Conceptos generales explicaciones de los mismos

Pruebas de integración

Una prueba de integración tiene como objetivo probar el comportamiento de un componente o la integración entre un conjunto de componentes. El término prueba funcional se usa a veces como sinónimo para prueba de integración. Las pruebas de integración comprueban que todo el sistema funciona según lo previsto, por lo que reducen la necesidad de pruebas manuales intensivas.

Este tipo de pruebas le permiten traducir sus historias de usuario en un conjunto de pruebas. La prueba se asemejaría a una interacción esperada del usuario con la aplicación.

Frameworks de pruebas

Existen una gran variedad de herramientas o frameworks disponibles para las pruebas de integración, tanto para componentes del backend como del frontend. Estas pueden ser comerciales, de código abierto o desarrolladas y utilizadas internamente por las compañías de software.

Para este trabajo práctico vamos a probar aplicaciones web y rest y para ello utilizaremos Codeceptjs como ejemplo.

Selenium

Selenium es una herramienta de prueba de software automatizada y de código abierto para probar aplicaciones web. Tiene capacidades para operar en diferentes navegadores y sistemas operativos. Selenium es un conjunto de herramientas que ayuda a los testers a automatizar las aplicaciones basadas en la web de manera más eficiente.

Podemos codificar las pruebas directamente en un lenguaje de programación, por ejemplo javascript y ejecutarlas como parte del proceso de CI/CD.

Codeceptjs https://codecept.io/

Codeceptjs es un framework end to end para pruebas de integración y de aceptación de usuario, es muy simple de usar y abstrae al que escribe los tests de trabajar directamente con el driver de Selenium o algún otro driver.

1- Familiarizarse con CodeceptJs

En la página de CodeceptJs se encuentra: "The I object is an actor, an abstraction for a testing user. The I is a proxy object for currently enabled Helpers."

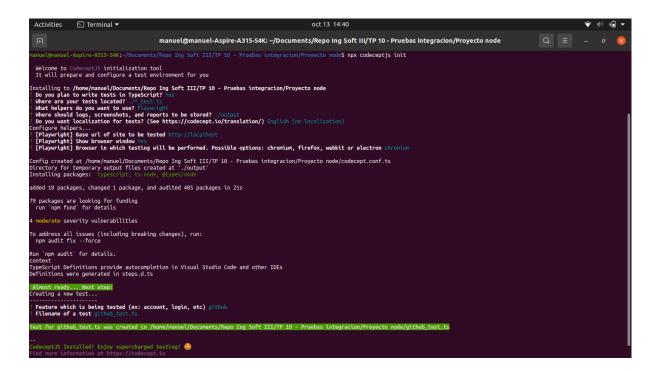
2- Testeando la página de GitHub

```
manuel@manuel-Aspire-A315-54K:~$ node -v
v16.17.0
```

En un directorio, por ejemplo .\proyectos\ut ejecutar:



```
manuel@manuel-Aspire-A315-54K:~/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node$ npm install
up to date, audited 467 packages in 2s
79 packages are looking for funding
   run `npm fund` for details
4 moderate severity vulnerabilities
To address all issues (including breaking changes), run:
   npm audit fix --force
```



```
1 Feature('github');
2
3 Scenario('test something', ({ I }) => {
4     I.amOnPage('https://github.com');
5     I.see('GitHub');
6 });
```

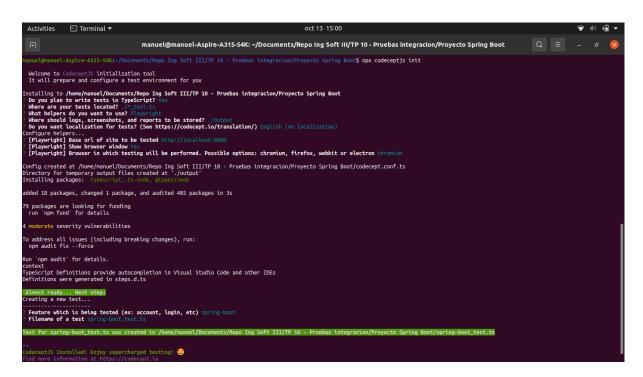
```
manuel@manuel-Aspire-A315-54K:-/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node$ npx codeceptjs run --steps context
CodeceptJS v3.3.6 #StandWithUkraine
Using test root "/home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node"

github --
test something
I am on page "https://github.com"
I see "GitHub"
✓ OK in 2164ms

OK 1 passed // 3s
```

```
1 Feature('github');
  3 Scenario('test something', ({ I }) => {
  4
           I.amOnPage('https://github.com');
           I.see('GitHub');
  5
  6
           I.see('The home for all developers')
  7
           I.scrollPageToBottom()
           I.seeElement("//li[contains(.,'@ 2022 GitHub, Inc.')]")
  8
  9 });
       anuel-Aspire-A315-54K:-/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node$ npx codeceptjs run --steps
CodeceptJS v3.3.6 #StandWithUkraine
Using test root "/home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node"
github --
  Itest something
I am on page "https://github.com"
I see "GitHub"
I see "The home for all developers"
   I see element "//li[contains(.,'@ 2022 GitHub, Inc.')]"
```

3- Testeando la aplicación spring-boot



Se levantó un contenedor con spring-boot en el puerto 8080, y se corrió los test

El test básicamente comprueba que al ir al / en el navegador en el puerto 8080 se obtenga un código de estado 200, luego se verifica que el mensaje de la respuesta obtenida sea Spring boot says hello from a Docker container

4- Habilitar reportes para utilizarlos en CICD

```
10 exports.config = {
    tests: './*_test.js',
output: './output',
11
12
13
    helpers: {
14
       REST: {
         endpoint: "http://localhost:8080",
15
16
         onRequest: () => {
17
         }
       }
18
19
    },
20
    include: {
21
       I: './steps_file.js'
22
    name: 'Proyecto Spring Boot',
23
24
       mocha: {
25
       "reporterOptions": {
         "codeceptjs-cli-reporter": {
26
           "stdout": "-",
27
           "options": {
28
             "steps": true,
29
30
           }
31
         },
32
         "mocha-junit-reporter": {
           "stdout": "./output/console.log",
33
34
           "options": {
             "mochaFile": "./output/result.xml"
35
36
37
           "attachments": true //add screenshot for a failed test
38
             }
           }
39
40
       }
41 }
```

```
manuel@manuel-Asptre-A315-54K:-/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto Spring Boot$ npx codeceptjs run --steps --reporter mo cha-multi
CodeceptjS v3.3.6 #StandWithUkraine
Using test root "/home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto Spring Boot"

spring-boot --
Vertry a successful call
I send get request "/"
✓ OK in 44ms

Verify return value
I send get request "/"
✓ OK in 10ms

OK 2 passed // 71ms
```

5- Integrar la ejecución en Jenkins

código del pipeline de jenkins Jenkins

```
node {
    stage('Build') {

dir("/home/manuel/Documents/Repo_Ing_Soft_III/TP10-Pruebas_integracion/ProyectoSpringBoot/"){
        sh 'npx codeceptjs run --steps --reporter mocha-multi'
      }
    }
    stage('Results') {

dir("/home/manuel/Documents/Repo_Ing_Soft_III/TP10-Pruebas_integracion/ProyectoSpringBoot/output/"){
            junit 'result.xml'
      }
    }
}
```

