



Qué son y como usar...

SIGNALS ANGULAR 16



ESERCIECODE



A ANGULAR 16

SIGNALS

Angular 16 introduce signals, una forma nueva de gestionar cambios de estado en aplicaciones. Las signals son funciones que pueden actualizarse y depender de otras, creando un gráfico reactivo que se actualiza automáticamente. Con observables RxJS, las signals permiten flujos de datos declarativos y potentes.

En comparación con el mecanismo tradicional de detección de cambios en Angular, las signals permiten la detección de cambios sólo en los componentes afectados, mejorando el rendimiento y reduciendo la complejidad. Las señales también hacen que el código sea más fácil de entender y permiten un mayor control sobre los cambios de valores.



A ANGULAR 16 SIGNALS



```
const x = signal(5);
const y = signal(3);
const z = computed(() => x() + y());
console.log(z()); // 8
```

asigna el valor inicial de x en 5

z será un valor computado, no se puede reasignar,
quedará escuchando siempre el cálculo

las señales se llaman con () paréntesis
porque actúan como una función que
devuelve el valor de la señal

se utiliza set para darle nuevo valor a x

```
x.set(10);
console.log(z()) // 13
```

z sigue escuchando los cambios,
entonces al llamarla luego del cambio
de X, cambia su valor



A ANGULAR 16 SIGNALS



```
const contador = signal<number>(0);  
contador.update(valor => valor + 1);
```

acá se setea el tipo de dato
update se usa para modificar el valor de
una señal desde el valor anterior

```
console.log('El valor de contador es: ' + contador()); //2
```

toma el valor nuevo y se llama a ese
valor con los paréntesis



A ANGULAR 16 SIGNALS



```
const usuarios = signal([ { nombre: 'Sergie Code', tecnologia: 'Angular' } ]);  
usuarios.mutate(usuario => {  
  usuario[0].tecnologia = 'Angular con Signals';  
});  
console.log(usuarios()[0])
```

un array de objetos (con un objeto)

mutate se usa para modificar sólo un valor puntual (o varios) del interior de una señal

solo se modifica el valor de la clave tecnología en la posición 0 del array

los paréntesis van ANTES que los corchetes porque la señal de vuelve el array



A ANGULAR 16 SIGNALS



```
const contador = signal(0)  
effect(() => {  
  console.log(`El valor del contador es: ${contador()}`);  
});
```

como contador está dentro de este effect lo va
a estar escuchando y cada vez que contador
cambie se ejecutará nuevamente effect

se inicia contador en 0

Se usa effect para escuchar cualquier cambio
que se haga sobre los signals que estén dentro
y ejecutar acciones inmediatas al cambio

En este caso hará un console.log()



A ANGULAR 16 SIGNALS



```
const usuarioActual = signal({nombre: 'Sergie Code'})
```



funcion de limpieza se manda
por parámetro

```
effect((funcionLimpieza) => {  
  const usuario = usuarioActual();
```

```
  const temporizador = setTimeout(() => {  
    console.log(`Hace un segundo el usuario fue ${usuario.nombre}`);  
  }, 1000);
```

```
  funcionLimpieza(() => {  
    clearTimeout(temporizador);  
  });  
});
```



Los efectos pueden iniciar operaciones de
larga duración, que deben cancelarse si el
efecto se destruye o vuelve a ejecutarse antes
de que finalice la primera operación

