

Comentarios a la solución de esta PEC:

Ejercicio 1:

- A medida que iba implementando formularios en angular material, me fui dando cuenta que el control de errores es muy similar en todos los controles así que decidí implementar un errorHandler común para todos ellos. Me basé para esto en <https://gist.github.com/klement97/58ab606031c4fe4a2a04b6ed8c2e9d92>

Ejercicio 2:

- Para maquetar la tabla he utilizado tablas de angular material (mat-table"
- Para los iconos de editar y borrar de los elementos de las tablas añadido font-awesome mediante

npm i font-awesome

y lo importo globalmente en styles.scss con `../node_modules/font-awesome/scss/font-awesome.scss`

Ejercicio 3:

- Para poner el spinner he declarado una variable Observable de tipo State en cada componente y he activado/desactivado el spinner dependiendo de la variable "loading" de los estados:

```
postState: Observable<PostsState>;  
...  
this.postState = store.select('posts');
```

Y en el HTML

```
<div *ngIf="postState | async as p">  
  <div class="overlay" *ngIf="p.loading">  
    <div class="center">  
      <mat-progress-spinner diameter=50 mode="indeterminate"  
color="accent">  
    </mat-progress-spinner>  
  </div>  
</div>
```

Ejercicio 5

Añado a cada "card" una imagen aleatoria desde la web de imágenes de la teoría.

Añado animación a todas las cards para que aparezcan haciendo un "fadeIn".

Ejercicio 6

Utilizo ngxChars para representar el gráfico de barras con los likes/dislikes.