

Angular

Grupo 4

Natalia Barra - Luis Chodiman - Mauricio Ortiz

Demo

Puntos principales

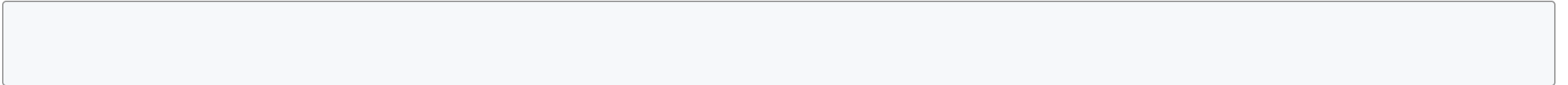
- Angular
 - Componentes
 - Angular vs Vue vs React
- NGXS
- API
- Testing
 - Unit Testing con Karma

Angular

Aspectos importantes utilizados en la aplicación

Components

Archivos `.vue` que contienen toda la estructura de un componente (HTML, JS, CSS)



Angular vs Vue vs React

Similitudes

Diferencias

- Curva de aprendizaje difícil en comparación
- *Convention over configuration*
- La mayoría de las utilidades vienen instaladas y configuradas
- El más pesado de los tres
- Uso de TypeScript

NGXS

- Quitar todo el *boilerplate* de NGRX

Principios a tener en cuenta

- Los mismos de Redux!
-

API

Versión gratuita de [OpenWeatherMap](#)

```
const weatherData = await fetch(  
  `http://api.openweathermap.org/data/2.5/weather?id=${id}&APPID=${  
    process.env.VUE_APP_APP_ID  
  }&units=${scale.param}`  
) .then(data => data.json());
```

Testing

Unit testing con Karma

Instalar dependencias: **Ya viene todo instalado!**

Configurar el cómo correr tests: `karma.conf.js`

Definiciones importantes:

- `Fixture` : El ambiente del componente
- `DebugElement` : El `html` del componente ya renderizado

Todo el *boilerplate* de los tests ya viene hecho!

Ejemplo: Las utilidades funcionan de igual manera a con Jest

```
describe("MyAwesomeComponent", () => {  
  beforeEach(() => {  
    // reproduce the test state  
  });  
  
  it("should be awesome", () => {  
    expect(component).toBe(awesome);  
  });  
  
  // More specs here  
});
```

Y ejecutar `ng test` o `ng e2e`

Ejemplo en el proyecto:

```
export class AlertButtonComponent implements OnInit {  
  content = "This is a notification to be tested!";  
  hideContent = true;  
  severity = 423;  
  
  constructor() {}  
  
  ngOnInit() {}  
  
  toggle() {  
    this.hideContent = !this.hideContent;  
  }  
  
  toggleAsync() {  
    timer(500).subscribe(() => {  
      this.toggle();  
    });  
  }  
}
```

Sus tests serían:

```
it("should create", () => {
  expect(component).toBeTruthy();
});

it("should have a message with 'tested'", () => {
  expect(component.content).toContain("tested");
});

it("should have an H1 tag of 'Alert Button'", () => {
  expect(de.query(By.css("h1")).nativeElement.innerText).toBe("Alert Button");
});

it("should toggle the message boolean", () => {
  expect(component.hideContent).toBeTruthy();
  component.toggle();
  expect(component.hideContent).toBeFalsy();
});
);
```

Y para testear cosas asíncronas: `fakeAsync()`

```
it("should toggle the message boolean asynchronously", fakeAsync(() => {  
  expect(component.hideContent).toBeTruthy();  
  component.toggleAsync();  
  tick(500);  
  expect(component.hideContent).toBeFalsy();  
}))
```


Conclusiones del testing

- Nos faltó probar la experiencia de e2e testing con Protractor o Cypress
- Tiene el ambiente de testing más completo y amigable gracias a la CLI
- Bastante documentación

Ranking final en testing:

1. Angular (por lejos)
2. React
3. Vue

Conclusiones finales

- Curva de aprendizaje difícil en comparación
- Requiere de más estudio para aprender las *conventions*
- Buenas herramientas para el *developer*: CLI, DevTools, etc
- Útil en un ambiente empresarial

Angular

Grupo 4

Natalia Barra - Luis Chodiman - Mauricio Ortiz

