



Curso de Angular

Unidad Didáctica 10: Testing



Ayuntamiento
de Vitoria-Gasteiz
Vitoria-Gasteizko
Udala

Índice de contenidos

- Introducción
- Pruebas de Unidad
- Pruebas de Aceptación
- Referencias
- Conclusiones



Introducción

Las pruebas se han convertido en una parte central al desarrollo en vez de una parte complementaria



Introducción

Para empezar deberemos diferenciar entre aquellos principales tipos de pruebas que debemos realizar en un proyecto



Introducción

El lanzamiento de las pruebas dentro del proyecto se realiza a través del comando:

ng test



Introducción

Este comando permite el lanzamiento de un navegador virtual que va probando uno a uno los tests escritos



Introducción

Podremos ver su funcionamiento a través de la url:

<http://0.0.0.0:9876/>

<http://cursosdedesarrollo.com/>



Introducción

Los dos frameworks principales de pruebas presentes en angular/cli son karma y jasmine



Introducción

Dentro del proyecto se buscarán todos aquellos
ficheros terminados en .spec.ts que deberán
disponer de las pruebas necesarias



Pruebas de Unidad

Este tipo de pruebas son esenciales a la hora de mantener la calidad del código generado



Pruebas de Unidad

Su enfoque principal es el de probar individualmente cada una de las clases que están presentes en el proyecto



Pruebas de Unidad

Vemos un ejemplo de Prueba de un Controlador



Pruebas de Unidad

//Importamos los módulos necesarios

```
import { TestBed, async } from '@angular/core/testing'; //
componentes de pruebas
```

//para que no falle el <router-outlet>

```
import {RouterTestingModule} from '@angular/router/testing';
```

// Componente a probar

```
import { AppComponent } from './app.component';
```



Pruebas de Unidad

```
describe('AppComponent', () => {  
  //beforeEach se ejecutará antes de cada prueba  
  
  beforeEach(async(() => {  
    TestBed.configureTestingModule({  
      //Declaramos el módulo a probar y sus dependencias  
  
      declarations: [  
        AppComponent  
      ],  
  
      //Incluida la dependencia con las rutas  
  
      imports: [ RouterTestingModule ]  
    }).compileComponents();  
  
  }));  
  
  ...  
  
});
```

<http://cursosdedesarrollo.com/>



Pruebas de Unidad

//Dentro de cada conjunto de pruebas metemos cada una de ellas

//it define una prueba

```
it('should create the app', async() => {
```

```
  //Creamos el objeto del componente
```

```
  const fixture = TestBed.createComponent(AppComponent);
```

```
  //cogemos el objeto de la app
```

```
  const app = fixture.debugElement.componentInstance;
```

```
  //expect comprueba que un valor es real (no null)
```

```
  expect(app).toBeTruthy();
```

```
});
```

<http://cursosdedesarrollo.com/>



Pruebas de Unidad

```
it(`should have as title 'app'`, async(() => {  
  
  const fixture =  
  TestBed.createComponent(AppComponent);  
  
  const app = fixture.debugElement.componentInstance;  
  
  /*Comprobamos que la variable title del componente tiene  
  un valor específico */  
  
  expect(app.title).toEqual('app');  
  
}));
```



Pruebas de Unidad

```
it('should render title in a h1 tag', async(() => {  
  const fixture = TestBed.createComponent(AppComponent);  
  fixture.detectChanges();  
  
  //Accedemos a la plantilla compilada  
  
  const compiled = fixture.debugElement.nativeElement;  
  
  //Comprobamos que hay un texto específico, ojo al querySelector  
  expect(compiled.querySelector('h1').textContent)  
  
    .toContain('Welcome to app!');  
  
}));
```

<http://cursosdedesarrollo.com/>



Pruebas de Aceptación

Las pruebas de aceptación tratan de gestionar un navegador consultando la aplicación, interactuando con ella y haciendo comprobaciones



Pruebas de Aceptación

Para realizar estas pruebas será necesario rellenar pruebas e2e (end2end) dentro del directorio e2e



Pruebas de Aceptación

Debemos comprobar que tenemos la biblioteca
ultron incluida en el packages.json

"ultron": "1.0.2",



Pruebas de Aceptación

Comprobamos el contenido de la carpeta e2e, veremos una carpeta src que tendrá los ficheros de las pruebas



Pruebas de Aceptación

Comprobamos el contenido de la carpeta e2e, veremos una carpeta src que tendrá los ficheros de las pruebas



Pruebas de Aceptación

Veremos dos tipos de ficheros: `po.ts` y `e2e-specs.ts`

los `po.ts` definen páginas (componentes principales a cargar)

los `e2e-specs.ts` definen la página a cargar y las pruebas a realizar



Pruebas de Aceptación

app.po.ts

```
import { browser, by, element } from 'protractor'; //importamos desde protractor

export class AppPage { //Definimos la página

    navigateTo() { //definimos la función que nos mueve a una ruta preestablecida

        return browser.get('/');

    }

    getParagraphText() { //definimos el acceso a un texto de un h1 de app-root

        return element(by.css('app-root h1')).getText();

    }

}
```

<http://cursosdedesarrollo.com/>



Pruebas de Aceptación

app.e2e-specs.ts

```
import { AppPage } from './app.po'; // Importamos la clase de la página

describe('workspace-project App', () => { //Describimos la prueba

  let page: AppPage;

  beforeEach(() => {

    page = new AppPage(); //Antes de cada prueba cargamos la página

  });

  it('should display welcome message', () => {

    page.navigateTo(); // navegamos a la ruta

    expect(page.getParagraphText()).toEqual('Welcome to app!'); //Comprobamos el valor

  });

});
```

<http://cursosdedesarrollo.com/>



Referencias

Testing en Angular:

<https://angular.io/guide/testing>

Testing de una app de pokemon:

<https://github.com/pepesan/angular6-pokedex-reborn>

<http://cursosdedesarrollo.com/>



Conclusiones



Datos de Contacto

<http://www.cursosdedesarrollo.com>
david@cursosdedesarrollo.com

<http://cursosdedesarrollo.com/>



Licencia



David Vaquero Santiago

Esta obra está bajo una
Licencia Creative Commons
Atribución-NoComercial-
CompartirIgual 4.0 Internacional

