas

abramos el par componen y tendremos que inyectar nuestro servicio pero de manera pública para poderlo

utilizar en el HTML de mi componente de tipo Service TAP.

Asegúrese de importarlo.

Ok.

Los cambios realizados al Nachbar y ahora es simplemente colocar un gif pero ese gif va a tener una

condición interesante y Tab y viene la expresión vamos a ver que es la expresión que nos recomiendan

es así.

Punto.

Lockett y de esta manera voy a saber si está bloqueado o no está bloqueado.

En el signo de admiración que significa la negación de esta variable que debería ser Yuliana.

Entonces voy a pegarla y éste le voy a quitar.

Si todo sale bien en teoría cuando yo pongo un punto debería ver esas propiedades.

No sé por qué no me están apareciendo ahí pero me está pareciendo todo lo que tiene lo que tengo.

Snippets que tengo relacionados a él o que veamos los cambios aquí tenemos un error.

Vamos a ver esto ya se debería de quitar en cualquier momento que ya se quitó.

Vamos a ver si es cierto y el botón está ingresado.

Obviamente si yo lo topo no hace nada pero eso lo vamos a manejar en el siguiente video por los momentos

ya hemos creado nuestro servicio y ya tenemos la conexión y implícitamente ya tenemos la parte para

detectar cuando está bueno cual hay que mostrar el botón de ingresar y cuando hay que mostrar el botón

de salir todavía no hacemos ningún cambio estos botones no hacen absolutamente nada pero eso es lo que

haremos en el siguiente video.