En esa lección vamos a hablar sobre los decoradores de clases.

Esto es solo para que ustedes no le tengan miedo cuando miren esta sintaxis en angular porque es en

lo personal es algo que cuando yo lo miré la primera vez dije Dios santo qué diablos es lo que estoy

viendo.

O Qué será esto.

Entonces el objetivo de esta lección es que ustedes cuando lo vean no se vayan a asustar y sepan qué

es lo que se está haciendo porque estamos en el mismo ejercicio anterior de la clase pasada por eso

les dije que no hiciéramos un raspado ni nada.

Voy a abrir acá MPM estar para ver los cambios en tiempo ejecución y todo esto que voy a usar las herramientas

de desarrollo y tenemos el mensaje nada del otro mundo así como lo dejamos en el video pasado.

Qué es eso de los decoradores.

Voy a cerrar aquí con una X.

Regresemos al examen clase .3 aquí en la clase voy a quedarme una simple función llamado imprimir consola

ojales yo aquí recibiría un argumento que será constructor.

Así lo voy a poner constructor o le podemos poner const o el nombre que ustedes quieran darle.

Voy a ponerle constructor clase dos puntos y esto va a ser de tipo fungieron.

Yo sé que será algo extraño todo esto que acabo de hacer.

Pero miremos.

La idea es que voy a recibir el constructor de la función.

En teoría la firma de la clase.

Pero vamos a ver eso es una función.

Lo único que voy a hacer es imprimir en consola el constructor clase ok lúdicamente es imprimirlo en

consola eso es todo bien es una simple función.

Eso es lo importante es una simple función.

Qué es lo que hace es que recibe una función.

Eso es todo porque sonará como raro pero eso es todo lo que está haciendo ahora qué es eso del decorador

un decorador es algo que se le va a colocar en este caso y hacer ser un decorador de clase se le va

a colocar antes de la definición de la clase.

De esta manera con la arroba y vamos a poner imprimir consola sin ejecutarlo así nada más voy a grabar

los cambios y voy a tener un error acá dejo el cursor encima y me dice que los decoradores son una característica

potencialmente que puede cambiar en un futuro pero ya están aceptados pero siempre lo está marcando

como este error.

Entonces tengo que hacer una modificación para decirle que utilice lo que son los decoradores.

Para eso abrimos aquí el archivo aquí es config apuntó Jayson.

Hay que colocar una coma después de esta característica aquí no vamos a perder una coma y remuevan el

comentario con controles Layo como las de experimental de cruceiros iban a ponerlo en Tru graban los

cambios en el archivo Etty Escogió el punto Jayson y lo cierran automáticamente acá debería de reconocerles

ya removerle el error y si no lo hace tienen que cerrar el estudio y volverlo abrir en el peor de los

casos pero usualmente tarda unos cinco segundos en que lo reconozca y listo.

Ok eso es todo lo que yo acabo de hacer.

Van a anotar voy a quitar este mensaje de imprimir simplemente estoy usando la clase grabar los cambios.

Voy a regresar al navegador web y posiblemente aquí tengamos un error.

Vamos a ver entonces lo que vamos a hacer acá como hicimos una modificación en el archivo de ti es lo

que va a haber que hacer es cancelarlo con control cancelar nuestra ejecución del programa y nuevamente

MPM Stadt para que tome los nuevos cambios del archivo de config y no tengamos ese error tan espantoso

que teníamos allá abajo de esa manera toma los nuevos cambios.

Qué es lo que acaba de pasar.

Aquí tenemos la impresión de consola de toda la definición de mi clase.

Todos lo tenemos acá si revisamos nuevamente nuestro examen Toklas eso lo que hace es imprimir consola

un decorador nos permite a nosotros a la hora de definir la clase poderle añadir o ejecutar cosas a

la misma.

No es algo que ustedes vayan a hacer en Angola pero lo único que es importante es que comprendan que

un decorador simplemente nos va a servir para añadirle funcionalidades a nuestra clase.

Ustedes van a notar más adelante que aquí van a ver que esto es un componente que es un servicio y otras

cosas.

Al tener decoradores específicos en cada una de las clases que nosotros vamos a hacer en angular que

eso todo lo va a hacer automáticamente por nosotros al hacer esas importaciones angular le dice que

esta clase va a ser un servicio que esta clase va a ser un componente que esta clase va a hacer que

yo un atributo o cualquier otra cosa.

No se preocupen si ustedes no saben de qué diablos estoy hablando pero lo único que me interesa es que

con ustedes miren este decorador.

Lo único que simboliza es que esta clase va a tener todas las funcionalidades que yo le agregué mediante

imprimir consola en este caso no vamos a hacer en el cruce ahora no vamos a ser decoradores porque no

es necesario hacer esas cosas a ese nivel pero esto lo utiliza angular para transformar mis clases en

otros objetos dependiendo de las necesidades.

Eso es básicamente un decorador.

Es una función que se ejecuta en las clases o en cualquier otro lado que expande la funcionalidad y

más de uno me va a decir Ah bueno pero yo he escuchado que teníamos los extensos.

Qué diferencia hay lo que pasa es que los extensos a diferencia de los decoradores los extensos expanden

la clase.

Esto lo que hace es añadirlo a un nivel superior.

A qué me refiero con eso con solo definirlo de esta manera la clase va a recibir todas esas funcionalidades

que yo quiera sin necesidad de poner la palabra extent sin necesidad de complicar más mi código aquí

no se preocupe mucho si este concepto no les queda del todo claro lo vamos a ver más adelante en el

curso porque aquí van a ver que el decorador se reciben objetos o algún argumento eso lo van a ver cuando

ya estemos trabajando con angula pero lo único que me interesa es que ustedes recuerden y hago hincapié

porque esto fue lo que más me costó aprender en angular un decorador simplemente es una función que

expande la clase añadiéndole características y funcionalidades propias pero nosotros no vamos a crear

nuestros decoradores simplemente vamos a usar todos los que ya creó por nosotros bueno lo vamos a dejar