

Bootcamp FullStack Part-Time

Módulo 3. Práctica 3. Uso de componentes

Requerimientos:

- Atom instalado (o un IDE alternativo)
- Sourcetree instalado
- Cuenta en Github
- Repositorio privado en Github para el proyecto, con la versión de la práctica anterior subida
- Acceso al repositorio para los profesores. **Nombre del repositorio: memory-game**
- Node.js instalado

Objetivos:

1. Practicar lógica y sintaxis de Typescript
2. Uso de Input, Output y directivas de Angular
3. Otros conceptos como
 - a. Delegación de responsabilidades
 - b. Documentación de código

Descripción de la práctica

Vamos a recrear el juego memory-game. Se debe completar el código que se empezó a ver durante la sesión del fin de semana. El juego consta de dos componentes esenciales: **board** y **card**. Podemos tener componentes extras, como un contador (si lo hacéis, lo debéis implementar o bien con ng-content o bien con ngTemplateOutlet)

NO SE PUEDE USAR BOOTSTRAP

Board

Va a contener la lógica de la partida. Para ello vamos a tener lo siguiente:

- Un array de cards para la partida
- Un contador para conocer el número de turnos.
- Un método para iniciar la partida
- Un método que va a comprobar si las cartas seleccionadas tienen el mismo nombre.

Card

Será cada uno de los elementos de la partida. Tendrá:

- Un Input que define la card
- Un Output para enviar la info necesaria al componente Board para que compruebe si se acierta o no.
- Deberá mostrar css acorde a diferentes estados:
 - sin clicar
 - clickeada

- resuelta

Al terminar debemos tener:

General

En general, debemos **tener un comentario en cada elemento de la clase** indicando cuál es la función de esa propiedad o bien qué hace el método.

Solamente podemos tener **máximo dos cards a la vista a la vez**

Board

board.component.html:

- Usar al menos una directiva estructural, como ngFor
- Usar al menos una directiva de atributo

board.component.ts:

- importar el tipo card
- al menos 2 propiedades (de momento tenemos **names** y **cards**)
- método para inicializar la partida
- método para comprobar si las cards seleccionadas son correctas
- método auxiliar shuffle

Card

card.component.html:

- Al menos un evento
- Al menos dos directivas de atributo (puede ser [hidden], [ngStyle], [ngClass])

card.component.ts:

- No se puede hacer nada más. **No se puede tocar**. Solamente podéis expandir (añadir propiedades) si lo consideráis necesario el **type card**

Bonus

- Poder determinar desde board el máximo número de rondas que podemos hacer. Se deberá mostrar un "Game Over"
 - Verde si terminamos por debajo del número de rondas
 - Rojo si terminamos por encima del número de rondas
 - Botón de resolve para que de un solo click resuelva el juego
- Si se hace el número de rondas se puede hacer un código de colores de forma que:
 - Si intentos/total < 50% => verde
 - Si 80% > intentos/total > 50% => naranja
 - Si intentos/total > 80% => rojo
- Tener un componente para hacer un reset, pero que pregunte antes si se está seguro antes de ejecutarse (con un confirm)

Recursos:

- Documentación oficial w3schools:
 - HTML: <https://www.w3schools.com/html/default.asp>
 - CSS: <https://www.w3schools.com/css/default.asp>
 - Javascript: <https://www.w3schools.com/js/default.asp>
- Typescript: <https://www.typescriptlang.org/>
- Angular: <https://angular.io/>