



Università degli Studi di Salerno

RAAF-GAMING
Test Execution Report
Versione 1.0

Data: 31/12/2021

Partecipanti:

Nome	Matricola
Rocco Iuliano	0512106804
Francesco Peluso	0512107194
Antonio De Lucia	0512109225
Antonio Maddaloni	0512107890

Revision History

Data	Versione	Descrizione	Autore
31/12/2021	1.0	Prima stesura del documento	Membri del team

INDICE

1. Profilo

- 1.1. Registrazione**
- 1.2. Modifica dati profilo**
- 1.3. Login**
- 1.4. Login gestore**

2. Acquisto

- 2.1. Acquisto prodotto**
- 2.2. Eliminazione prodotto**
- 2.3. Visualizzazione carrello**

3. Prodotto

- 3.1. Visualizza ordine**
- 3.2. Gestione ordine**
- 3.3. Recensione**
- 3.4. Aggiunta al carrello**
- 3.5. Rifornitura**
- 3.6. Ricerca per categoria**
- 3.7. Ricerca per nome**
- 3.8. Fornitura nuovo prodotto**

1. PROFILO

1.1. REGISTRAZIONE

Registrazione eseguita:

```

public void testRegistrazioneEseguito() throws InterruptedException {
    // Step # | name | target | value
    // 1 | open | http://localhost:8080/RAAF-GAMING/ServletRegistrazione |
    driver.get("http://localhost:8080/RAAF-GAMING/ServletRegistrazione");
    // 2 | setWindowSize | 1050x700 |
    //driver.manage().window().setSize(new Dimension(1050, 700));
    // 3 | click | id=validationCustom01 |
    driver.findElement(By.id("validationCustom01")).click();
    // 4 | type | id=validationCustom01 | Francesco
    driver.findElement(By.id("validationCustom01")).sendKeys("Francesco");
    Thread.sleep(1000);
    // 5 | click | id=validationCustom02 |
    driver.findElement(By.id("validationCustom02")).click();
    // 6 | type | id=validationCustom02 | Peluso
    driver.findElement(By.id("validationCustom02")).sendKeys("Peluso");
    Thread.sleep(1000);
    // 7 | click | id=datadinascita |
    driver.findElement(By.id("datadinascita")).click();
    // 8 | type | id=datadinascita | 2000-09-11
    driver.findElement(By.id("datadinascita")).sendKeys("11/09/2000");
    //Thread.sleep(2000);
    // 9 | click | id=validationCustom07 |
    driver.findElement(By.id("validationCustom07")).click();
}

```

Registrazione non eseguita:

```

public void testRegistrazioneEseguito() throws InterruptedException {
    // Step # | name | target | value
    // 1 | open | http://localhost:8080/RAAF-GAMING/ServletRegistrazione |
    driver.get("http://localhost:8080/RAAF-GAMING/ServletRegistrazione");
    // 2 | setWindowSize | 1050x700 |
    //driver.manage().window().setSize(new Dimension(1050, 700));
    // 3 | click | id=validationCustom01 |
    driver.findElement(By.id("validationCustom01")).click();
    // 4 | type | id=validationCustom01 | Francesco
    driver.findElement(By.id("validationCustom01")).sendKeys("Francesco");
    Thread.sleep(1000);
    // 5 | click | id=validationCustom02 |
    driver.findElement(By.id("validationCustom02")).click();
    // 6 | type | id=validationCustom02 | Peluso
    driver.findElement(By.id("validationCustom02")).sendKeys("Peluso");
    Thread.sleep(1000);
    // 7 | click | id=datadinascita |
    driver.findElement(By.id("datadinascita")).click();
    // 8 | type | id=datadinascita | 2000-09-11
    driver.findElement(By.id("datadinascita")).sendKeys("11/09/2000");
    //Thread.sleep(2000);
    // 9 | click | id=validationCustom07 |
    driver.findElement(By.id("validationCustom07")).click();
}

public void testRegistrazioneErroreEmailEsistente() throws InterruptedException {
    // Step # | name | target | value
    // 1 | open | http://localhost:8080/RAAF-GAMING/ServletRegistrazione |
    driver.get("http://localhost:8080/RAAF-GAMING/ServletRegistrazione");
    // 2 | setWindowSize | 1050x700 |
    //driver.manage().window().setSize(new Dimension(1050, 700));
    // 3 | click | id=validationCustom01 |
    driver.findElement(By.id("validationCustom01")).click();
    // 4 | type | id=validationCustom01 | Francesco
    driver.findElement(By.id("validationCustom01")).sendKeys("Francesco");
    Thread.sleep(1000);
    // 5 | click | id=validationCustom02 |
    driver.findElement(By.id("validationCustom02")).click();
    // 6 | type | id=validationCustom02 | Peluso
    driver.findElement(By.id("validationCustom02")).sendKeys("Peluso");
    Thread.sleep(1000);
    // 7 | click | id=datadinascita |
    driver.findElement(By.id("datadinascita")).click();
    // 8 | type | id=datadinascita | 2000-09-11
    driver.findElement(By.id("datadinascita")).sendKeys("11/09/2000");
    //Thread.sleep(2000);
    // 9 | click | id=validationCustom07 |
    driver.findElement(By.id("validationCustom07")).click();
}

```

1.2 MODIFICA DATI DEL PROFILO

Modifica eseguita:

```

public void testProfiloEseguito() throws InterruptedException {
    // Test name: TestProfiloEseguito
    // Step # | name | target | value | comment
    // 1 | login | id=validationCustom01 | Francesco | 
    driver.findElement(By.id("validationCustom01")).click();
    driver.findElement(By.id("validationCustom01")).sendKeys("Francesco");
    Thread.sleep(1000);
    // 2 | login | id=validationCustom02 | Peluso | 
    driver.findElement(By.id("validationCustom02")).click();
    driver.findElement(By.id("validationCustom02")).sendKeys("Peluso");
    Thread.sleep(1000);
    // 3 | click | id=datadinascita |
    driver.findElement(By.id("datadinascita")).click();
    // 4 | type | id=datadinascita | 2000-09-11
    driver.findElement(By.id("datadinascita")).sendKeys("11/09/2000");
    //Thread.sleep(2000);
    // 5 | click | id=validationCustom07 |
    driver.findElement(By.id("validationCustom07")).click();
}

```


The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Help, File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Left Sidebar:** Project Explorer, JUnit View, Outline View, Task List.
- Middle Area:** A code editor window titled "TestProfilo.java X" containing Java code for a Selenium test. The code performs various interactions with a web application, including entering a password, clicking buttons, and selecting dropdowns. It includes several sleep statements and assertions.
- Bottom Status Bar:** Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit X, Synchronize. It also shows the message "finished after 32,107 seconds".
- Bottom Navigation Bar:** Runs: 1/1, Errors: 0, Failures: 0. The failure count is highlighted in red.
- Bottom Status Bar:** Failure Trace.

The screenshot shows the Eclipse IDE interface in a different workspace (eclipse-workspace - RAAF-GAMING) with the following details:

- Top Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Left Sidebar:** Project Explorer, JUnit View, Outline View, Task List.
- Middle Area:** A code editor window titled "TestProfilo.java X" containing Java code for a Selenium test. The code is identical to the one in the first screenshot, performing the same series of interactions with the web application.
- Bottom Status Bar:** Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit X, Synchronize. It shows the message "finished after 31,997 seconds".
- Bottom Navigation Bar:** Runs: 1/1, Errors: 0, Failures: 0. The failure count is highlighted in red.
- Bottom Status Bar:** Failure Trace.

Modifica non eseguita:

```
256@Test
257 public void testProfiloErroreData() throws InterruptedException {
258     // Test name: TestProfiloErroreData
259     // Step # | name | target | value
260     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |
261     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");
262     Thread.sleep(2000);
263     // 2 | click | name=email |
264     driver.findElement(By.name("email")).click();
265     Thread.sleep(2000);
266     // 3 | type | name=email | f.peluso25@gmail.com
267     driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
268     Thread.sleep(2000);
269     // 4 | click | name=password |
270     driver.findElement(By.name("password")).click();
271     Thread.sleep(2000);
272     // 5 | type | name=password | veloce123
273     driver.findElement(By.name("password")).sendKeys("veloce123");
274     Thread.sleep(2000);
275     // 6 | click | css=.invio |
276     driver.findElement(By.cssSelector(".invio")).click();
277     Thread.sleep(2000);
278     // 7 | click | css=.fa-user-astronaut |
279     driver.findElement(By.cssSelector(".fa-user-astronaut")).click();
280     Thread.sleep(2000);
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 44,51 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestProfilo [Runner: JUnit 5] (44,456 s)

testProfiloErroreData (44,456 s)

Failure Trace

The screenshot shows a Java test script named `TestProfilo.java` in an IDE. The code is annotated with line numbers (232 to 256) and contains several Selenium WebDriver commands. The script interacts with three input fields (`inputCarta2`, `inputCarta3`, and `codicecartaselenuim`) and a primary button. It also handles an alert message. The final line attempts to find an element by name `codicecartaselenuim`.

```
232 e | id=inputCarta2 | 123
233 Element(By.id("inputCarta2")).sendKeys("123");
234 p(2000);
235 ck | id=inputCarta3 |
236 Element(By.id("inputCarta3")).click();
237 p(2000);
238 e | id=inputCarta3 | 2022-02-05
239 Element(By.id("inputCarta3")).sendKeys("05/02/2022");
240 p(2000);
241 ck | css=.btn-primary |
242 Element(By.cssSelector(".btn-primary")).click();
243 p(3000);
244 ertAlert | password non valida |
245 = driver.switchTo().alert();
246 = alert.getText();
247 println(str);
248 s("password non valida",str);
249 t();
250 p(2000);
251 ertText | name=codicecartaselenuim | *****8989
252 driver.findElement(By.name("codicecartaselenuim")).getText(), is("*****8989"));
253
254
255
256
```

Below the code editor, the IDE's navigation bar includes tabs for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (which is currently selected), and Synchronize. The status bar indicates the test completed after 45,158 seconds with 1 run, 0 errors, and 0 failures.

Run details:

- Runs: 1/1
- Errors: 0
- Failures: 0

Failure Trace:

- TestProfilo [Runner: JUnit 5] (45,104 s)
- testProfiloErrorPWD (45,104 s)

The screenshot shows an IDE interface with a Java code editor and various toolbars and panels.

Code Editor:

```
Help  
File Project Tools Window Help  
TestProfilo.java X  
669 @Test  
670 public void testProfiloErroreSoloCarta() throws InterruptedException {  
671     // Test name: TestProfiloErroreSoloData  
672     // Step # | name | target | value  
673     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |  
674     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");  
675     Thread.sleep(2000);  
676     // 2 | setWindowSize | 1050x700 |  
677     // driver.manage().window().setSize(new Dimension(1050, 700));  
678     // 3 | click | name=email |  
679     driver.findElement(By.name("email")).click();  
680     Thread.sleep(2000);  
681     // 4 | type | name=email | f.peluso25@gmail.com  
682     driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");  
683     Thread.sleep(2000);  
684     // 5 | click | name=password |  
685     driver.findElement(By.name("password")).click();  
686     Thread.sleep(2000);  
687     // 6 | type | name=password | veloce123  
688     driver.findElement(By.name("password")).sendKeys("veloce123");  
689     Thread.sleep(2000);  
690     // 7 | click | css=.invio |  
691     driver.findElement(By.cssSelector(".invio")).click();  
692     Thread.sleep(2000);  
693     // 8 | click | class=fa fa-unlock-alt |
```

Toolbars and Status Bar:

- File Project Tools Window Help
- Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize
- Finished after 34,047 seconds
- Runs: 1/1 Errors: 0 Failures: 0

Failure Trace:

- TestProfilo [Runner: JUnit 5] (33,993 s)
 - testProfiloErroreSoloCarta (33,993 s)

The screenshot shows a Java test class named `TestProfilo.java` in an IDE. The code implements a test method `testProfiloErroreSoloCarta()` using Selenium WebDriver. The test steps include opening a URL, setting window size, clicking on email input, sending keys, clicking on password input, sending keys, and clicking on the send button. The test is annotated with `@Test`. The code is heavily commented with step descriptions.

```
669 @Test  
670 public void testProfiloErroreSoloCarta() throws InterruptedException {  
671     // Test name: TestProfiloErroreSoloData  
672     // Step # | name | target | value  
673     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |  
674     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");  
675     Thread.sleep(2000);  
676     // 2 | setWindowSize | 1050x700 |  
677     // driver.manage().window().setSize(new Dimension(1050, 700));  
678     // 3 | click | name=email |  
679     driver.findElement(By.name("email")).click();  
680     Thread.sleep(2000);  
681     // 4 | type | name=email | f.peluso25@gmail.com  
682     driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");  
683     Thread.sleep(2000);  
684     // 5 | click | name=password |  
685     driver.findElement(By.name("password")).click();  
686     Thread.sleep(2000);  
687     // 6 | type | name=password | veloce123  
688     driver.findElement(By.name("password")).sendKeys("veloce123");  
689     Thread.sleep(2000);  
690     // 7 | click | css=.invio |  
691     driver.findElement(By.cssSelector(".invio")).click();  
692     Thread.sleep(2000);  
693     // 8 | click | name=submit |  
694 }
```

Below the code editor, the IDE interface includes tabs for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (which is selected), and Synchronize. The status bar indicates "Finished after 36,088 seconds". At the bottom, there is a summary: "Runs: 1/1", "Errors: 0", "Failures: 0", and a "Failure Trace" section which is currently empty.

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Help, various icons for file operations, and a search bar.
- Left Sidebar:** Shows the project structure with a tree view of files and folders.
- Central Area:** A code editor window titled "TestProfilo.java X" containing the following Java code:

```
610 @Test
611 public void testProfiloErroreSoloData() throws InterruptedException {
612     // Test name: TestProfiloErroreSoloData
613     // Step # | name | target | value
614     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |
615     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");
616     Thread.sleep(2000);
617     // 2 | setWindowSize | 1050x700 |
618     // driver.manage().window().setSize(new Dimension(1050, 700));
619     // 3 | click | name=email |
620     driver.findElement(By.name("email")).click();
621     Thread.sleep(2000);
622     // 4 | type | name=email | f.peluso25@gmail.com
623     driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
624     Thread.sleep(2000);
625     // 5 | click | name=password |
626     driver.findElement(By.name("password")).click();
627     Thread.sleep(2000);
628     // 6 | type | name=password | veloce123
629     driver.findElement(By.name("password")).sendKeys("veloce123");
630     Thread.sleep(2000);
631     // 7 | click | css=.invio |
632     driver.findElement(By.cssSelector(".invio")).click();
633     Thread.sleep(2000);
634     // 8 | click | class=buttonSubmit |
635 }
```

- Bottom Status Bar:** Shows "Runs: 1/1", "Errors: 0", and a progress bar indicating the test finished after 33,863 seconds.
- Bottom Left:** Shows the JUnit runner interface with a summary: "TestProfilo [Runner: JUnit 5] (33,807 s)" and "testProfiloErroreSoloData (33,807 s)".
- Bottom Right:** A "Failure Trace" button.

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Help, File, New, Open, Save, Cut, Copy, Paste, Find, Replace, etc.
- Left Sidebar:** Test Profilo.java (selected), Servers, Data Source Explorer, Snippets, Console, JUnit (selected), Synchronize.
- Bottom Status Bar:** Finished after 31,939 seconds.
- Bottom Navigation:** Runs: 1/1, Errors: 0, Failures: 0, Failure Trace.
- Bottom Tree View:** Test Profilo [Runner: JUnit 5] (31,882 s) (selected), testProfiloErroreSoloPWD (31,882 s).

Code Content (TestProfilo.java):

```
399 @Test
400 public void testProfiloErroreSoloPWD() throws InterruptedException {
401     // Test name: TestProfiloErroreSoloPWD
402     // Step # | name | target | value
403     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |
404     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");
405     Thread.sleep(2000);
406     // 2 | click | name=email |
407     driver.findElement(By.name("email")).click();
408     Thread.sleep(2000);
409     // 3 | type | name=email | f.peluso25@gmail.com
410     driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
411     Thread.sleep(2000);
412     // 4 | click | name=password |
413     driver.findElement(By.name("password")).click();
414     Thread.sleep(2000);
415     // 5 | type | name=password | veloce123
416     driver.findElement(By.name("password")).sendKeys("veloce123");
417     Thread.sleep(2000);
418     // 6 | sendKeys | name=password | ${KEY_ENTER}
419     driver.findElement(By.name("password")).sendKeys(Keys.ENTER);
420     Thread.sleep(2000);
421     // 7 | click | id=dropdownMenuButton |
422     driver.findElement(By.id("dropdownMenuButton")).click();
423     Thread.sleep(2000);
```

The screenshot shows an IDE interface with the following details:

- File Menu:** Help
- Toolbar:** Standard icons for file operations, search, and navigation.
- Code Editor:** The active tab is `TestProfilo.java`. The code is a JUnit test method `testProfiloErroreCarta` that performs the following steps:
 - Open the URL `http://localhost:8080/RAAF-GAMING/servletloginfirst`.
 - Click on the `email` input field and type `f.peluso25@gmail.com`.
 - Click on the `password` input field and type `veloce123`.
 - Press `ENTER` to submit the form.
 - Click on the `dropdownMenuButton`.
- Bottom Status Bar:** Shows "Finished after 46,111 seconds".
- Run Status:** Runs: 1/1, Errors: 0, Failures: 0.
- Test Results:** A tree view shows the test run: `TestProfilo [Runner: JUnit 5] (46,056 s)` and its child `testProfiloErroreCarta (46,056 s)`.
- Failure Trace:** A link to view the failure trace.

1.3 LOGIN

Login eseguita:

The screenshot shows the Eclipse IDE interface with the following details:

- Code Editor:** The main window displays the `TestLogin.java` file containing Java code for a Selenium-based login test.
- Java Code:**

```
93@Test
94 public void testLoginEseguito() throws InterruptedException {
95     // Test name: TestLoginEseguito
96     // Step # | name | target | value
97     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |
98     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");
99     // 2 | click | name=email |
100    Thread.sleep(2000);
101    driver.findElement(By.name("email")).click();
102    Thread.sleep(2000);
103    // 3 | type | name=email | f.peluso25@gmail.com
104    driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
105    Thread.sleep(2000);
106    // 4 | click | name=password |
107    driver.findElement(By.name("password")).click();
108    Thread.sleep(2000);
109    // 5 | type | name=password | veloce123
110    driver.findElement(By.name("password")).sendKeys("veloce123");
111    Thread.sleep(2000);
112    // 6 | click | css=.invio |
113    driver.findElement(By.cssSelector(".invio")).click();
114    Thread.sleep(2000);
115    // 7 | assertTitle | RAAF-GAMING |
116    assertEquals(driver.getTitle(), "RAAF-GAMING");
117    Thread.sleep(2000);
```
- Run Status:** The bottom status bar indicates "Runs: 1/1", "Errors: 0", and "Failures: 0".
- Test Runner:** The bottom left shows the JUnit runner with "TestLogin [Runner: JUnit 5] (17,513 s)" and "testLoginEseguito (17,513 s)".
- Failure Trace:** The bottom right panel is labeled "Failure Trace" but is currently empty.

Login non eseguita:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Help, TestLogin.java (active), Run, Debug, etc.
- Code Editor:** Displays a Java code snippet named `TestLogin.java`. The code is a JUnit test for a login function, containing steps for opening a URL, clicking email and password fields, sending keys, clicking the send button, and asserting an error message. Lines 66-89 are visible.
- Toolbar:** Includes icons for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit, and Synchronize.
- Status Bar:** Shows "Finished after 20,829 seconds".
- Bottom Status Bar:** Shows "Runs: 1/1", "Errors: 0", "Failures: 0".
- Failure Trace:** A panel on the right showing the test results, indicating one failure for the test method `testLoginFallitoPassword`.

The screenshot shows an IDE interface with a code editor and a results panel.

Code Editor:

```
40• @Test
41 public void testLoginFallitoEntrambi() throws InterruptedException {
42     // Test name: TestLoginFallitoEntrambi
43     // Step # | name | target | value | comment
44     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst | |
45     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");
46     Thread.sleep(2000);
47     // 2 | click | name=email | |
48     driver.findElement(By.name("email")).click();
49     Thread.sleep(2000);
50     // 3 | type | name=email | abc@gmail.com |
51     driver.findElement(By.name("email")).sendKeys("abc@gmail.com");
52     Thread.sleep(2000);
53     // 4 | click | name=password | |
54     driver.findElement(By.name("password")).click();
55     Thread.sleep(2000);
56     // 5 | type | name=password | abcdesfg123 |
57     driver.findElement(By.name("password")).sendKeys("abcdesfg123");
58     Thread.sleep(2000);
59     // 6 | click | css=.invio | |
60     driver.findElement(By.cssSelector(".invio")).click();
61     Thread.sleep(2000);
62     // 7 | assertText | name=messaggioerrore | Email/Password errata |
63     assertThat(driver.findElement(By.name("messaggioerrore")).getText(), is("Em"))
64     Thread.sleep(2000);
```

Results Panel:

- Run status: Runs: 1/1, Errors: 0, Failures: 0
- Test summary: TestLogin [Runner: JUnit 5] (17,525 s)
- Failure Trace: (empty)

The code implements a Selenium-based test for a login page. It performs the following steps:

- Open the login page at `http://localhost:8080/RAAF-GAMING/servletloginfirst`.
- Click on the email input field.
- Type `abc@gmail.com` into the email input field.
- Click on the password input field.
- Type `abcdesfg123` into the password input field.
- Click the submit button (css selector `.invio`).
- Assert that the error message "Email/Password errata" is displayed.

1.4 LOGIN GESTORE

Login gestore eseguita:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "RAAF-GAMING".
- Code Editor:** Displays the file `TestRegistrazione.java` containing Java code for a Selenium test. The code performs a series of steps: opening a URL, entering email and password, and clicking a button. It then asserts that a specific error message is present.
- Outline View:** Shows the class structure for `TestLoginGestore`.
- Task List:** Shows three tasks: `testLoginEseguitoOrdine()`, `testLoginEseguitoPresto()`, and `testLoginEseguitoOrdine()`.
- Run View:** Shows the results of the run: 2/2 tests passed, 0 errors, 0 failures.
- System Tray:** Shows the date and time as 10/01/2022 19:13.

Login gestore fallita:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "RAAF-GAMING".
- Code Editor:** Displays the file `TestRegistrazione.java` containing Java code for a Selenium test. The code performs a series of steps: opening a URL, entering email and password, and clicking a button. It then asserts that a specific page title is present.
- Outline View:** Shows the class structure for `TestLoginGestore`.
- Task List:** Shows three tasks: `testLoginFallitoOrdine()`, `testLoginFallitoEntrambiGestori()`, and `testLoginFallitoPasswordProdotto()`.
- Run View:** Shows the results of the run: 3/3 tests passed, 0 errors, 0 failures.
- System Tray:** Shows the date and time as 10/01/2022 19:16.

2. ACQUISTO

2.1. ACQUISTO PRODOTTO

Acquisto prodotto riuscito:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Eclipse IDE, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Code Editor:** The file `TestAcquistoProdotto.java` is open, containing a JUnit test method `testAcquistoDeiProdottiRiuscito`. The code performs a series of actions on a web page, including opening the URL, entering email and password, clicking the "Invio" button, and asserting a success message in an alert.
- Bottom Status Bar:** Finished after 39,169 seconds.
- Run Status:** Runs: 1/1, Errors: 0, Failures: 0.
- Run View:** Shows the test run named `TestAcquistoProdotto [Runner: JUnit 5] (39,114 s)` with one test case `testAcquistoDeiProdottiRiuscito (39,114 s)`.
- Failure Trace:** An empty panel.

Acquisto prodotti non riuscito:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Eclipse IDE, Help.
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and navigation.
- Code Editor:** The active editor window contains the code for `TestAcquistoProdotto.java`. The code is annotated with comments explaining the steps of the test: opening the URL, clicking a button, and asserting an alert message.
- Bottom Status Bar:** Shows "Runs: 1/1", "Errors: 0", and "Failures: 0".
- Bottom Navigation:** Includes links for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (highlighted), and Synchronize.
- Bottom Message:** "Finished after 4,623 seconds".
- Bottom Log:** A table showing the run results:

Test	Time
TestAcquistoProdotto [Runner: JUnit 5]	4,568 s
testAcquistoDeiProdottiNessunProdotto	4,568 s

```
TestAcquistoProdotto.java X
36  }
37  @After
38  public void tearDown() {
39      driver.quit();
40  }
41
42  @Test
43  public void testAcquistoDeiProdottiNessunProdotto() {
44      // Test name: TestAcquistoDeiProdotti_NessunProdotto
45      // Step # | name | target | value
46      // 1 | open | http://localhost:8080/RAAF-GAMING/servletcarrello |
47      driver.get("http://localhost:8080/RAAF-GAMING/servletcarrello");
48      // 2 | click | css=.btn-outline-warning |
49      driver.findElement(By.cssSelector(".btn-outline-warning")).click();
50      // 3 | assertAlert | Non hai prodotti nel carrello |
51      assertThat(driver.switchTo().alert().getText(), is("Non hai prodotti nel carrello"));
52  }
53
54
55
56
57  @Test
58  public void testAcquistoDeiProdottiNonAutenticato() throws InterruptedException {
59      // Test name: TestAcquistoDeiProdotti_NonAutenticato
60      // Step # | name | target | value
61      // 1 | open | http://localhost:8080/RAAF-GAMING/ |
62      driver.get("http://localhost:8080/RAAF-GAMING/");
63      // 2 | click | id=dropdownMenuButton |
64      driver.findElement(By.id("dropdownMenuButton")).click();

```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 41,909 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestAcquistoProdotto [Runner: JUnit 5] (41,852 s)

testAcquistoDeiProdottiIndirizzoDiConsegnaNonValido (41,852 s)

Failure Trace

Eclipse IDE

Help

TestAcquistoProdotto.java

```
57  @Test
58  public void testAcquistoDeiProdottiNonAutenticato() throws InterruptedException {
59      // Test name: TestAcquistoDeiProdotti_NonAutenticato
60      // Step # | name | target | value
61      // 1 | open | http://localhost:8080/RAAF-GAMING/
62      driver.get("http://localhost:8080/RAAF-GAMING/");
63      // 2 | click | id=dropdownMenuButton |
64      driver.findElement(By.id("dropdownMenuButton")).click();
65      Thread.sleep(3000);
66      // 3 | click | linkText=Login |
67      driver.findElement(By.linkText("Login")).click();
68      // 4 | click | name=email |
69      driver.findElement(By.name("email")).click();
70      // 5 | type | name=email | f.peluso25@gmail.com
71      driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
72      // 6 | click | name=password |
73      driver.findElement(By.name("password")).click();
74      // 7 | type | name=password | veloce123
75      driver.findElement(By.name("password")).sendKeys("veloce123");
76      // 8 | click | css=.invio |
77      Thread.sleep(3000);
78      driver.findElement(By.cssSelector(".invio")).click();
79      // 9 | click | css=.fa-user-astronaut |
80      driver.findElement(By.cssSelector(".fa-user-astronaut")).click();
81      // 10 | click | linkText=LogOut |
82      Thread.sleep(3000);
83      driver.findElement(By.linkText("LogOut")).click();
84      Thread.sleep(3000);
85      // 11 | click | css=.row:nth-child(1) li:nth-child(1) .card__title |
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 31,886 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestAcquistoProdotto [Runner: JUnit 5] (31,880 s)
 testAcquistoDeiProdottiNonAutenticato (31,880 s)

Failure Trace

Eclipse IDE

Help

TestAcquistoProdotto.java

```
193  @Test
194  public void testAcquistoDeiProdottiNonDisponibile() throws InterruptedException {
195      // Test name: TestAcquistoDeiProdotti_NonDisponibile
196      // Step # | name | target | value
197      // 1 | open | http://localhost:8080/RAAF-GAMING/servletindex |
198      driver.get("http://localhost:8080/RAAF-GAMING/servletindex");
199      // 2 | click | id=dropdownMenuButton |
200      Thread.sleep(5000);
201      driver.findElement(By.id("dropdownMenuButton")).click();
202      // 3 | click | linkText=Login |
203      driver.findElement(By.linkText("Login")).click();
204      // 4 | click | name=email |
205      Thread.sleep(5000);
206      driver.findElement(By.name("email")).click();
207      // 5 | type | name=email | f.peluso25@gmail.com
208      driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
209      // 6 | click | name=password |
210      driver.findElement(By.name("password")).click();
211      // 7 | type | name=password | veloce123
212      driver.findElement(By.name("password")).sendKeys("veloce123");
213      // 8 | click | css=.invio |
214      Thread.sleep(5000);
215      driver.findElement(By.cssSelector(".invio")).click();
216      // 9 | click | css=.row:nth-child(1) li:nth-child(3) .card_header |
217      driver.findElement(By.cssSelector(".row:nth-child(1) li:nth-child(3) .card_header"));
218      // 10 | click | css=.pl-2 |
219      Thread.sleep(5000);
220      driver.findElement(By.cssSelector(".pl-2")).click();
221      // 11 | alertDialog | Aggiunta nel carrello fatta con successo! |
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 29,504 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestAcquistoProdotto [Runner: JUnit 5] (29,450 s)
 testAcquistoDeiProdottiNonDisponibile (29,450 s)

Failure Trace

2.2

ELIMINAZIONE PRODOTTO

Eliminazione riuscita:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows the Eclipse IDE logo and various menu icons.
- Left Sidebar:** Displays the project structure with "TestEliminazioneProdotto" selected.
- Central Editor:** Contains the source code for `TestEliminazioneProdotto.java`. The code is a JUnit test for deleting a product from a shopping cart. It includes steps for opening the product page, clicking the delete button, accepting an alert, and verifying the cart is empty.
- Bottom Status Bar:** Shows the message "Finished after 22,101 seconds".
- Bottom Navigation:** Includes tabs for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (which is active), and Synchronize.
- Bottom Footer:** Displays run statistics: Runs: 1/1, Errors: 0, Failures: 0, and a green progress bar indicating success.
- Bottom Left:** Shows the test tree with "TestEliminazioneProdotto [Runner: JUnit 5] (22,047 s)" and "testeliminazioneProdottoRiuscito (22,047 s)".
- Bottom Right:** Shows the "Failure Trace" tab.

Eliminazione non riuscita:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Bar:** Shows "Java - Eclipse IDE".
- Editor Area:** Displays the code for `TestEliminazioneProdotto.java`. The code is annotated with JUnit annotations (`@Before`, `@After`, `@Test`) and WebDriver setup.
- Toolbars:** Standard Eclipse toolbars for file operations, search, and navigation.
- Bottom Status Bar:** Shows "Runs: 1/1", "Errors: 0", and "Failures: 0".
- Bottom View Bar:** Includes tabs for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (highlighted), and Synchronize.
- Message Bar:** Shows "Finished after 3.081 seconds".
- Failure Trace:** A collapsed section labeled "Failure Trace" is visible at the bottom right.

```
27 public class TestEliminazioneProdotto {
28     private WebDriver driver;
29     private Map<String, Object> vars;
30     JavascriptExecutor js;
31     @Before
32     public void setUp() {
33         System.setProperty("webdriver.chrome.driver", "test/acquistosistema/chromedriver.exe");
34         driver = new ChromeDriver();
35         js = (JavascriptExecutor) driver;
36         vars = new HashMap<String, Object>();
37     }
38     @After
39     public void tearDown() {
40         driver.quit();
41     }
42     @Test
43     public void testEliminazioneProdottiFallito() {
44         // Test name: TestEliminazioneProdotti_Fallito
45         // Step # | name | target | value
46         // 1 | open | http://localhost:8080/RAAF-GAMING/servletcarrello |
47         driver.get("http://localhost:8080/RAAF-GAMING/servletcarrello");
48         // 2 | assertText | name=NonHaiProdotti | NON HAI NESSUN PRODOTTO NEL CARRELLO!
49         assertThat(driver.findElement(By.name("NonHaiProdotti")).getText(), is("NON HAI NESSUN PRODOTTO NEL CARRELLO!"));
50     }
51     @Test
52     public void testeliminazioneProdottoRiuscito() throws InterruptedException {
53         // Test name: TesteliminazioneProdotto_Riuscito
54         // Step # | name | target | value
55         // 1 | open | http://localhost:8080/RAAF-GAMING/servletprodotto?id=1 |
```

2.3 VISUALIZZAZIONE CARRELLO

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Eclipse IDE, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Left Sidebar:** Project Explorer, Navigator, Outline, Properties, Javadoc, Problems, Synchronizer.
- Central Area:** Code editor showing `TestVisualizzaCarrello.java`. The code is a JUnit test for a shopping cart visualization. It includes steps to open a URL, click on an item, handle an alert, and verify the cart is empty.

```
61
62  @Test
63  public void testVisualizzaCarrelloConElementi() throws InterruptedException {
64      // Test name: TestVisualizzaCarrello_CarrelloConElementi
65      // Step # | name | target | value
66      // 1 | open | http://localhost:8080/RAAF-GAMING/servletprodotto?id=1 |
67      driver.get("http://localhost:8080/RAAF-GAMING/servletprodotto?id=1");
68      // 2 | click | css=.pl-2
69      Thread.sleep(5000);
70      driver.findElement(By.cssSelector(".pl-2")).click();
71      Thread.sleep(5000);
72      // 3 | alertDialog | Aggiunta nel carrello fatta con successo! |
73      Alert alert=driver.switchTo().alert();
74      String s= alert.getText();
75      if(s.equals("Aggiunta nel carrello fatta con successo!"))
76      {
77          alert.accept();
78          Thread.sleep(5000);
79          // 4 | click | id=sostituisciCarrello |
80          driver.findElement(By.id("sostituisciCarrello")).click();
81          Thread.sleep(5000);
82          // 5 | assertNotText | name=NonHaiProdotti | NON HAI NESSUN PRODOTTO NEL CARRELLO!
83          List<WebElement> list = driver.findElements(By.name("NonHaiProdotti"));
84          Assert.assertTrue("NON HAI NESSUN PRODOTTO NEL CARRELLO!", list.size() == 0);
85          //assertThat(driver.findElement(By.name("NonHaiProdotti")).getText(), is(not("NON I
86      }
87  }
88
89 }
```

- Bottom Status Bar:** Runs: 1/1, Errors: 0, Failures: 0.
- Bottom View:** JUnit view showing the test run results.

Eclipse IDE

Help

TestVisualizzaCarrello.java

```
42 public void tearDown() {
43     driver.quit();
44 }
45
46● @Test
47 public void testVisualizzaCarelloCarrelloVuoto() throws InterruptedException {
48     // Test name: TestVisualizzaCarello_CarelloVuoto
49     // Step # | name | target | value
50     // 1 | open | http://localhost:8080/RAAF-GAMING/
51     driver.get("http://localhost:8080/RAAF-GAMING/");
52     // 2 | click | id=sostituisciCarello |
53     Thread.sleep(5000);
54     driver.findElement(By.id("sostituisciCarello")).click();
55     // 3 | assertTitle | CARELLO |
56     assertThat(driver.getTitle(), is("CARELLO"));
57     Thread.sleep(5000);
58     // 4 | assertText | name=NonHaiProdotti | NON HAI NESSUN PRODOTTO NEL CARELLO!
59     assertThat(driver.findElement(By.name("NonHaiProdotti")).getText(), is("NON HAI NESSUN
60 })
61
62● @Test
63 public void testVisualizzaCarelloCarelloConElementi() throws InterruptedException {
64     // Test name: TestVisualizzaCarello_CarelloConElementi
65     // Step # | name | target | value
66     // 1 | open | http://localhost:8080/RAAF-GAMING/servletprodotto?id=1 |
67     driver.get("http://localhost:8080/RAAF-GAMING/servletprodotto?id=1");
68     // 2 | click | css=.pl-2
69     Thread.sleep(5000);
70     driver.findElement(By.cssSelector(".pl-2")).click();
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 14.98 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestVisualizzaCarrello [Runner: JUnit 5] (14,926 s) testVisualizzaCarelloCarrelloVuoto (14,926 s)

Failure Trace

3. Prodotto

3.1 Visualizza Ordine:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Eclipse IDE, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Code Editor:** The file `VisualizzaOrdineTest.java` is open. The code implements a Selenium test for a login and order visualization process. It includes steps for opening the URL, setting window size, clicking on email and password fields, sending keys, and clicking the "Invio" button. A tooltip for the `sendKeys` method is visible, stating: "void org.openqa.selenium.WebElement.sendKeys(CharSequence... keysToSend) Note: This element neither has attached source nor attached Javadoc and hence no Javadoc could be found".
- Bottom Status Bar:** Runs: 1/1, Errors: 0, Failures: 0.
- Bottom View Bar:** Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (highlighted), Synchronize.
- Bottom Log Area:** Shows the message "Finished after 30,372 seconds".
- Bottom Taskbar:** VisualizzaOrdineTest [Runner: JUnit 5] (30,313 s), visualizzaOrdine (30,313 s), Failure Trace.

3.2 GESTIONE ORDINE:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Eclipse IDE, Help.
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and navigation.
- Code Editor:** The main window displays `GestioneOrdineTest.java`. The code is a JUnit test for sending an order. It includes steps for opening a URL, setting window size, clicking input fields, and sending keys. A tooltip for the `By.cssSelector` method is visible, stating: "Note: This element neither has attached source nor attached Javadoc and hence no Javadoc could be found." The code spans lines 74 to 102.
- Bottom Status Bar:** Shows "Runs: 1/1", "Errors: 0", "Failures: 0".
- Bottom Navigation:** Buttons for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (highlighted), and Synchronize.
- Message Bar:** "Finished after 19.481 seconds".
- Bottom Log Area:** Shows the test run results: "GestioneOrdineTest [Runner: JUnit 5] (19,426 s)" and "ordineSpedito (19,426 s)".

```
41  }
42● @Test
43 public void ordineNonSpedito() throws InterruptedException {
44     // Test name: OrdineNonSpedito
45     // Step # | name | target | value
46     // 1 | open | http://localhost:8080/RAAF-GAMING/servletaccessoadmin |
47     driver.get("http://localhost:8080/RAAF-GAMING/servletaccessoadmin");
48
49     driver.findElement(By.id("exampleInputEmail1")).click();
50     // 4 | type | id=exampleInputEmail1 | ordine@admin.com
51     driver.findElement(By.id("exampleInputEmail1")).sendKeys("ordine@admin.com");
52     // 5 | click | id=exampleInputPassword1 |
53     driver.findElement(By.id("exampleInputPassword1")).click();
54     // 6 | type | id=exampleInputPassword1 | veloce123
55     driver.findElement(By.id("exampleInputPassword1")).sendKeys("veloce123");
56     // 7 | click | css=.btn |
57     Thread.sleep(2000);
58     driver.findElement(By.cssSelector(".btn")).click();
59     // 8 | click | id=spedizioneProdotto |
60     Thread.sleep(2000);
61     driver.findElement(By.id("spedizioneProdotto")).click();
62     // 9 | click | id=consegnaOrdine |
63     Thread.sleep(2000);
64     Thread.sleep(2000);
65     driver.findElement(By.id("consegnaOrdine")).sendKeys("06-12-2021");
66     // 11 | click | css=.ml-3 |
67     Thread.sleep(2000);
68     driver.findElement(By.cssSelector(".ml-3")).click();
69     Thread.sleep(5000);

```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 18,354 seconds

Runs: 1/1 Errors: 0 Failures: 0

Failure Trace

3.3 RECENSIONE

Recensione eseguita:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows "Eclipse IDE" and "Help".
- Toolbar:** Includes icons for file operations, search, and navigation.
- Code Editor:** Displays the file "RecensioneTest.java" containing Java code for a Selenium test. The code performs various steps to log in, enter email and password, click the "Invio" button, and click a comment button. It includes several Thread.sleep(1800) calls and driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst") calls.
- Bottom Status Bar:** Shows "Markers", "Properties", "Servers", "Data Source Explorer", "Snippets", "Console", "JUnit X", and "Synchronize".
- Bottom Progress Bar:** Shows "Runs: 1/1", "Errors: 0", "Failures: 0", and a green progress bar.
- Bottom Log Area:** Shows the message "Finished after 25,578 seconds".
- Bottom Task List:** Shows a single task: "RecensioneTest [Runner: JUnit 5] (25,525 s)" with a checkmark and a "recensioneEffettuataConSuccesso (25,525 s)" entry.
- Right Side:** Shows the Eclipse Navigator and Outline panes.

Recensione non eseguita:

```
RecensioneTest.java X
88● @Test
89 public void recensioneGiaEffettuata() throws InterruptedException {
90     // Test name: RecensioneGiaEffettuata
91     // Step # | name | target | value
92     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |
93     Thread.sleep(1800);
94     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");
95     // 2 | setWindowSize | 945x1020 |
96     Thread.sleep(1800);
97     driver.manage().window().setSize(new Dimension(945, 1020));
98     // 3 | click | name=email |
99     Thread.sleep(1800);
100    driver.findElement(By.name("email")).click();
101    // 4 | type | name=email | f.peluso25@gmail.com
102    Thread.sleep(1800);
103    driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
104    // 5 | click | name=password |
105    Thread.sleep(1800);
106    driver.findElement(By.name("password")).click();
107    // 6 | type | name=password | veloce123
108    Thread.sleep(1800);
109    driver.findElement(By.name("password")).sendKeys("veloce123");
110    // 7 | click | css=.invio |
111    Thread.sleep(1800);
112    driver.findElement(By.cssSelector(".invio")).click();
113    // 8 | click | css=.row:nth-child(2) li:nth-child(1) .card__image |
114    Thread.sleep(1800);
115    driver.findElement(By.cssSelector(".row:nth-child(2) li:nth-child(1) .card__image"));
116    // 9 | click | id=commento |
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Runs: 1/1 Errors: 0 Failures: 0

RecensioneTest [Runner: JUnit 5] (28,962 s)

recensioneGiaEffettuata (28,962 s)

Failure Trace

The screenshot shows a Java test script named RecensioneTest.java being run in an IDE. The code is annotated with step numbers and comments describing the actions taken by the test. The test involves navigating to a login page, entering credentials, and clicking a button to perform an action.

```
136• @Test
137 public void recensioneSenzaCommento() throws InterruptedException {
138     // Test name: RecensioneSenzaCommento
139     // Step # | name | target | value
140     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |
141     Thread.sleep(1800);
142     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");
143     // 2 | setWindowSize | 945x1020 |
144     Thread.sleep(1800);
145     driver.manage().window().setSize(new Dimension(945, 1020));
146     // 3 | click | name=email |
147     Thread.sleep(1800);
148     driver.findElement(By.name("email")).click();
149     // 4 | type | name=email | f.peluso25@gmail.com
150     Thread.sleep(1800);
151     driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
152     // 5 | click | name=password |
153     Thread.sleep(1800);
154     driver.findElement(By.name("password")).click();
155     // 6 | type | name=password | veloce123
156     Thread.sleep(1800);
157     driver.findElement(By.name("password")).sendKeys("veloce123");
158     // 7 | click | css=.invio |
159     Thread.sleep(1800);
160     driver.findElement(By.cssSelector(".invio")).click();
161     // 8 | click | css=.row:nth-child(2) li:nth-child(2) .card__title |
162     Thread.sleep(1800);
163     driver.findElement(By.cssSelector(".row:nth-child(2) li:nth-child(2) .card__title"));
164     // 9 | click | id=stellaa7 |
```

Below the code editor, the IDE displays the JUnit runner interface. It shows the test has run 1/1 times with 0 errors and 0 failures. The test method recensioneSenzaCommento has been run 23,284 times. A 'Failure Trace' section is visible on the right side of the runner panel.

IDE Help

RecensioneTest.java X

```
177● @Test
178 public void recensioneSenzaLoggare() throws InterruptedException {
179     // Test name: RecensioneSenzaLoggare
180     // Step # | name | target | value
181     // 1 | open | http://localhost:8080/RAAF-GAMING/servletprodotto?id=9 |
182     Thread.sleep(1800);
183     driver.get("http://localhost:8080/RAAF-GAMING/servletprodotto?id=9");
184     // 2 | setWindowSize | 945x1020 |
185     Thread.sleep(1800);
186     driver.manage().window().setSize(new Dimension(945, 1020));
187     // 3 | click | id=commento |
188     Thread.sleep(1800);
189     driver.findElement(By.id("commento")).click();
190     // 4 | type | id=commento | sdsdsf
191     Thread.sleep(1800);
192     driver.findElement(By.id("commento")).sendKeys("sdsdsf");
193     // 5 | click | id=stella7 |
194     Thread.sleep(1800);
195     driver.findElement(By.id("stella7")).click();
196     // 6 | click | css=.btn-dark |
197     Thread.sleep(1800);
198     driver.findElement(By.cssSelector(".btn-dark")).click();
199     // 7 | assertAlert | Effettua l'accesso per recensire! |
200     Thread.sleep(1800);
201     Alert alert=driver.switchTo().alert();
202     String text=alert.getText();
203     assertEquals("Effettua l'accesso per recensire!",text);
204 }
205
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 16,003 seconds

Runs: 1/1 Errors: 0 Failures: 0

✓ RecensioneTest [Runner: JUnit 5] (15,943 s)
 ✓ recensioneSenzaLoggare (15,943 s)

Failure Trace

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows "Eclipse IDE" and "Help".
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and navigation.
- Left Sidebar:** Shows "RecensioneTest.java X".
- Code Editor:** Displays a Java code snippet for a Selenium test. The code performs various steps: opening a URL, setting window size, clicking on email and password fields, sending keys, and clicking on an "invio" button. It includes several calls to `Thread.sleep(1800);`. A tooltip at the bottom of the editor area says: "Note: This element neither has attached source nor attached Javadoc and hence no Javadoc could be found."
- Bottom Status Bar:** Shows "Runs: 1/1", "Errors: 0", "Failures: 0", and "Press 'F2' for focus".
- Bottom Navigation:** Shows "Markers", "Properties", "Servers", "Data Source Explorer", and "Failure Trace".
- Bottom Log Area:** Shows the log output: "RecensioneTest [Runner: JUnit 5] (25,220 s)" and "recensioneSenzaVoto (25,220 s)".

The screenshot shows an IDE interface with a Java test class named `RecensioneTest.java`. The code is annotated with `@Test` and contains several `Thread.sleep(1800);` statements. It uses `driver.findElement(By.name("email")).click();`, `driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");`, `driver.findElement(By.name("password")).click();`, `driver.findElement(By.name("password")).sendKeys("veloce123");`, and `driver.findElement(By.cssSelector(".invio")).click();`. The test runs successfully with 1/1 run, 0 errors, and 0 failures.

```
206● @Test
207 public void recensioneSenzaVotoECommento() throws InterruptedException {
208     // Test name: RecensioneSenzaVoto
209     // Step # | name | target | value
210     // 1 | open | http://localhost:8080/RAAF-GAMING/servletloginfirst |
211     Thread.sleep(1800);
212     driver.get("http://localhost:8080/RAAF-GAMING/servletloginfirst");
213     // 2 | setWindowSize | 945x1020 |
214     Thread.sleep(1800);
215     driver.manage().window().setSize(new Dimension(945, 1020));
216     // 3 | click | name=email |
217     Thread.sleep(1800);
218     driver.findElement(By.name("email")).click();
219     // 4 | type | name=email | f.peluso25@gmail.com
220     Thread.sleep(1800);
221     driver.findElement(By.name("email")).sendKeys("f.peluso25@gmail.com");
222     // 5 | click | name=password |
223     Thread.sleep(1800);
224     driver.findElement(By.name("password")).click();
225     // 6 | type | name=password | veloce123
226     Thread.sleep(1800);
227     driver.findElement(By.name("password")).sendKeys("veloce123");
228     // 7 | click | css=.invio |
229     Thread.sleep(1800);
230     driver.findElement(By.cssSelector(".invio")).click();
231     // 8 | click | css=.row:nth-child(2) li:nth-child(2) .card__header |
232     Thread.sleep(1800);
233     driver.findElement(By.cssSelector(".row:nth-child(2) li:nth-child(2) .card__header"));
234     // 9 | click | id=commento |
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 21,941 seconds

Runs: 1/1 Errors: 0 Failures: 0

RecensioneTest [Runner: JUnit 5] (21,883 s)
recensioneSenzaVotoECommento (21,883 s)

Failure Trace

3.4 AGGIUNTA AL CARRELLO

Aggiunta al carrello eseguita:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows the Eclipse IDE logo and various icons for file operations, search, and help.
- Left Sidebar:** Displays the package structure with several Java files listed under the package name.
- Central Editor:** The main editor window displays the code for `TestAggiuntaAlCarrello.java`. The code is annotated with comments explaining the steps of the test: opening the product page, clicking the add-to-cart button, and asserting the success message in an alert.
- Bottom Status Bar:** Shows the status "Finished after 12,736 seconds".
- Bottom Navigation:** Includes links for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (which is currently selected), and Synchronize.
- Bottom Progress Bar:** Shows the test results: Runs: 1/1, Errors: 0, Failures: 0.
- Bottom Log View:** Shows the log output for the test run, indicating success: "testAggiuntaAlCarrello [Runner: JUnit 5] (12,683 s)" and "testAggiuntaAlCarrelloProdottoAggiunto (12,683 s)".

Aggiunta al carrello non riuscita:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Eclipse IDE, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Code Editor:** The file `TestAggiuntaAlCarrello.java` is open. The code is a JUnit test for adding an item to a shopping cart. It includes steps to open a URL, click on an item, handle an alert, and click on a shopping cart icon.
- Bottom Bar:** Shows tabs for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, JUnit (selected), and Synchronize.
- Status Bar:** Finished after 45,263 seconds.
- Run View:** Shows 1 run, 0 errors, and 0 failures. The test `testAggiuntaAlCarrelloProdottoGiAggiunto` has a duration of 45,210 s.
- Failure Trace:** An empty panel.

```
56@  @Test
57 public void testAggiuntaAlCarrelloProdottoGiAggiunto() throws InterruptedException {
58     // Test name: TestAggiuntaAlCarrello_ProdottoGiAggiunto
59     // Step # | name | target | value
60     // 1 | open | http://localhost:8080/RAAF-GAMING/
61     driver.get("http://localhost:8080/RAAF-GAMING/");
62     // 2 | click | css=.row:nth-child(1) li:nth-child(1) span:nth-child(1) |
63     Thread.sleep(5000);
64     driver.findElement(By.cssSelector(".row:nth-child(1) li:nth-child(1) span:nth-child(1")));
65     // 3 | click | css=.pl-2 |
66     Thread.sleep(5000);
67     driver.findElement(By.cssSelector(".pl-2")).click();
68     // 4 | alertDialog | Aggiunta nel carrello fatta con successo! |
69     Thread.sleep(5000);
70     Alert alert=driver.switchTo().alert();
71     String t=alert.getText();
72     if(t.equals("Aggiunta nel carrello fatta con successo!"))
73     {
74         alert.accept();
75     }
76     // 5 | click | css=.navbar-brand > img |
77     Thread.sleep(5000);
78     driver.findElement(By.cssSelector(".navbar-brand > img")).click();
79     // 6 | click | css=.row:nth-child(1) li:nth-child(1) .card__title |
80     Thread.sleep(5000);
81     driver.findElement(By.cssSelector(".row:nth-child(1) li:nth-child(1) .card__title"));
82     // 7 | click | css=.fa-shopping-cart |
83     Thread.sleep(5000);
84     driver.findElement(By.cssSelector(".fa-shopping-cart")).click();
```

- Eclipse IDE

Help

TestAggiuntaAlCarrello.java X

```
38 }
39 @After
40 public void tearDown() {
41     driver.quit();
42 }
43 @Test
44 public void testAggiuntaAlCarrelloProdottoNonDisponibile() throws InterruptedException {
45     // Test name: TestAggiuntaAlCarrello_ProdottoNonDisponibile
46     // Step # | name | target | value
47     // 1 | open | http://localhost:8080/RAAF-GAMING/servletprodotto?id=2 |
48     driver.get("http://localhost:8080/RAAF-GAMING/servletprodotto?id=2");
49     // 2 | click | css=.pl-2 |
50     Thread.sleep(5000);
51     driver.findElement(By.cssSelector(".pl-2")).click();
52     // 3 | alertDialog | Prodotto non disponibile in magazzino |
53     assertThat(driver.switchTo().alert().getText(), is("Prodotto non disponibile in magazzino"));
54     Thread.sleep(5000);
55 }
56 @Test
57 public void testAggiuntaAlCarrelloProdottoGiaAggiunto() throws InterruptedException {
58     // Test name: TestAggiuntaAlCarrello_ProdottoGiaAggiunto
59     // Step # | name | target | value
60     // 1 | open | http://localhost:8080/RAAF-GAMING/ |
61     driver.get("http://localhost:8080/RAAF-GAMING/");
62     // 2 | click | css=.row:nth-child(1) li:nth-child(1) span:nth-child(1) |
63     Thread.sleep(5000);
64     driver.findElement(By.cssSelector(".row:nth-child(1) li:nth-child(1) span:nth-child(1)"));
65     // 3 | click | css=.pl-2 |
66     Thread.sleep(5000);
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit Synchronize

Finished after 12,966 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestAggiuntaAlCarrello [Runner: JUnit 5] (12,911 s)

testAggiuntaAlCarrelloProdottoNonDisponibile (12,911 s)

Failure Trace

3.5 RIFORNITURA

Rifornitura eseguita:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "RAAF-GAMING [PROGETTO] Antonio_Di_Lucia". It includes packages like "acquisto", "acquistosistema", "magazzino", "profilo", and "test".
- Code Editor:** Displays a Java code snippet for a test method named "testRifornituraProdottoOK". The code uses WebDriver to interact with a web application at "http://localhost:8080/RAAF-GAMING/servletaccessoadmin". It performs various actions such as clicking input fields and sending keys.
- Outline View:** Shows the class structure with methods like "setUp()", "tearDown()", "testRifornituraProdottoOK()", and "testRifornituraProdottoQuantoOK()".
- Run View:** Shows the results of a JUnit run: "Runs: 1/1", "Errors: 0", "Failures: 0". It lists two tests: "TestRifornituraProdotto [Runned: JUnit 5 (0,748 s)]" and "testRifornituraProdottoOK [Runned: JUnit 5 (0,748 s)]".
- System Tray:** Shows system status including "9°C Nuvoloso" and the date "10/01/2022".

Rifornitura non eseguita:

The screenshot shows two instances of the Eclipse IDE interface, each displaying a Java test class named `TestRifornituraProdotto.java`. The code in both instances is identical, indicating a copy-and-paste operation.

```
70@test
71 public void testRifornituraProdottoQuantitaNo() {
72     // Test name: testRifornituraProdottoQuantitaNo
73     driver.get("http://localhost:8080/RAAF-GAMING/servletaccessoadmin");
74     driver.manage().window().setSize(new Dimension(1050, 616));
75     driver.findElement(By.id("exampleInputEmail1")).click();
76     driver.findElement(By.id("exampleInputPassword1")).sendKeys("veloce123");
77     driver.findElement(By.id("exampleInputPassword1")).sendKeys(Keys.ENTER);
78     driver.findElement(By.cssSelector(".form-row > label:nth-child(2)")).click();
79     // 2 | click | css=.form-row > label:nth-child(2) |
80     driver.findElement(By.name("quantita")).click();
81     // 3 | click | name=quantita |
82     driver.findElement(By.name("quantita")).click();
83     // 4 | type | name=quantita | -20
84     driver.findElement(By.name("quantita")).sendKeys("-20");
85     // 5 | click | css=#idrow0 .btn |
86     driver.findElement(By.cssSelector("#idrow0 .btn")).click();
87     // 6 | assertElementNotPresent | name=succes |
88     {
89         List<WebElement> elements = driver.findElements(By.name("succes"));
90         assertEquals(elements.size() == 0);
91     }
92     // 7 | assertElementNotPresent | name=errore |
93 }
```

The test class contains two methods: `testRifornituraProdottoQuantitaNo()` and `testRifornituraProdottoCapienzaNo()`. Both methods perform the same sequence of actions: navigating to the URL, clicking on input fields, entering specific values, and then checking for the presence of success or error messages. The difference lies in the input values and the assertions made at the end of each method.

Below the code editor, the Eclipse interface shows the following details:

- Project Explorer: Shows the project structure with various Java files and resources.
- Outline and Task List: Show the method definitions and their associated annotations.
- Run Configuration: Shows the current run configuration with a green bar indicating successful execution.
- Console: Displays the output of the test execution.
- Bottom Status Bar: Shows system information like temperature (9°C), battery status (Nuvoloso), and date/time (10/01/2022).

3.6 RICERCA PER CATEGORIA

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "RAAF-GAMING".
- Editor:** Displays two Java files: `TestFormulariNuovoProdotto.java` and `TestRicercaPerCategoria.java`.
- Outline View:** Shows the class structure and methods for `TestRicercaPerCategoria`.
- Task List:** Shows the tasks related to the current file.
- Run View:** Shows the run configuration for the test.
- Console:** Shows the output of the test execution.

```
43     driver.quit();
44 }
45
46 @Test
47 public void testRicercaPerCategoriaOK() {
48     // Test name: testRicercaPerCategoriaOK
49     // Step # | name | target | value | comment
50     // 1 | open | http://localhost:8080/RAAF-GAMING/ |
51     driver.get("http://localhost:8080/RAAF-GAMING/");
52     // 2 | click | css=.navbar-toggler | 
53     driver.findElement(By.cssSelector(".navbar-toggler")).click();
54     // 3 | click | css=.nav-item:nth-child(5) .fa | 
55     driver.findElement(By.cssSelector(".nav-item:nth-child(5) .fa")).click();
56     // 4 | assertText | nome=nomeProdotto | fifa 21 |
57     assertEquals("fifa 21",driver.findElement(By.name("nomeProdotto0")).getText()
58 }
59
60 @Test
61 public void testRicercaPerCategoriaNO() {
62     // Test name: testRicercaPerCategoriaNO
63     // Step # | name | target | value | comment
64     // 1 | open | http://localhost:8080/RAAF-GAMING/ |
65     driver.get("http://localhost:8080/RAAF-GAMING/");
66     // 2 | setWindowSize | 912x600 |
67     driver.manage().window().setSize(new Dimension(912, 600));
68     // 3 | click | .navbar-toggler | 
69     driver.findElement(By.cssSelector(".navbar-toggler")).click();
70
71     // 5 | open | http://localhost:8080/RAAF-GAMING/servletcategorie?per=abc
72     driver.get("http://localhost:8080/RAAF-GAMING/servletcategorie?per=abc");
73     // 6 | assertTitle | RAAF-GAMING |
74     assertEquals(driver.getTitle(), "RAAF-GAMING");
75 }
76 }
77
```



This screenshot is identical to the ones above, showing the Eclipse IDE interface with the same project structure, files, and test results.

3.7 RICERCA PER NOME

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Displays the project structure for "RAAF-GAMING [PROGETTO]".
- TestRicercaPerNome.java:** The active code editor window contains the following Java code:

```
78     }
79
80 }
81
82 @Test
83 public void testRicercaPerNomeNO() {
84     // Test name: testRicercaPerNomeNO
85     // Step # | name | target | value | comment
86     // 1 | open | http://localhost:8080/RAAF-GAMING/ |
87     driver.get("http://localhost:8080/RAAF-GAMING/");
88     // 2 | setWindowSize | 912x600 |
89     driver.manage().window().setSize(new Dimension(912, 600));
90     // 3 | click | name=ricerca |
91     driver.findElement(By.name("ricerca")).click();
92     // 4 | type | name=ricerca | q |
93     driver.findElement(By.name("ricerca")).sendKeys("q");
94     // 5 | sendKeys | name=ricerca | ${KEY_ENTER} |
95     driver.findElement(By.name("ricerca")).sendKeys(Keys.ENTER);
96     // 6 | assertElementNotPresent | name=prodottoCard |
97     List<WebElement> elements = driver.findElements(By.name("prodottoCard"));
98     assertEquals(elements.size(), 0);
99 }
100
```

- Outline:** Shows the class structure with methods: `testRicercaPerNomeNO()`, `testRicercaPerNomeCompletoOK()`, and `testRicercaPerNomeOK()`.
- Task List:** Shows the tasks available for the current file.
- Markers:** Shows 1 run, 0 errors, and 0 failures.
- Servers:** Shows the configuration for the RAAF-GAMING project.
- Bottom Status Bar:** Displays the system tray icons, battery level (9%), and date/time (10/01/2022).

The screenshot shows two instances of the Eclipse IDE interface, each displaying a Java test class named `TestRicercaPerNome.java`. The top instance shows the class `testRicercaPerNomeParziale()`, and the bottom instance shows the class `testRicercaPerNomeCompletoOK()`. Both classes are part of the `RAAF-GAMING (PROGETTO Antonio_De_Lucia)` project. The code uses Selenium WebDriver to interact with a web application running at `http://localhost:8080/RAAF-GAMING`. The tests involve clicking a search button, sending keys to the search input field, and asserting the presence of specific text elements. The bottom instance also includes a `tearDown()` method that performs a `driver.quit();` operation.

```
60
61@Test
62 public void testRicercaPerNomeParziale() {
63     // Test name: testRicercaPerNomeParziale
64     // Step # | name | target | value | comment
65     // 1 | open | http://localhost:8080/RAAF-GAMING/ |
66     driver.get("http://localhost:8080/RAAF-GAMING/");
67     // 2 | click | name=ricerca | |
68     driver.findElement(By.name("ricerca")).click();
69     // 3 | type | name=ricerca | ass |
70     driver.findElement(By.name("ricerca")).sendKeys("ass");
71     // 4 | sendKeys | name=ricerca | ${KEY_ENTER} |
72     driver.findElement(By.name("ricerca")).sendKeys(Keys.ENTER);
73     // 5 | assertText | name=nomeProdotto | ass |
74     for(int i=0;i<3;i++) {
75         String testo= driver.findElement(By.name("nomeProdotto"+i)).getText();
76         System.out.println(testo);
77         assertEquals("Non contiene il testo!!", testo.contains("ass"));
78     }
79
80 }
82@Test
83 public void testRicercaPerNomeOK() {
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145@Test
146 public void testRicercaPerNomeCompletoOK() {
147     // Test name: testRicercaPerNomeOK
148     // Step # | name | target | value | comment
149     // 1 | open | http://localhost:8080/RAAF-GAMING/ |
150     driver.get("http://localhost:8080/RAAF-GAMING/");
151     // 2 | click | name=ricerca | |
152     driver.findElement(By.name("ricerca")).click();
153     // 3 | type | name=ricerca | the last of us 2 |
154     driver.findElement(By.name("ricerca")).sendKeys("the last of us 2");
155     // 4 | sendKeys | name=ricerca | ${KEY_ENTER} |
156     driver.findElement(By.name("ricerca")).sendKeys(Keys.ENTER);
157     // 5 | assertText | name=nomeProdotto | the last of us 2 |
158     assertEquals("the last of us 2",driver.findElement(By.name("nomeProdotto0"))
159 }
160
161@Test
162 public void testRicercaPerNomeParziale() {
163     // Test name: testRicercaPerNomeParziale
164     // Step # | name | target | value | comment
165     // 1 | open | http://localhost:8080/RAAF-GAMING/ |
166     driver.get("http://localhost:8080/RAAF-GAMING/");
167     // 2 | click | name=ricerca | |
168     driver.findElement(By.name("ricerca")).click();
169     // 3 | type | name=ricerca | ass |
170     driver.findElement(By.name("ricerca")).sendKeys("ass");
171     // 4 | sendKeys | name=ricerca | ${KEY_ENTER} |
172     driver.findElement(By.name("ricerca")).sendKeys(Keys.ENTER);
173     // 5 | assertText | name=nomeProdotto | ass |
174     for(int i=0;i<3;i++) {
175         String testo= driver.findElement(By.name("nomeProdotto"+i)).getText();
176         System.out.println(testo);
177         assertEquals("Non contiene il testo!!", testo.contains("ass"));
178     }
179
180 }
```

3.8 FORNITURA NUOVO PRODOTTO

Runs: 22/22

Errors: 0

- ▼  TestFornituraNuovoProdotto [Runner: JUnit 5] (373,828 s)
 -  testFornituraNuovoProdottoNomeNO (15,172 s)
 -  testFornituraNuovoProdottoPEGINO (8,041 s)
 -  testFornituraNuovoProdottoVKEYNO (26,831 s)
 -  testFornituraNuovoProdottoScontoNO (13,504 s)
 -  testFornituraNuovoVideogiocoFisicoMagazzinoNO (19,809 s)
 -  testFornituraNuovoProdottoConsoleOK (13,935 s)
 -  testFornituraNuovoProdottoDescrizioneDlcNO (18,713 s)
 -  testFornituraNuovoProdottoCodiceAbbonamentoNO (18,625 s)
 -  testFornituraNuovoProdottoQuantitaNO (14,196 s)
 -  testFornituraNuovoProdottoDigitaleOK (14,416 s)
 -  testFornituraNuovoProdottoColoreConsoleNO (18,537 s)
 -  testFornituraNuovoProdottoNCDVideogiocoNO (14,240 s)
 -  testFornituraNuovoProdottoSpecificaconsoleNO (20,156 s)
 -  testFornituraNuovoProdottoCopertinaNO (22,574 s)
 -  testFornituraNuovoProdottoDurataAbbonamentoNO (18,542 s)
 -  testFornituraNuovoProdottoGestoreNO (7,349 s)
 -  testFornituraNuovoProdottoDimensioneVideogiocoNO (14,152 s)
 -  testFornituraNuovoProdottoDlcOK (18,706 s)
 -  testFornituraNuovoProdottoAbbonamentoOK (18,878 s)
 -  testFornituraNuovoProdottoPrezzoNO (18,620 s)
 -  testFornituraNuovoProdottoDimensioneDlcNO (19,105 s)
 -  testFornituraNuovoVideogiocoFisicoOK (19,726 s)