

DWC

(Desarrollo Web en entorno cliente)



Angular

Práctica 05

Comunicación entre componentes

Índice

1.- Objetivos.....	1
2.- El proyecto.....	1
3.- Funcionamiento.....	1
3.1.- Componente trabajadores.....	2
3.2.- Componente trabajador.....	2

1.- Objetivos

En esta práctica vamos a trabajar con la comunicación entre componentes mediante el modelo padre-hijo.

Usaremos una comunicación bidireccional entre ambos.

2.- El proyecto

Se pide diseñar un proyecto en Angular para poder mostrar la lista de trabajadores de la empresa y poder dar o quitar likes a cada uno de ellos.

Los datos de cada trabajador se deberán definir del tipo Trabajador, para ello habrá que crear una interface con el nombre Trabajador en una carpeta llamada Modelos

Las propiedades de la interface Trabajador serán:

- id y votos de tipo numérico
- nombre, cargo y foto de tipo string

También habrá que crear un array llamado TRABAJADORES de la clase Trabajador, el array también se creará como la interface Trabajador en la carpeta Modelos

Se proporciona:

- Un array con los datos de los trabajadores
- Las fotos de cada trabajador, con el nombre del id del trabajador y la extensión jpg
- Las imágenes para los iconos del like y el unlike.
- Los ficheros html de los componentes trabajadores y trabajador

Todas las imágenes se deberán almacenar en la carpeta assets.

3.- Funcionamiento

La aplicación estará formada por dos componentes, el componente principal trabajadores y el componente trabajador.

El componente trabajador será hijo del componente trabajadores y se pasarán datos entre ellos.

3.1.- Componente trabajadores

El componente trabajador accederá a los datos de los trabajadores y por cada uno de ellos llamará al componente trabajador.

Tendrá una variable titulo con el valor “Listado de Trabajadores”, esta variable se interpolará en la parte superior de la página.



Se deberán implementar los métodos:

- **sumaVoto:** Esta función se ejecutará cuando se produzca el evento **likeTrabajador** emitido desde el componente trabajador que le pasará al padre el id del trabajador. Esta función lo que hará es incrementar la propiedad votos de ese trabajador en 1
- **restaVoto:** Esta función se ejecutará cuando se produzca el evento **unlikeTrabajador** emitido desde el componente trabajador que le pasará al padre el id del trabajador. Esta función lo que hará es decrementar la propiedad votos de ese trabajador en 1. Si el número de votos es cero no se podrá restar más y se mostrará un mensaje mediante un alert.
- **borrar:** Esta función se ejecutará cuando se produzca el evento **borrarTrabajador** emitido desde el componente trabajador que le pasará al padre el id del trabajador. Esta función lo que hará es eliminar ese trabajador

3.2.- Componente trabajador

Recibirá los datos de cada trabajador desde el componente trabajadores mediante el decorador @Input. Se mostrará cada trabajador en una estructura de card incluyendo:

- Los datos de cada trabajador
- La imagen like.jpg para darle votos a ese trabajador. Al hacer click sobre esa imagen se ejecutará la función like pasándole como parámetro el trabajador
- La imagen unlike.jpg para quitarle votos a ese trabajador.
- Un botón para borrar el trabajador



Emitirá los eventos `likeTrabajador`, `unlikeTrabajador` y `borrarTrabajador` al componente `trabajadores` mediante el decorador `@Output`

Se deberán implementar los métodos:

- **like:** es la función que se ejecutará al hacer click sobre la imagen de `like.jpg` y lo que hará es emitir el evento `likeTrabajador` con el id del trabajador
- **unlike:** es la función que se ejecutará al hacer click sobre la imagen de `unlike.jpg` y lo que hará es emitir el evento `unlikeTrabajador` con el id del trabajador
- **borrar:** es la función que se ejecutará al hacer click sobre el botón borrar y lo que hará es emitir el evento `borraTrabajador` con el id del trabajador