

Ejercicios 4

En esta sección de ejercicios vamos a continuar trabajando con la aplicación iVideoclub. En este caso vamos a completar la arquitectura de Angular para transmitir los datos desde los controladores a las plantillas.

Ejercicio 4.1 - Página de autor y currículum

En el controlador de la página de autor declaramos la variable "autor" de tipo objeto con las siguientes propiedades:

```
1  autor: {  
2    name: string,  
3    email: string,  
4    twitter: string,  
5    phone: string  
6  };
```

La inicializamos en el constructor con nuestros datos (`this.autor = {name: 'Javi', ...};`) y la utilizamos para rellenar los campos correspondientes de la plantilla `.html` utilizando el binding por interpolación (por ejemplo: `{{ autor.name }}`).

En la subpágina con el currículum vamos a hacer lo mismo. En primer lugar declaramos la variable `listaTrabajos` con el siguiente formato:

```
1  listaTrabajos: {fecha: string, desc: string}[];
```

Y la rellenamos con los datos en el constructor (`this.listaTrabajos = [{fecha: '2012-2015', ...}, ...]`). A continuación procedemos a modificar la plantilla para que utilice estos datos. Para esto vamos a hacer uso de la directiva `*ngFor` sobre la etiqueta `"ion-item"` para que cree un elemento de la lista por cada uno de los elementos del array `listaTrabajos`. Además vamos a aplicar la directiva `*ngIf` sobre la etiqueta de apertura de la lista (`ion-list`) para que únicamente se cree si el array contiene algún elemento:

```
1  <ion-list *ngIf="listaTrabajos">  
2    <ion-item *ngFor="let trabajo of listaTrabajos">  
3      <ion-label class="ion-text-wrap">  
4        <!-- TODO: Asignar los datos -->
```

```

5     </ion-label>
6   </ion-item>
7 </ion-list>

```

La clase "ion-text-wrap" indica que se parta la cadena de texto en caso de que ocupe más que el ancho de pantalla. Si no lo ponemos el texto se saldría por el margen derecho y no se podría leer. En la sección "CSS Utilities" de la documentación podemos encontrar más etiquetas similares para ajustar la disposición de texto y contenidos.

Ejercicio 4.2 - Página de videoclub

En este ejercicio vamos a terminar la página con el listado de películas. Para esto haremos uso de los servicios y del paso de parámetros entre pantallas.

En primer lugar crea un servicio llamado "películas". Define la variable "películas" dentro del nuevo servicio para que sea un array de objetos con todos los campos que vamos utilizar:

```

1 private películas: {
2   title: string,
3   year: string,
4   director: string,
5   poster: string,
6   synopsis: string} [];

```

Inicializa esta variable en el constructor con todos los datos de las películas. Una vez hecho esto añade un método (llamado `getPelículas`) que devuelva directamente la lista de películas completa (`return this.películas`).

Ahora vamos a modificar la página "videoclub" para que haga uso del servicio que acabamos de definir. En primer lugar añade el `import` para que cargue esta clase:

```

1 import { PeliculasService } from '../services/peliculas.service';

```

Añádalo también a los parámetros del constructor para que se inyecte como dependencia. Una vez hecho esto ya podemos usar el servicio dentro de esta clase. En este caso vamos a utilizarlo dentro del constructor para llamar a su método "getPelículas" y guardar el resultado en la variable "listaPelículas" (la cual la puedes declarar del tipo "listaPelículas: any[]").

Modifica ahora la vista asociada a esta página para que utilice la variable "listaPelículas" rellena con los datos del servicio. Para esto podemos utilizar también las directivas `*ngFor` y `*ngIf` como vimos para la página de autor:

```
1 <ion-list *ngIf="listaPelículas">
2   <ion-item *ngFor="let pelicula of listaPelículas; index as id"
3   (click)="verPaginaDetalle(id)" detail>
4     ...
5   </ion-item>
</ion-list>
```

Fíjate en el código anterior como se ha añadido `"index as id"` dentro del bucle para guardar el índice de la iteración (es decir, el índice de la película que se muestra en cada fila de la lista). Este índice se utiliza en la llamada al método `verPaginaDetalle` del evento `click` para posteriormente poder recuperar la película sobre la que se ha pulsado.

La directiva `detail` utilizada en la etiqueta `<ion-item>` añade una flecha en la parte derecha para indicar visualmente que es un elemento pulsable.

Ejercicio 4.3 - Página detalle

Por último vamos a terminar la página `detalle`. Para esto, en primer lugar vamos a modificar el servicio `PelículasService` para añadir un método que devuelva una película a partir de su identificador:

```
1 getPelicula(id): any {
2   return this.peliculas[id];
3 }
```

Modifica a continuación el código del controlador de la página `detalle` para importar este servicio y para que se inyecte como dependencia en el controlador.

Una vez hecho esto ya podemos utilizar el nuevo método `getPelicula(id)` del servicio para recuperar la película a partir del `id` recibido como parámetro. Guarda los datos de la película en una nueva variable `pelicula` de tipo `any` y utiliza esta variable para rellenar los campos del template asociado.

Recuerda cambiar también el título de la barra de navegación para que indique el título de la película que se esté visualizando.