QA / TESTING - 04/05/2022 - Abel Rios

> *** Testeando de forma automatizada con el framework Cypress ***
> documentación: https://docs.cypress.io/

¡OJO! Franconsejo: si en el proyecto hacemos algunos test de Cypress en vivo que funcionan queda muy visual e impresiona a los recruiters.

De primeras creamos una carpeta y nos aseguramos de tener instalado el node:

```
npm init -y
```

Una vez creada, a través de consola hacemos:

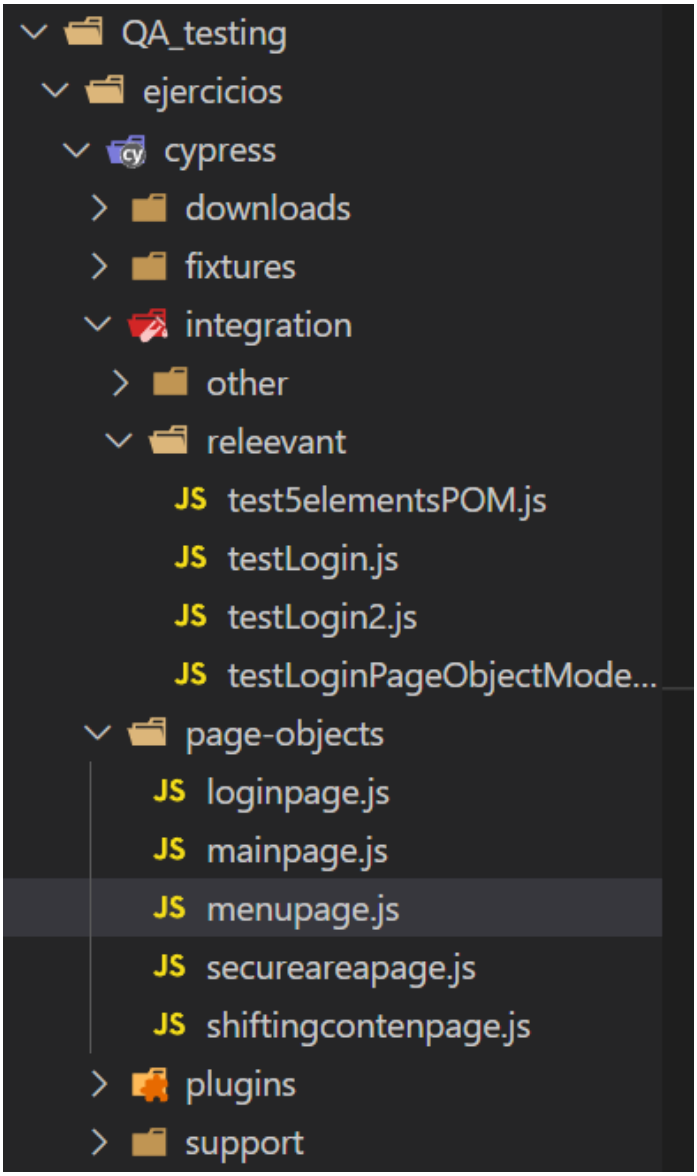
```
npm install cypress
```

Con cypress instalado se nos habrán creado varias carpetas. En la carpeta ‘integration’ creamos una carpeta nueva (‘releevant’ por ejemplo) y ahí creamos nuestro archivo en el que vamos a poner el código de testeo.

Hemos hecho un primer ejercicio con testeo muy arcaico, y un segundo ejercicio agrupando el testeo usando el ‘describe’ y el ‘beforeEach’.

Tras esto hemos aplicado el modelado tipo Page Object Model. Hemos creado una carpeta ‘page-objects’ a la misma altura que la carpeta ‘integration’ y ahí hemos declarado las “clases” de cada página, y en cada “clase” las funciones de los test que tiene que hacer en esas páginas. Estas clases las importamos a nuestro archivo de testeo en la carpeta ‘integration’ y ahora programamos las funciones de los distintos test. Así modulamos nuestros test automatizados y si alguna de las páginas fuese modificada en el futuro podríamos modificar nuestra clase de esa página y ahorrarnos un montón de trabajo.

Estructura de las carpetas:



Exportación de clases:

JS mainpage.js | JS loginpage.js | JS menupage.js | JS shiftingcontenpa

QA_testing > ejercicios > cypress > page-objects > JS mainpage.js > MainPage

```
1   /// <reference types="cypress" />
2
3   export class MainPage{
4
5       navigateMainPage(){
6           cy.visit('https://the-internet.herokuapp.com/')
7       }
8
9       clickFormAuthentication(){
10          cy.contains('Form Authentication').click()
11      }
12
13      clickShiftingContent(){
14          cy.contains('Shifting Content').click()
15      }
16  }
```

JS mainpage.js | JS loginpage.js | JS menupage.js | JS shiftingcontenpage.js

QA_testing > ejercicios > cypress > page-objects > JS loginpage.js > LogInPage

```
1   /// <reference types="cypress" />
2
3   export class LogInPage{
4
5       insertUsername(user){
6           cy.get('#username').type(user)
7       }
8
9       insertPassword(pass){
10          cy.get('#password').type(pass)
11      }
12
13      clickLoginButton(){
14          cy.get('.fa').click()
15      }
16
17  }
```

QA_testing > ejercicios > cypress > page-objects > JS secureareapage.js > ...

```
1  /// <reference types="cypress" />
2
3  export class SecureAreaPage{
4
5      checkFlashMessage(message){
6          cy.get('#flash').contains(message)
7      }
8
9  }
```

Importación en el archivo final (en carpeta integration):

e.js | JS loginpage.js | JS menupage.js | JS shiftingcontenpage.js | JS test5elementsPOM.js | JS testLoginPageObjectModel.js

QA_testing > ejercicios > cypress > integration > releevant > JS testLoginPageObjectModel.js > describe('tests de login') callback > beforeEach() callb

```
1  /// <refe
2
3  import {
4  import { LogInPage } from "../../page-objects/loginpage";
5  import { SecureAreaPage } from "../../page-objects/secureareapage.js";
6
7  describe('tests de login', () => {
8
9      const mainpage = new MainPage;
10     const loginpage = new LogInPage;
11     const secureareapage = new SecureAreaPage;
12
13     beforeEach(() => {
14
15         mainpage.navigateMainPage()
16         mainpage.clickFormAuthentication()
17
18     })
19
20     it('check valid login', () => {
21
22         loginpage.insertUsername("tomsmith");
23         loginpage.insertPassword("SuperSecretPassword!");
```

(alias) class LogInPage
import LogInPage

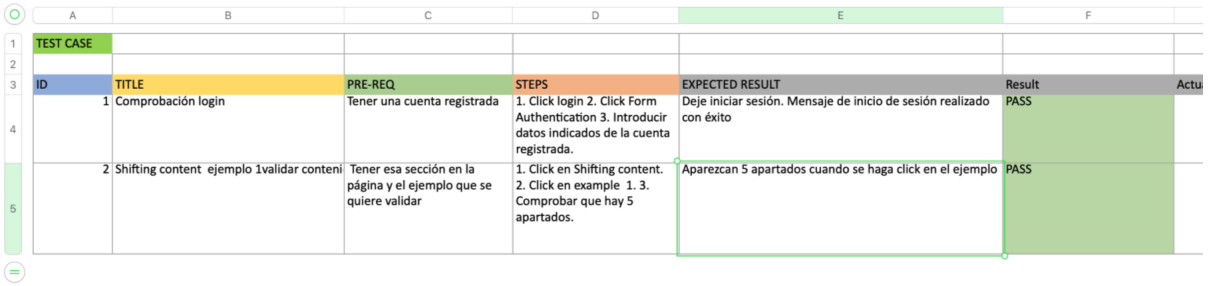
Para abrir la consola de cypress usamos el comando de terminal:

npx cypress open

Y aquí ejecutamos los tests. Para hacerlo a través de consola usamos el comando:

npx cypress run

EJERCICIO: modular los test para el ejercicio de comprobar que en la página de Shifting Content hay 5 elementos (https://the-internet.herokuapp.com/)



	A	B	C	D	E	F	
1	TEST CASE						
2							
3	ID	TITLE	PRE-REQ	STEPS	EXPECTED RESULT	Result	Actu
4	1	Comprobación login	Tener una cuenta registrada	1. Click login 2. Click Form Authentication 3. Introducir datos indicados de la cuenta registrada.	Deje iniciar sesión. Mensaje de inicio de sesión realizado con éxito	PASS	
5	2	Shifting content ejemplo 1validar conteni	Tener esa sección en la página y el ejemplo que se quiere validar	1. Click en Shifting content. 2. Click en example 1. 3. Comprobar que hay 5 apartados.	Aparezcan 5 apartados cuando se haga click en el ejemplo	PASS	