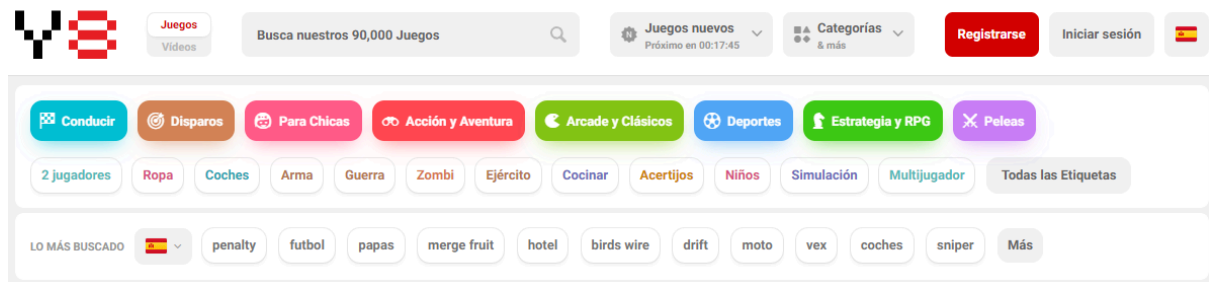


Test i Qualitat del Software

UI Testing

Segona Pràctica



Repository github: <https://github.com/GerardGV/cucumber.git>

Pagina per testear: <https://es.y8.com/>

Jan Rubio Rico	1603753
Gerard Josep Guarín Velez	1605947

Índex:

Índex:	2
FEATURES LIST	3
Complex Scenaris	4
1. Check Instagram	4
2. Loading time	5
3. Page arrows	7
4. Register	8
5. Watch Video	10
6. Add Friend	12
Conclusions	13

FEATURES LIST

1. **Going home from icon:** d'aquest feature test iniciar en una pàgina diferent de la inicial i comprovar que es pot anar a la pàgina inicial fent click a l'icona de Y8.
2. **Select category:** en la pàgina principal de Y8 trobem un apartar amb categories. L'objectiu d'aquest test és veure que es carreguen les pàgines de 2 categories diferents.
3. **Watch all the labels:** test per al correcte funcionament del botó que et porta a veure totes les etiquetes que n'hi ha a Y8.
4. **Change language:** Y8 té la funcionalitat que permet canviar l'idioma de la pàgina web. Aquest test comprova que amb la pàgina, inicialment en castellà, fa el canvi del text del botó d'inici de sessió a Log In que es selecciona la llengua anglesa.
5. **Add Friend:** Feature per a comprovar que es pot afegir amics a la plataforma per tal de generar una xarxa d'amistats.
6. **Add Game to Favourite:** Afegir jocs que vulguem a preferits per si volem accedir a ells seguidament i tenir-los en una llista al nostre perfil.
7. **Check Instagram:** explicat amb profunditat en el següent apartat.
8. **Comment on video:** Afegir comentaris a vídeos que hi ha en la plataforma per si volem comentar alguna cosa públicament.
Like video: Marcar que el video anterior video t'ha agradat.
9. **Edit Profile:** Editar la informació del nostre perfil sempre que vulguem
10. **Loading Time:** explicat amb profunditat en el següent apartat.
11. **Login:** Iniciar sessió en el nostre usuari anteriorment registrat
12. **Page arrows:** explicat amb profunditat en el següent apartat.
13. **Register:** Enregistrar-nos a la plataforma per a tenir un usuari.
14. **Search game (Russian car driver):** Cercar un joc en concret i entrar a ell per a posteriorment jugar
15. **Watch video:** Mirar un vídeo en concret en la secció de vídeos de la plataforma.

Complex Scenaris

1. Check Instagram

Aquest test té l'objectiu de comprovar que el funcionament correcte del botons al final de la página els quals redireccionen cap una xarxa social. L'escenari es el següent:

Scenario Outline: Checking Instagram

Given the user is in the main page for Social Media test

When the user clicks `<social_media>` icon

Then the title of new web page contains `<social_media>`

El parametre social_media té 2 valors que es testegen:

Examples:

| social_media |

| Instagram |

| Facebook |

En els steps començem a la página inicial on esperem que aparegui el pop up demanant acceptar les cookies:

```
gerard *
@Given("the user is in the main page for Social Media test")
public void theUserInMainPage(){
    System.setProperty("webdriver.chrome.driver", "Drivers/chromedriver");
    driver = new ChromeDriver();
    driver.navigate().to( s: "https://es.v8.com");

    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
    wait.until(ExpectedConditions.presenceOfElementLocated(By.className("css-ik5ir8")));
    if (driver.findElement(By.className("css-ik5ir8")).isDisplayed()) { // Cookies popup displayed
        driver.findElement(By.className("css-ik5ir8")).click();
    }
}
```

En el següent step busquem la icona de la xarxa social i fem click:

```

+gerard
@When("^the user clicks (.*) icon")
public void userClicksSocialMedia(String socialMedia) throws InterruptedException {

    driver.findElement(By.cssSelector("[aria-label='" + socialMedia + "'][title='" + socialMedia + "']")).click();
    sleep( millis: 3000);
}

```

Després d'esperar 3 segons es fa l'últim step on obtenim les finestres al driver, ja que els icones de les xarxes socials obren aquestes en noves finestres.

```

//Getting all the windows un the driver
ArrayList<String> tabs = new ArrayList<String> (driver.getWindowHandles());
while(tabs.size() <= 1){
    tabs = new ArrayList<String> (driver.getWindowHandles());
}

```

Una vegada guardem totes les finestres del driver en la variable tabs, es fa que el driver canvi a la nova finestra on s'obté el títol i es comprova si el nom de la xarxa social es troba en aquest.

```

driver.switchTo().window(tabs.get(1));
String text= driver.getTitle();
Assert.assertTrue(text.contains(socialMedia));
driver.quit();

```

Aquest steps es fan per les xarxes socials instagram i Facebook. Hem considerat aquest feature test complexa ja que hem dedicat molta estona a intentar solventar el problema de que es detecta que selenium és qui està navegant i no una persona, encara que posem molts sleeps o no esborrar les cookies el resultat en la pàgina es d'error 401 indicant que la petició (request) no ha estat executada perquè no té credencials vàlides d'autenticació per al recurs sol·licitat. Al final vam concloure que lo millor seria comprovar que es redirecciona a instagram amb el títol de la pàgina.

2. Loading time

L'objectiu d'aquest test és comprobar que el temps de càrrega d'un joc no és superior a 7 segons. Per aquest test tenim els 3 steps següents:

```
Given the user is in main page to select to measuring loading time
When the user clicks the first game
Then checking loading time
```

Quan carreguem la pagina inicial hem de acceptar les cookies. Per aquesta rao tenim el web driver esperant que aparegui un element amb la class="css-ik5ir8".

```
@Given("the user is in main page to select to measuring loading time")
public void userMainPage(){
    System.setProperty("webdriver.chrome.driver", "Drivers/chromedriver");
    driver = new ChromeDriver();
    driver.navigate().to(s: "https://es.y8.com/");
    driver.manage().deleteAllCookies();

    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
    wait.until(ExpectedConditions.presenceOfElementLocated(By.className("css-ik5ir8")));
    if (driver.findElement(By.className("css-ik5ir8")).isDisplayed()) { // Cookies popup displayed
        driver.findElement(By.className("css-ik5ir8")).click();
    }
}
```

En el següent step fem click al primer joc de la pagina que n'hi ha a la pàgina principal i guardem el temps actual en un atribut amb el nom start de tipus Instant:

```
@When("the user clicks the first game")
public void clicksFirstGame(){
    //saving list of games
    List<WebElement> games = driver.findElements(By.cssSelector("#items_container li a"));

    //clicking first game
    games.get(0).click();

    start= Instant.now();
}
```

En l'últim step s'espera que el web driver carregui el container que té el joc i una vegada carregat es guarda el instant de temps en la variable end. Amb les variables start i end es calcula la diferència de temps per saber si no ha sigut major a 7 segons.

```

@Then("checking loading time")
public void checkingLoadingTime(){
    try {

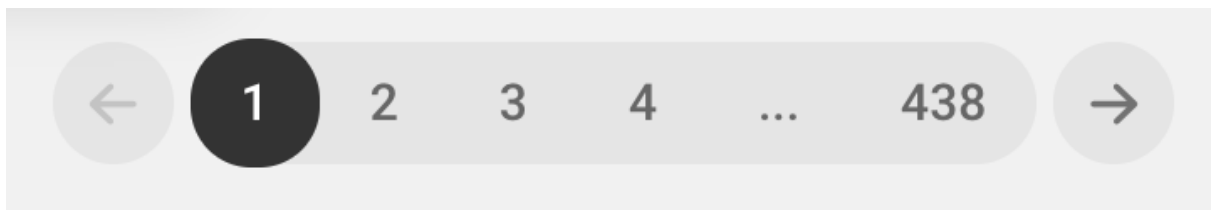
        WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector("#item-container")));

        Instant end=Instant.now();
        long milliSecDiff=end.toEpochMilli()-start.toEpochMilli();
        System.out.println(milliSecDiff);
        Assert.assertTrue((milliSecDiff <= 70000));
    } catch (NoSuchElementException e) {
        System.out.println("El elemento no se encontró.");
    }
    driver.quit();
}

```

3. Page arrows

En aquest test es per la feature de canvi de pagina. Es fa la comprovació amb la fletxa dreta en la part inferior de la página.



Per aquest feature tenim 3 steps:

```

Given the user is in main page to select to measuring loading time
When the user clicks the first game
Then checking loading time

```

El primer step és el mateix primer step que al test anterior. Es carrega amb el web driver la pàgina inicial de Y8. EN el segon step es busca l'element que conté la fletxa dreta y es fa click.

```

@When("the user clicks the right arrow to go to the next web page of games")
public void clicksNextArrow() { driver.findElement(By.cssSelector("a.arrow.next")).click(); }

```

En l'últim step es verifica que a la URL hi ha un paràmetre page amb el valor 2:

```
@Then("check if the web page is the next one")
public void checkingPageChange() throws MalformedURLException {
    String currentUrl = driver.getCurrentUrl();

    // creating URL class
    URL url = new URL(currentUrl);

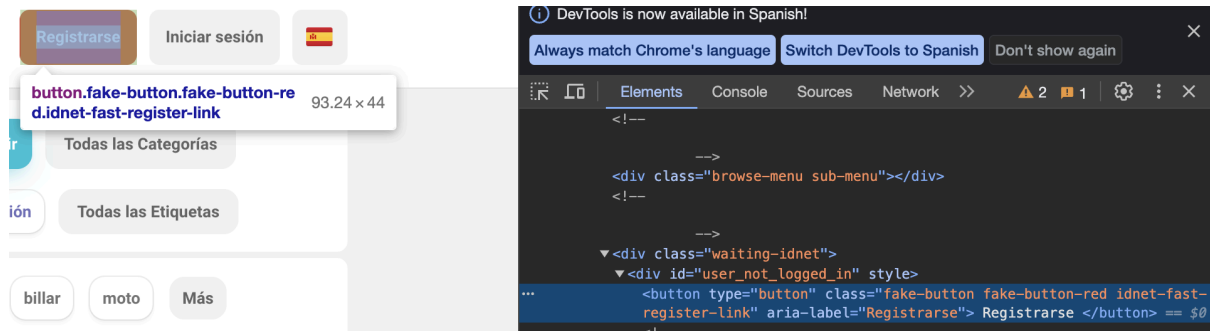
    // Getting REST parameters from the URL
    String parametersRest = url.getQuery();

    // Imprime los parámetros
    Assert.assertEquals(parametersRest, expected: "page=2");
    driver.quit();
}
```

4. Register

Aquesta feature va tenir certes complicacions per tant explicarem els passos que hem seguit per a aquesta feature.

Inicialment iniciem per carregar la pàgina inicial de la web, com a totes les features, per tant saltarem aquest pas inicial. Seguidament una vegada carreguem aquesta pàgina fem click sobre el botó de register que es pot trobar adalt a la dreta de la pàgina inicial fent ús de la seva classe (idnet-fast-register-link).



Una vegada fem click en aquest botó es desplega un "iframe" per tant el que farem serà canviar a aquest per a poder interaccionar amb ell ja que si no ho fem el nostre driver no podrà seleccionar els elements que conté ja que no estarà buscant en el nivell que toca, sino que es quedarà en la finestra anterior.


```
String iframeName = null;
try {
    PrintWriter writer = new PrintWriter( fileName: "log.txt", cs: "UTF-8");
    final List<WebElement> iframes = driver.findElements(By.tagName("iframe"));
    for (WebElement iframe : iframes) {
        if (!Objects.equals(iframe.getAttribute(s: "id"), b: "easyXDM_id_xd_api_provider")){
            writer.println(iframe.getAttribute(s: "id"));
            iframeName = iframe.getAttribute(s: "id");
        }
    }
    writer.close();
} catch (IOException e) {
    System.out.println("An error occurred.");
    e.printStackTrace();
}
```

Com podem veure en aquest codi tenim un arxiu en el qual estem escrivint, en aquest cas no fa res en especial però ho tenim posat per tal de controlar si és necessari el nom dels iframes generats una vegada inspeccionem l'arxiu on s'escriu.

Una vegada hem canviat al iframe que toca comencem a introduir les dades en les seccions pertinents i passant als següents punts del formulari de registre, com per exemple omplint el nom i el correu i donant al botó de “next”:

```
@When("^the user enters (.*) in the name button")
public void userEntersName(String name) { driver.findElement(By.id("meta_first_name")).sendKeys(name); }
```

```

TheRospetit
@When("^the user enters in the email bar his (.*)")
public void userEntersTemporalEmail(String emailRegister){
    WebDriverWait wait = new WebDriverWait(driver, timeoutInSeconds: 2);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("meta_email")));
    driver.findElement(By.id("meta_email")).sendKeys(emailRegister);
}

```

```

@When("the user clicks next 1")
public void userClickNext(){
    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 2);
    wait.until(ExpectedConditions.visibilityOf(driver.findElement(By.className("next")).findElement(By.className("norm"))));
    driver.findElement(By.className("next")).click();
}

```

També no només hem d'omplir els camps necessaris sinó que hem de seleccionar entre dues opcions de sexe (home o dona) i seleccionar en uns desplegable la data de naixement (any i mes).

```

TheRospetit
@When("the user chooses boy or girl")
public void userClickBoyOrGirl(){
    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 2);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.className("male-icon")));
    driver.findElement(By.className("male-icon")).click();
}

```

```

TheRospetit
@When("the user chooses year")
public void userSelectsYear(){
    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 5);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("meta_date_of_birth_1i")));
    WebElement dropdownElement = driver.findElement(By.id("meta_date_of_birth_1i"));
    Select dropdown = new Select(dropdownElement);

    // Seleccionamos un año en particular
    dropdown.selectByValue("1996");
}

```

Finalment donem al botó de “finish” per a acabar amb el registre, tornem al iframe principal i comprovem que ens hem registrat correctament fent ús de l'aparició d'un avatar.

```

TheRospetit *
@When("the user clicks finish")
public void userClickFinish(){
    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 2);
    wait.until(ExpectedConditions.visibilityOf(driver.findElement(By.className("next")).findElement(By.className("finish"))));
    driver.findElement(By.className("finish")).click();
    driver.switchTo().parentFrame(); // Return to original iframe
}

TheRospetit
@Then("the user is registered")
public void userRegistered(){
    // El título es el que toca
    try {
        WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 3);
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.className("avatar")));
        Assert.assertNotNull(driver.findElement(By.className("avatar"))); // Esto lanzará una excepción si el elemento no se encuentra
    } catch (NoSuchElementException e) {
        System.out.println("El elemento no se encontró.");
    }
}

```

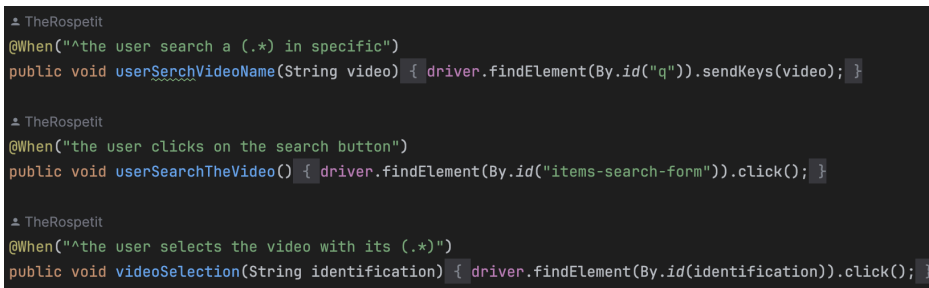
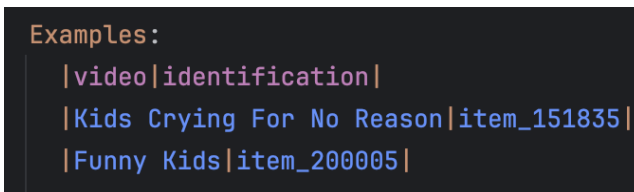
5. Watch Video

En aquesta feature fem ús d'un altre apartat que té la nostra pàgina web i és que tenen una secció de videos semblant a la mundialment coneguda YouTube. En aquest cas sembla ser una versió bastant més senzilla per tant podem realitzar features com la següent on ens disposem a visualitzar un video en concret.

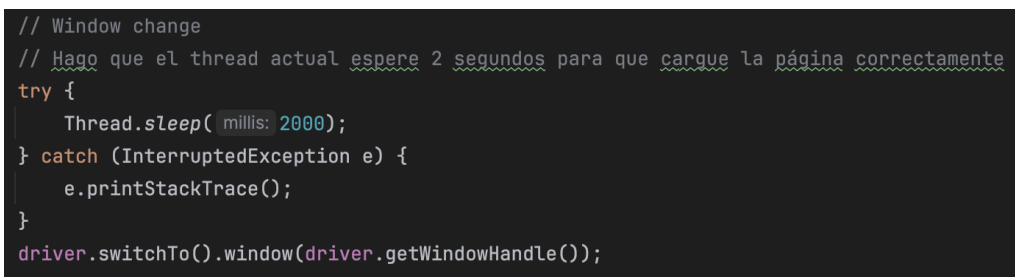
Iniciarem directament en la pàgina inicial on no ens farà falta iniciar sessió per tal de poder buscar i visualitzar un video. Començarem per dirigir-nos a la secció de videos fent click al switch que canvia entre la secció de jocs i la de videos.



Buscarem el video en qüestió fent ús del cercador, on introduïm el seu nom, i seleccionarem el video fent ús de la seva id d'identificació única que passem per paràmetres.



Una vegada seleccionem el video farem que la pàgina esperi 2 segons per tal de que carregui correctament la nova finestra abans de canviar a aquesta.



Donarem "play" fent click al video.

```
// Play video
WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
wait.until(ExpectedConditions.presenceOfElementLocated(By.className("item-container")));
driver.findElement(By.className("item-container")).click();
```

I finalment comprovarem que el video s'està executant fent ús de l'aparició d'una classe anomenada (`is-playing`), la qual controlarà que la feature s'ha dut a terme correctament.

```

TheRospetit
@Then("video is playing")
public void confirmationVideoPlaying(){

    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.className("is-playing")));
    String classes = driver.findElement(By.id("item-direct-container")).getAttribute("class");
    boolean isPlaying = classes.contains("is-playing");

    // Utiliza Assert para verificar si el video está en pausa
    Assert.assertTrue(isPlaying, message: "El video no se está reproduciendo");

    driver.quit();
}

```

6. Add Friend

L'última feature que comentarem serà la d'afegir a amics a un usuari aleatori.

Per a aquesta feature iniciarem de nou a la pàgina inicial de la web i iniciem sessió per tal de poder accedir a les funcions d'amics.

```

driver.switchTo().frame(driver.findElement(By.id(iframeName))); // We switch to the iframe tha

wait = new WebDriverWait(driver, timeOutInSeconds: 3); // wait some time if necessary
wait.until(ExpectedConditions.presenceOfElementLocated(By.id("account_email")));

driver.findElement(By.id("account_email")).sendKeys(userEmail); // Enter email

driver.findElement(By.id("account_password")).sendKeys(userPassword); // Enter password

wait = new WebDriverWait(driver, timeOutInSeconds: 2);
wait.until(ExpectedConditions.visibilityOf(driver.findElement(By.className("login"))));
driver.findElement(By.className("login")).click();
driver.switchTo().parentFrame();

driver.navigate().refresh();

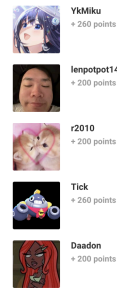
```

Una vegada iniciada la sessió, per tal d'afegir un amic aleatori que ens suggereix la pàgina web el que farem serà anar al nostre perfil, on si anem una miqueta cap abaix trobem una secció que posa “¿Quieres más amigos?” on podem seleccionar qualsevol dels usuaris.

¿QUIERES MÁS AMIGOS?

```
± TheRospetit
@When("the user enters in its profile")
public void theUserEntersInItsProfile() {
    driver.navigate().refresh();
    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 2);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.className("user-toggle")));
    driver.findElement(By.className("user-toggle")).click();

    wait = new WebDriverWait(driver, timeOutInSeconds: 2);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("account-menu-link-edit-profile")));
    driver.findElement(By.id("account-menu-link-profile")).click();
}
```



D'aquesta llista d'usuaris seleccionem el primer d'ells que troba el nostre driver al fer una cerca mitjançant la classe "suggested-friend" de la qual farem click a la classe "avatar" que conté per tal de que ens redireccioni al perfil de l'usuari per posteriorment enviar sol·licitud d'amistat.

```
± TheRospetit
@When("the user selects the first person in its friend suggestions")
public void theUserSelectsTheFirstPersonInItsFriendSuggestions() {
    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.className("suggested-friend")));
    driver.findElement(By.className("suggested-friend")).findElement(By.className("avatar")).click();
}
```

Enviem la sol·licitud d'amistat fent ús del botó amb classe "add-friend-button".

```
± TheRospetit
@When("the user sends the friend invitation")
public void theUserSendsTheFriendInvitation() {
    // CHANGE DRIVER WINDOW
    Set<String> allWindows = driver.getWindowHandles();
    allWindows.remove(driver.getWindowHandle());
    String newWindow = allWindows.iterator().next();
    driver.switchTo().window(newWindow);

    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 5);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.className("add-friend-button")));
    driver.findElement(By.className("add-friend-button")).click();
}
```

Finalment comprovem que s'ha mostrat una notificació amb classe "flash-notice" la qual com a text intern té "Friend request sent".

```
± TheRospetit
@Then("the friend invitation is sent correctly")
public void theFriendInvitationIsSentCorrectly() {
    WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("flash-notice")));
    String friendRequest = driver.findElement(By.id("flash-notice")).getText();
    Assert.assertEquals(friendRequest, expected: "Friend request sent", message: "The user did not send the friend request");
}
```

Conclusions

En aquest projecte hem après a usar cucumber i selenium en cert nivell, si bé hem tingut certs problemes en algunes features i ens hem quedat encallats en algun punt, com per exemple en controlar el driver si canviàvem a una nova finestra o es generava un iframe en la pàgina, considerem que ens ha aportat certs coneixements molt útils a l'hora de provar característiques d'una pàgina web amb la informació accessible d'aquesta.

Aquesta pràctica ens ha obert les portes a certs aspectes realment necessaris per a comprovar i assegurar una qualitat del software, permetent aplicar aquells coneixements fins ara teòrics de manera pràctica.