

## 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$ npx create-codeceptjs .

CREATE
CODECEPTJS

🚀 Supercharged End 2 End Testing 🌟

Creating CodeceptJS project in /home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node
Powered by Playwright engine
package.json file does not exist in current dir, creating it...
Installing packages: codeceptjs@3, @codeceptjs/ui, @codeceptjs/examples, @codeceptjs/configure, playwright@1
```



```
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
```

```
Activities  Terminal  oct 13 14:40

manuel@manuel-Aspire-A315-54K: ~/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node

manuel@manuel-Aspire-A315-54K:~/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node$ npx codeceptjs init
Welcome to CodeceptJS initialization tool
It will prepare and configure a test environment for you

Installing to /home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node
? Do you plan to write tests in TypeScript? Yes
? Where are your tests located? ./tests
? What helpers do you want to use? Playwright
? Where should logs, screenshots, and reports be stored? ./output
? Do you want localization for tests? (See https://codecept.io/translation/) English (no localization)
Configure helpers...
? [Playwright] Base url of site to be tested http://localhost
? [Playwright] Show browser window Yes
? [Playwright] Browser in which testing will be performed. Possible options: chromium, firefox, webkit or electron chromium

Config created at /home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node/codecept.conf.ts
Directory for temporary output files created at './output'
Installing packages: typescript, ts-node, @types/node
added 18 packages, changed 1 package, and audited 485 packages in 21s

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

Run 'npm audit' for details.
context
TypeScript Definitions provide autocompletion in Visual Studio Code and other IDEs
Definitions were generated in steps.d.ts

Almost ready... Next step:
Creating a new test...
-----
? Feature which is being tested (ex: account, login, etc) github
? Filename of a test github_test.ts

Test for github_test.ts was created in /home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas integracion/Proyecto node/github_test.ts

..
CodeceptJS Installed! Enjoy supercharged testing! 🥳
Find more information at https://codecept.io
```

```

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');
2
3 Scenario('test something', ({ I }) => {
4     I.amOnPage('https://github.com');
5     I.see('GitHub');
6     I.see('The home for all developers')
7     I.scrollPageToBottom()
8     I.seeElement("//li[contains(.,'© 2022 GitHub, Inc.')]")
9 });|

```

```

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"
    I see "The home for all developers"
    I scroll page to bottom
    I see element "//li[contains(.,'© 2022 GitHub, Inc.')]")
  ✓ OK in 2769ms

```

### 3- Testeando la aplicación spring-boot

```
Activities Terminal oct 13 15:00
manuel@manuel-Aspire-A315-54K: ~/Documents/Repo Ing Soft III/TP 10 - Pruebas Integracion/Proyecto Spring Boot$ npx codeceptjs init

Welcome to CodeceptJS initialization tool
It will prepare and configure a test environment for you

Installing to /home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas Integracion/Proyecto Spring Boot
Do you plan to write tests in TypeScript? Yes
Where are your tests located? ./*_test.ts
What helpers do you want to use? Playwright
Where should logs, screenshots, and reports be stored? ./output
Do you want localization for tests? (See https://codecept.io/translation/) English (no localization)
Configure helpers...
[Playwright] Base url of site to be tested http://localhost:8080
[Playwright] Show browser window Yes
[Playwright] Browser in which testing will be performed. Possible options: chromium, firefox, webkit or electron chromium

Config created at /home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas Integracion/Proyecto Spring Boot/codecept.conf.ts
Directory for temporary output files created at './output'
Installing packages: typescript, ts-node, @types/node

added 18 packages, changed 1 package, and audited 485 packages in 3s
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

Run 'npm audit' for details.
context
TypeScript Definitions provide autocompletion in Visual Studio Code and other IDEs
Definitions were generated in steps.d.ts

Almost ready... Next step:
Creating a new test...
-----
Feature which is being tested (ex: account, login, etc) spring-boot
Filename of a test spring-boot_test.ts

Test for spring-boot_test.ts was created in /home/manuel/Documents/Repo Ing Soft III/TP 10 - Pruebas Integracion/Proyecto Spring Boot/spring-boot_test.ts

--
CodeceptJS Installed! Enjoy supercharged testing! 🥳
Find more information at https://codecept.io
```

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

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

spring-boot --
  Verify a successful call
    I send get request "/"
    ✓ OK in 42ms

  Verify return value
    I send get request "/"
    ✓ OK in 17ms

OK 2 passed // 71ms
```

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

```
manuel@manuel-Aspire-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 --
  Verify 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'
    }
    }
}
```

The screenshot shows the Jenkins web interface. At the top is a navigation bar with the Jenkins logo, a search bar, and user information (Manuel Bobadilla). Below the navigation bar is a breadcrumb trail: Dashboard > JUnit > #13 > Test Results. On the left side, there is a sidebar with links to History, Test Result (which is highlighted), Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area is titled 'Test Result' and shows '0 failures'. Below this, there is a section titled 'All Tests' which contains a table with test results. The table has columns for Package, Duration, Fail (diff), Skip (diff), Pass (diff), and Total (diff). The only entry in the table is for the '(root)' package, which has a duration of 42 ms, 0 failures, 0 skips, 2 passes, and a total of 2 tests.

Package	Duration	Fail (diff)	Skip (diff)	Pass (diff)	Total (diff)
(root)	42 ms	0	0	2 +2	2 +2