3 - Pruebas de principio a fin

Probarlo todo

Instalación

> En un proyecto aparte. Puede ser en el mismo repositorio

```
npm i cypress eslint eslint-plugin-cypress mocha mochawesome
```

Arranque

Con navegador

```
cypress open
```

Sin navegador

```
cypress run
```

```
cypress.json
```

13 - Pruebas básicas

- comandos de acción
- Comandos de consulta
- Comandos de comprobación
- Todo *pseudo* **Síncrono**

Basic - spec

```
describe('Visiting the url https://angularbuilders.github.io/angular-budget/', () => {
  it('should have _Cuadro de mando | Angular.Budget_ on its title', () => {
    cy.visit('https://angularbuilders.github.io/angular-budget/');
    cy.title().should('include', 'Cuadro de mando | Angular.Budget');
  });
});
```

14. Prueba organizada: Behavior Driven

- AAA
- GWT
- describe, context, it, expect?
- Features, scenarios, user stories...

BDD - spec

```
describe(`GIVEN: the url https://angularbuilders.github.io/angular-budget/`, () => {
  // Arrange
  const sutUrl = 'https://angularbuilders.github.io/angular-budget/';
  context(`WHEN: I visit it`, () => {
    // Act
    before(() => cy.visit(sutUrl));
    it(`THEN: should have _Cuadro de mando | Angular.Budget_ on its title`, () => {
      // Assert
      const expected = 'Cuadro de mando | Angular.Budget';
      cy.title().should('include', expected);
    });
  });
});
```

15. validate visible content

- usos creativos de actual y expected

Content - spec

```
describe(`GIVEN: A web app where i have 2 projects`, () => {
  // Arrange
  const sutUrl = 'https://angularbuilders.github.io/angular-budget/';
  context(`WHEN: I visit the home page`, () => {
    // Act
    before(() => cy.visit(sutUrl));
    it(`THEN: should have a section called Mis Proyectos`, () => {
      const expected = 'Mis Proyectos';
      // Assert
      const actual = cy.contains(expected); // implícito
      actual.should('exist'); // explícito
    });
    it(`THEN: should show 2 items on the projects list`, () => {
      // Assert
      const expected = 2;
      const actual = cy.get('aside:nth-child(1) dt');
      actual.should('have.length', expected);
    });
  });
});
```

16. Navigation

- navegación síncrona
- navegación automática

Navigate - spec

```
describe(`GIVEN: A web app with a menu`, () => {
  // Arrange
  const sutUrl = 'https://angularbuilders.github.io/angular-budget/';
  context(`WHEN: I click on proyectos menu`, () => {
    // Act
    before(() => {
      cy.visit(sutUrl);
      cy.get('a').contains('Proyectos').click();
    });
    it(`THEN: should navigate to the proyectos page`, () => {
      // Assert
      const expected = 'Proyectos | Angular.Budget';
      cy.title().should('include', expected);
    });
    it(`THEN: should see a table with 2 records`, () => {
      // Assert
      const expected = 1 + 2;
      cy.get('table tr').should('have.length', expected);
    });
  });
});
```

17 exploring data

- navegando ANGULAR 404
- usando configuración
- usando mock server

View Project - spec

```
describe(`GIVEN: the url of a project`, () => {
  // Arrange
  const projectsApiMatcher = RegExp(`^${Cypress.env('apiUrl')}projects$`);
  const firstProjectApiMatcher = RegExp(`^${Cypress.env('apiUrl')}projects/mi-primer-proyecto$`);
  context(`WHEN: i visit it`, () => {
    before(() => {
      cy.intercept(projectsApiMatcher,  { fixture: 'projects.json' });
      cy.intercept(firstProjectApiMatcher, { fixture: 'first-project.json' });
      // Act
      cy.visit(Cypress.env('baseUrl'));
      cy.get('a').contains('Proyectos').click();
      cy.contains('Mi primer proyecto').click();
    });
    it(`THEN: I can see project details`, () => {
      // Assert
      const expected = 'Mi primer proyecto';
      cy.get('h2').should('contain', expected);
    });
  });
});
```

18 form interact

- commands
- alias
- esperas

View Project - spec

```
describe(`GIVEN: th page for creating a project`, () => {
  context(`WHEN: fill the project name form and click Crear `, () => {
    before(() => {
      // Arrange
      const projectsApiMatcher = `${Cypress.env('apiUrl')}projects/`;
      cy.intercept('POST', projectsApiMatcher).as('post');
      cy.goToNewProject();
      // Act
      cy.get('#name').type('Testing Project');
      cy.get('button').contains('Crear').click();
    });
    it(`THEN: the project is sent to the server`, () => {
      // Assert
      cy.wait('@post').its('request.body.title').should('equal', 'Testing Project');
    });
  });
});
```

> **Repositorio:** angularbuilders/angular-budget/test_3_e2e
>
> > By Alberto Basalo