

Angular: "Puppynstagram"

Integrantes:

- Francisco Olivares
- Gabriel Valenzuela

Primero un repaso sobre angular

- Creación de componentes
- Instanciación de componentes
- Creación de store

Componentes (image-box.component.ts)

```
import { Component, OnInit, Input } from '@angular/core';

@Component({
  selector: 'app-image-box',
  templateUrl: './image-box.component.html',
  styleUrls: ['./image-box.component.scss']
})

export class ImageBoxComponent implements OnInit {

  @Input() dog: Puppy;

  constructor() { }

  ngOnInit() {
  }

}
```

Instanciación de componentes

El componente Home ocupa a la componente image-box

```
<div class="columns is-centered">
  <div class="column is-10 is-7-desktop is-6-widescreen">
    <app-image-box [dog]="dog" *ngFor="let dog of puppies$ | async | slice:-10 ">

      </app-image-box>
    </div>
  </div>
</div>
```

Enlace de componentes (app.module.ts)

```
@NgModule({  
  declarations: [  
    AppComponent,  
    ImageBoxComponent,  
    HomeComponent,  
    ActionBarComponent,  
    NavbarComponent,  
    LikedDogsComponent  
  ], ...  
})
```

Creación de la store (ngxs)

Como mínimo se necesitan estas 3 cosas

- Modelos
- Acciones
- Estado

Store: Modelos

```
export interface Puppy{  
  url: string;  
  likes: number;  
  liked: boolean;  
  comments: Array<string>;  
}
```


Store: Acciones

```
import { Puppy } from '../models/puppy.model';

export class AddPuppy{
    static readonly type = '[PUPPY] Add'

    constructor (public payload: Puppy[]){}
}

export class LikePuppy{
    static readonly type = '[PUPPY] Like'

    constructor (public payload: {puppy:Puppy, like:boolean}){}
}

export class CommentPuppy{
    static readonly type = '[PUPPY] Comment'

    constructor (public payload: {puppy:Puppy, comment:string}){}
}
```

Store: Estado

```
export class PuppyStateModel{
  puppies: Puppy[]
}

@State<PuppyStateModel>({
  name: 'puppies',
  defaults: {
    puppies: []
  }
})

export class PuppyState{
  @Action(AddPuppy)
  add({getState, patchState}: StateContext<PuppyStateModel>, {payload}:AddPuppy ){
    patchState({
      puppies: [...getState().puppies, ...payload]
    })
  }
  ...
}
```

Componentes

Home (Template)

```
<div class="columns is-centered">
  <div class="column is-10 is-7-desktop is-6-widescreen">
    <app-image-box [dog]="dog" *ngFor="let dog of puppies$ | async | slice:-10 ">

      </app-image-box>
    </div>
  </div>
```

Home (Component)

```
import { AddPuppy } from '../../../store/actions/puppy.actions';
import { Puppy } from '../../../store/models/puppy.model';
import { ApiService } from '../../../services/api/api.service';
import { Component, OnInit } from '@angular/core';
import { Store } from '@ngxs/store';
import { Observable } from 'rxjs'

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.scss']
})
export class HomeComponent implements OnInit {

  puppies$: Observable<Puppy>

  constructor(private api: ApiService, private store: Store) {
    this.puppies$ = this.store.select(state => state.puppies.puppies)
  }

  addPuppy(puppies){
    this.store.dispatch(new AddPuppy(puppies))
  }

  ...
}
```

Home (Component)

```
ngOnInit() {  
  let images: Puppy[] = [];  
  let likes: number;  
  this.api.loadImages().subscribe(x => {  
    console.log(x);  
    if (x['status'] === "success") {  
      x['message'].forEach(element => {  
        likes = Math.floor(Math.random() * (300 - 20 + 1)) + 20;  
        images.push({url:element, likes:likes, comments:[], liked:false});  
      });  
    }  
    this.addPuppy(images)  
  })  
}
```

Service api

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Injectable({
  providedIn: 'root'
})

export class ApiService {

  constructor(private http: HttpClient) { }

  loadImages() {
    console.log("Loading data...");
    return this.http.get("https://dog.ceo/api/breeds/image/random/10");
  }
}
```

State add action

```
@Action(AddPuppy)
add({getState, patchState}: StateContext<PuppyStateModel>, {payload}:AddPuppy ){
    patchState({
        puppies: [...getState().puppies, ...payload]
    })
}
```


image-box (Component)

```
import { Puppy } from './../../store/models/puppy.model';
import { Component, OnInit, Input } from '@angular/core';

@Component({
  selector: 'app-image-box',
  templateUrl: './image-box.component.html',
  styleUrls: ['./image-box.component.scss']
})

export class ImageBoxComponent implements OnInit {

  @Input() dog: Puppy;

  constructor() { }

  ngOnInit() {
  }

}
```

action-bar (Template)

```
<div class="card-footer">
  <div class="card-footer-item">
    <button (click)="like(dog)" class="button is-fullwidth">
      <span [ngClass]="{'is-liked': dog.liked}">
        Me gusta
      </span>
    </button>
  </div>
  <div class="card-footer-item">
    <button (click)="open_comment()" class="button is-fullwidth">
      Comentar
    </button>
  </div>
  <div class="card-footer-item">
    <button class="button is-fullwidth" disabled>
      Compartir
    </button>
  </div>
</div>
<div class="card-footer" *ngIf="comment">
  <div class="card-footer-item">
    <input type="text" placeholder="Ingresa tu comentario..." class="input is-primary" #comment>
    <button (click)="commentPuppy(dog, comment.value)" class="button is-success" >
      Enviar
    </button>
  </div>
</div>
```

action-bar (Component)

```
import { LikePuppy, CommentPuppy } from '../../../store/actions/puppy.actions';
import { Store } from '@ngxs/store';
import { Component, OnInit, Input } from '@angular/core';
import { Puppy } from 'src/store/models/puppy.model';

@Component({
  selector: 'app-action-bar',
  templateUrl: './action-bar.component.html',
  styleUrls: ['./action-bar.component.scss']
})
export class ActionBarComponent implements OnInit {

  @Input() dog: Puppy;
  comment: boolean = false;

  constructor(private store: Store) { }

  ngOnInit() {}
  ...
}
```

action-bar (Component)

```
like(dog){
  if (!this.dog.liked) {
    this.store.dispatch(new LikePuppy({puppy:dog, like:true}))
  }
  else{
    this.store.dispatch(new LikePuppy({puppy:dog, like:false}))
  }
}

open_comment(){
  this.comment = !this.comment
}

commentPuppy(dog, comment){
  this.store.dispatch(new CommentPuppy({puppy:dog, comment:comment}))
  this.open_comment()
}

}
```

State like action

```
@Action(LikePuppy)
like({getState, patchState}: StateContext<PuppyStateModel>, {payload}:LikePuppy ){
    let pos:number = getState().puppies.indexOf(payload.puppy)
    let new_state:Puppy[] = getState().puppies
    if (payload.like) {
        new_state[pos].likes += 1
    }
    else{
        new_state[pos].likes -= 1
    }
    new_state[pos].liked = !new_state[pos].liked
    patchState({
        puppies: [...new_state]
    })
}
```

State comment action

```
@Action(CommentPuppy)
comment({getState, patchState}: StateContext<PuppyStateModel>, {payload}:CommentPuppy ){
    let pos:number = getState().puppies.indexOf(payload.puppy)
    let new_state:Puppy[] = getState().puppies
    new_state[pos].comments.push(payload.comment)
    patchState({
        puppies: [...new_state]
    })
}
```

liked-dogs (Template)

```
<div class="columns is-centered">
  <div class="column is-10 is-7-desktop is-6-widescreen">
    <div *ngFor="let dog of puppies$ | async">
      <app-image-box [dog]="dog" *ngIf="dog.liked">

        </app-image-box>
      </div>
    </div>
  </div>
```


liked-dogs (Component)

```
import { Puppy } from '../../store/models/puppy.model';
import { Component, OnInit } from '@angular/core';
import { Store } from '@ngxs/store';
import { Observable } from 'rxjs'

@Component({
  selector: 'liked-dogs',
  templateUrl: './liked-dogs.component.html',
  styleUrls: ['./liked-dogs.component.scss']
})
export class LikedDogsComponent implements OnInit {

  puppies$: Observable<Puppy>

  constructor(private store: Store) {
    this.puppies$ = this.store.select(state => state.puppies.puppies)
  }
}
```

navbar (Template)

```
<nav class="navbar is-black" role="navigation" aria-label="main navigation">
  <div class="navbar-brand">
    <a class="navbar-item" href="/">
      <p>Pupynstagram 🐶</p>
    </a>
  </div>

  <div class="navbar-menu">
    <div class="navbar-start">
      <a class="navbar-item" routerLink="/">
        Descubre perros
      </a>

      <a class="navbar-item" routerLink="liked-dogs">
        Tus likes
      </a>
    </div>

  </div>
</nav>
```

Routing

```
import { HomeComponent } from '../components/home/home.component';
import { LikedDogsComponent } from 'src/components/liked-dogs/liked-dogs.component';
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'liked-dogs', component: LikedDogsComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```