2 - Pruebas de las vistas

 Pruebas de HTML y plantillas Angular

8 - Probando las vistas

Issue: Comportamiento página de proyectos

- componente
- fixture
- debug y native elements
- detección de cambios

S.U.T: ProjectsComponent

```
<header>
  <h2>
    Lista de proyectos
  </h2>
</header>
<main>
  <ng-template #noDataYet>
    <ab-no-data-yet></ab-no-data-yet>
  </ng-template>
</main>
```

Test: ProjectsComponent - spec

```
describe('GIVEN: the ProjectsComponent', () => {
  let fixture: ComponentFixture<ProjectsComponent>;
  let component: ProjectsComponent;
  let debugEl: DebugElement; let nativeEl: HTMLElement;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [SharedModule], declarations: [ProjectsComponent],
      providers: [
        {
          provide: ProjectsFacadeService,
          useValue: jasmine.createSpyObj('ProjectsFacadeService', {
            getProjects$: of([]).pipe(delay(100)),
            getTransactions$: of([]),
            getProjectViews: [],
          }),
        },
      ],
    }).compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(ProjectsComponent);
    component = fixture.componentInstance;
    debugEl = fixture.debugElement; nativeEl = fixture.nativeElement;
  });
});
```

```
it('WHEN starts THEN has a header', () => {
  // Consultando el elemento nativo
  const actual: HTMLElement = nativeEl.querySelector('header');
  expect(actual).toBeTruthy();
});
it('WHEN initializing THEN there is Esperando datos...', () => {
  // forzar la evaluación de datos para contenido dinámico
  fixture.detectChanges();
  // Consultando desde el wrapper angular
  const quoteDebug: DebugElement = debugEl.query(By.css('blockquote'));
  const quoteNative: HTMLElement = quoteDebug.nativeElement;
  const actual = quoteNative.textContent;
  const expected = 'Esperando datos...';
  expect(actual).toEqual(expected);
});
it('WHEN stable THEN there is no Esperando datos...', fakeAsync(() => {
  fixture.detectChanges();
  tick(100); // espera para que se resuelva el observable
  fixture.detectChanges(); // fuerza el recálculo de la vista
  const actual: DebugElement = debugEl.query(By.css('blockquote'));
  expect(actual).toBeFalsy();
}));
```

Ejercicio

> aplicar esto mismo a `home.component.spec.ts`

```
GIVEN: the HomeComponent
WHEN: is stable
THEN: there is no Esperando datos..
```

9 - Input y renders

Issue: Entrada de datos y efecto css en value objects

Probar el efecto de los datos en la presentación

Tratar la vista como una unidad distinta del controlador

S.U.T: Value Component

```
<dd [ngClass]="{'ok': isOk, 'ko': isOk===false  }"><b>{{ value  }}</b></dd>
```

Test: ValueComponent - spec

```
it('WHEN the input value is 42 THEN it renders a string 42', () => {
  component.value = String(42);
  fixture.detectChanges();
  const actualNative: HTMLElement = nativeEl.querySelector('b');
  const actual = actualNative.textContent;
  const expected = '42';
  expect(actual).toEqual(expected);
});
it('WHEN the input isOK is true THEN it renders with css class ok ', () => {
  component.isOk = true;
  fixture.detectChanges();
  const actualNative: HTMLElement = nativeEl.querySelector('dd');
  const actual = actualNative.classList.contains('ok');
  const expected = true;
  expect(actual).toEqual(expected);
});
```

Ejercicio

> Aplicar técnica a date-time

```
GIVEN: the DateTimeComponent
WHEN: the input date is the New York WTC crash,
THEN: it renders 11/09/2001
```

10 - Pruebas con dummies

Issue: Pruebas del timeAgo pipe usando un componente dummy

- cuando lo que queremos probar no tienen sentido por si mismo
- los pipes, muchas directivas
- los incluimos en una vista artificial

S.U.T: TimeAgoPipe

```
<p>{{ theDate | timeAgo }}</p>
```

Test: TimeAgoPipe - spec

```
@Component({
  template: `<p>{{ theDate | timeAgo }}</p>`,
})
class DummyComponent {
  theDate : Date;
}
...
  beforeEach(async () => {
    // Arrange
    await TestBed.configureTestingModule({
      declarations: [DummyComponent, TimeAgoPipe],
    }).compileComponents();
  });
   beforeEach(() => {
    // Arrange
    fixture = TestBed.createComponent(DummyComponent);
    ...
  });
```

```
it('WHEN the date is 01/01/2020 THEN renders hace mucho tiempo ', () => {
  // Act
  component.theDate = new Date(2020, 0, 1);
  // Assert
  const actualNative: HTMLElement = nativeEl.querySelector('p');
  const actual = actualNative.textContent;
  const expected = 'hace mucho tiempo';
  expect(actual).toEqual(expected);
});
```

11 - Pruebas de enrutado

Issue: Prueba del router con projectComponent

- usar `routerTestingModule`
- *no realiza la navegación real*

S.U.T: ProjectComponent

```
<header *ngIf="loaded">
    <h2>
      {{ project.title }}
    </h2>
    <p>{{ project.description }}</p>
...
</header>
```

```
ngOnInit(): void {
  this.projectSlug = this.service.getSlugFromRoute();
  this.loadData();
}
```

Test: ProjectComponent - spec

```
let router: Router;
let activatedRoute: ActivatedRoute;
beforeEach(async () => {
  // Arrange
  const routes: Routes = [{path:'projects/:id', component:ProjectComponent }];
  await TestBed.configureTestingModule({
    imports: [RouterTestingModule.withRoutes(routes), SharedModule],
    declarations: [ProjectComponent],
    schemas: [NO_ERRORS_SCHEMA],
    providers: [
      {
        provide: ProjectFacadeService,
        useValue: jasmine.createSpyObj('ProjectFacadeService', projectFacadeServiceMock),
      },
    ],
  }).compileComponents();
});
```

```
beforeEach(() => {
  // Arrange
  router = TestBed.inject(Router);
  activatedRoute = TestBed.inject(ActivatedRoute);
  fixture = TestBed.createComponent(ProjectComponent);
  component = fixture.componentInstance;
  debugEl = fixture.debugElement;
  nativeEl = fixture.nativeElement;
  router.initialNavigation();
  fixture.detectChanges();
});
```

```
it('WHEN The location is projects/1 THEN the url is well formed', fakeAsync(() => {
  // Act
  router.navigate(['/projects/1']).then(() => {
    const actual = router.url;
    // Assert
    const expected = '/projects/1';
    expect(actual).toEqual(expected);
  });
  tick();
}));
```

12 - Prueba de un formulario template driven

Issue: Alta de proyectos en NewProject con ngModel

- Mucha interacción con la template
- Dificultad de pruebas detalladas
- Los *model driven*, son más sencillos `+ts -html`

S.U.T: NewProjectComponent

```
<form #f="ngForm"
        (ngSubmit)="saveNewProject()">
  <label for="title">Nombre:</label>
  <input [(ngModel)]="newProject.title"
        required
        type="text"
        id="title"
        name="title"
        size="20">
  <button type="submit"
        [disabled]="!f.valid">Crear</button>
</form>
```

Test: NewProjectComponent - spec

```
  it('WHEN I fill the form THEN should send the values', async () => {
    await fixture.whenStable();
    fixture.detectChanges();
    // Act
    const titleDebug: DebugElement = debugEl.query(By.css('#title'));
    const titleInput: HTMLInputElement = titleDebug.nativeElement;
    titleInput.value = 'Testing my apps';
    titleInput.dispatchEvent(new Event('input'));
    fixture.detectChanges();
    const actual = component.newProject.title;
    // Assert
    const expected = 'Testing my apps';
    expect(actual).toEqual(expected);
  });
```

Extra

https://www.npmjs.com/package/ng-mocks

> **Repositorio:** angularbuilders/angular-budget/test_2_view
>
> > By Alberto Basalo