

Università degli Studi di Salerno

Corso di laurea in Informatica

**Ingegneria, Gestione ed Evoluzione
del Software**

“DisneyStore2.0”

Docente:

Andrea De Lucia

Studenti:

Buono Claudia

05225 01097

Fulgione Anna

05225 01103

Anno accademico 2020/2021

SOMMARIO

TEST EXECUTION.....	5
INTRODUZIONE	5
JUNIT.....	5
TEST DI UNITÀ MODEL.....	5
<i>AdminModel</i>	5
<i>AddressModel</i>	6
<i>FatturaModel</i>	6
<i>ProductModel</i>	7
<i>RecensioneModel</i>	7
<i>RegistreredModel</i>	8
<i>UserModel</i>	8
TEST DI UNITÀ SERVLET.....	9
<i>AddressOperations</i>	9
<i>AdminCatalog</i>	9
<i>AdminManager</i>	10
<i>Billing</i>	10
<i>CartServlet</i>	11
<i>Checkout</i>	11
<i>ConsumerCatalog</i>	12
<i>IndexServlet</i>	12
<i>Login</i>	13
<i>Logout</i>	13
<i>ProductAdminController</i>	14
<i>ProductControl</i>	14
<i>RecensioneServlet</i>	15
<i>Registration</i>	15
<i>Suggerer</i>	16
<i>SuggererUser</i>	16
<i>UserManager</i>	17
<i>VisualizzaIndirizzi</i>	17
TEST DI INTEGRAZIONE	18
<i>AddressOperationsIntegration</i>	18
<i>AdminCatalogIntegration</i>	18
<i>AdminManagerIntegration</i>	19
<i>BillingIntegration</i>	19
<i>CartServletIntegration</i>	20
<i>CheckoutIntegration</i>	20
<i>ConsumerCatalogIntegration</i>	21
<i>IndexServletIntegration</i>	21
<i>LoginIntegration</i>	22
<i>LogoutIntegration</i>	22
<i>ProductAdminControllerIntegration</i>	23
<i>ProductControlIntegration</i>	23
<i>RecensioneServletIntegration</i>	24
<i>RegistrationIntegration</i>	24
<i>SuggererIntegration</i>	25

<i>SuggesterUserIntegration</i>	25
<i>UserManagerIntegration</i>	26
<i>VisualizzaIndirizzilIntegration</i>	26
KATALON.....	27
TEST DI SISTEMA GESTIONE UTENTE	27
<i>Login</i>	27
<i>Registrazione</i>	27
<i>Cambio Nome</i>	27
<i>Cambio Cognome</i>	28
TEST DI SISTEMA GESTIONE PRODOTTO.....	28
<i>Aggiunta recensione</i>	28
<i>Modifica prodotto</i>	28
<i>Aggiungi Prodotto</i>	28
TEST DI SISTEMA GESTIONE INDIRIZZO	29
<i>Aggiungi Indirizzo</i>	29
TEST DI SISTEMA GESTIONE ORDINE	29
<i>Pagamento</i>	29

TEST EXECUTION

INTRODUZIONE

Tale documento riguarda l'esecuzione dei test di unità, integrazione e sistema dopo che sono state effettuate le modifiche. Il documento si compone di due parti, nella prima parte effettueremo test di unità sulle classi Model e sulle Servlet, successivamente procederemo con il test di integrazione. Tali test verranno effettuati con JUnit. La seconda parte invece rappresenta l'attività di esecuzione dei test case descritti all'interno del Test Case Specification. I test verranno eseguiti con Katalon.

JUNIT

TEST DI UNITÀ MODEL

ADMINMODEL

The screenshot shows the IntelliJ IDEA interface with the following details:

- Code Editor:** The main window displays `TC_AdminModel.java` containing Java code for testing the `AdminModel`. The code includes a `setup()` method and a `@Test` method named `doRetrieveByKey()`.
- Project Structure:** The left sidebar shows the project structure under the `test` directory, specifically the `model` package which contains several test classes like `TC_AdminModel`, `TC_AdresModel`, etc.
- Run Results:** The bottom panel shows the `Run` tool window with the results of the test run. It indicates "Tests passed: 1 of 1 test - 736 ms". The result for `TC_AdminModel` shows a green checkmark and the duration `736 ms`. The command used was `C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...`.

ADDRESSMODEL

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it is the project structure, which includes a 'test' folder containing 'integration' and 'model' sub-folders. Under 'model', there are several test classes: TC_AdminModel, TC_AdressModel, TC_FatturaModel, TC_ProductModel, TC_RecessioneModel, TC_RegisteredModel, and TC_UserModel. The 'TC_AdressModel.java' file is open in the editor, showing Java code with annotations like @Test and assertions like assertEquals. The 'Run' tool window at the bottom shows the results of a run named 'TC_AdressModel'. It indicates 'Tests passed: 4' and provides a detailed log of the test execution, including the command 'C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...' and the output of the tests, which all passed quickly.

FATTURAMODEL

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it is the project structure, which includes a 'test' folder containing 'integration' and 'model' sub-folders. Under 'model', there are several test classes: TC_AdminModel, TC_AdressModel, TC_FatturaModel, TC_ProductModel, TC_RecessioneModel, TC_RegisteredModel, and TC_UserModel. The 'TC_FatturaModel.java' file is open in the editor, showing Java code with annotations like @Test and assertions like assertEquals. The 'Run' tool window at the bottom shows the results of a run named 'TC_FatturaModel'. It indicates 'Tests passed: 7' and provides a detailed log of the test execution, including the command 'C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...' and the output of the tests, which all passed quickly.

PRODUCTMODEL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The 'test' tab is selected in the 'Project' tool window, showing the 'model' package which contains several test classes like TC_ProductModel, TC_FatturaModel, and TC_Paginatore. The code editor displays a portion of the TC_ProductModel.java file with several test methods. The 'Run' tool window at the bottom shows the test results for 'TC_ProductModel' with 9 tests passed in 1s 967 ms. The terminal output shows the execution command and the printed result of a test case.

```
void testDoRetrieveByKey() throws Exception {
    ProductBean prodotto= new ProductBean( id: 93);
    prodotto.setDiscount(99.99);
    prodotto.setIva(99.99);
    prodotto.setPrice(99.99);
    prodotto.setQty(99);
    Boolean flag = product.doUpdate(prodotto);
    assertEquals( expected: true, flag);
}
```

```
Process finished with exit code 0
```

RECENSIONEMODEL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The 'test' tab is selected in the 'Project' tool window, showing the 'model' package which contains several test classes like TC_RecensioneModel, TC_AdressModel, and TC_FatturaModel. The code editor displays a portion of the TC_RecensioneModel.java file with three test methods. The 'Run' tool window at the bottom shows the test results for 'TC_RecensioneModel' with 3 tests passed in 1s 202 ms. The terminal output shows the execution command and the printed result of a test case.

```
@Test
void testMediumVote () throws SQLException {
    double media= recensione.mediumVote(getprodotto());
    assertEquals( expected: 4.0, media);
}

@Test
void testUserComment()throws SQLException {
    RecensioneBean recensioneBean= recensione.userComment(getUtente(),getprodotto());
    RecensioneBean oracolo= new RecensioneBean( description: "Prodotto curato nei minimi dettagli", valutazione: 4, nome: "Riccardo");
    assertEquals(oracolo, recensioneBean);
}

@Test
void testGetComments()throws SQLException{
    List<RecensioneBean> listRecensione= recensione.getComments (getprodotto());
    List<RecensioneBean> oracolo= new ArrayList<>();
    RecensioneBean r1= new RecensioneBean( description: "Buona qualita e consegna in tempo", valutazione: 4, nome: "Riccardo");
    RecensioneBean r2= new RecensioneBean( description: "Prodotto curato nei minimi dettagli", valutazione: 4, nome: "Riccardo");
    assertEquals(oracolo, listRecensione);
}
```

```
Process finished with exit code 0
```

REGISTEREDMODEL

```
a-Disney > src > test > model > TC_RegisteredModel
private UserBean getUser(){
    UserBean utente= new Registered( name: "rosalia", cognome: "capozzolo", login: "rosalia@libero.it", password: "rosalia");
    return utente;
}

@Test
void testDoesExist() throws SQLException{
    Boolean flag= userReg.doesExist(getUser());
    assertEquals( expected: true,flag);
}

@Test
void testLogin () throws SQLException{
    UserBean utente= userReg.login( username: "rosalia@libero.it", password: "rosalia");
    assertEquals( expected: "rosalia@libero.it",utente.getLogin());
}
```

Run: TC_UserModel > Tests passed: 2 of 2 tests – 1s 157 ms
Test Results > TC_UserModel > testLogin() > testDoesExist()
Process finished with exit code 0

USERMODEL

```
a-Disney > src > test > model > TC_RegisteredModel
assertEquals( expected: "rosalia@libero.it", registered.getLogin());

@Test
void testRegistration()throws SQLException{
    Boolean flag= registrato.registration(getUtente());
    assertEquals( expected: true, flag);

}

@Test
void testDoModify()throws SQLException{
    Boolean flag= registrato.doModify(getUtente(), name: "leo", surname: "fulgione", login: "bleo110@gmail.com");
    assertEquals( expected: true, flag);

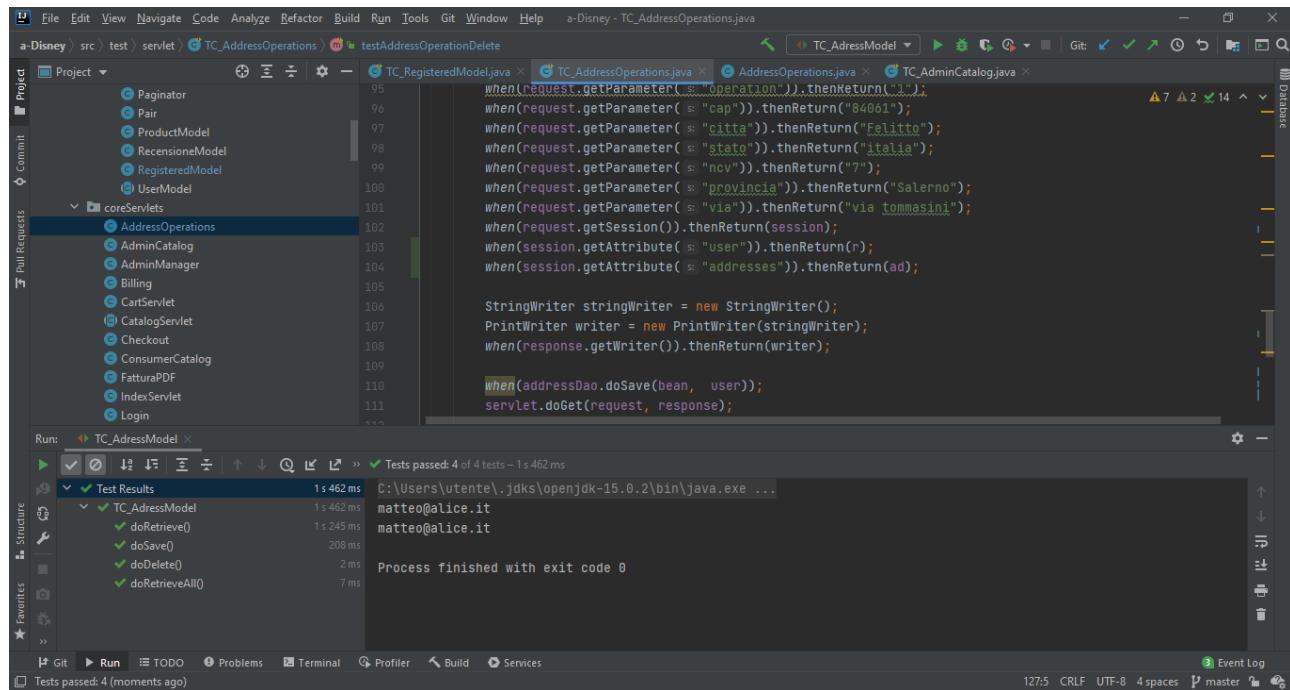
}

@Test
void testDoRetrieveBySearch() throws SQLException{
    ArrayList<Registered> listaRegistrati= registrato.doRetrieveBySearch("anna@virgilio.it");
    assertNotNull(listaRegistrati);
}
```

Run: TC_RegisteredModel > Tests passed: 4 of 4 tests – 1s 466 ms
Test Results > TC_RegisteredModel > testDoRetrieveByKey() > testRegistration() > testDoRetrieveBySearch() > testDoModify()
Process finished with exit code 0

TEST DI UNITÀ SERVLET

ADDRESSOPERATIONS



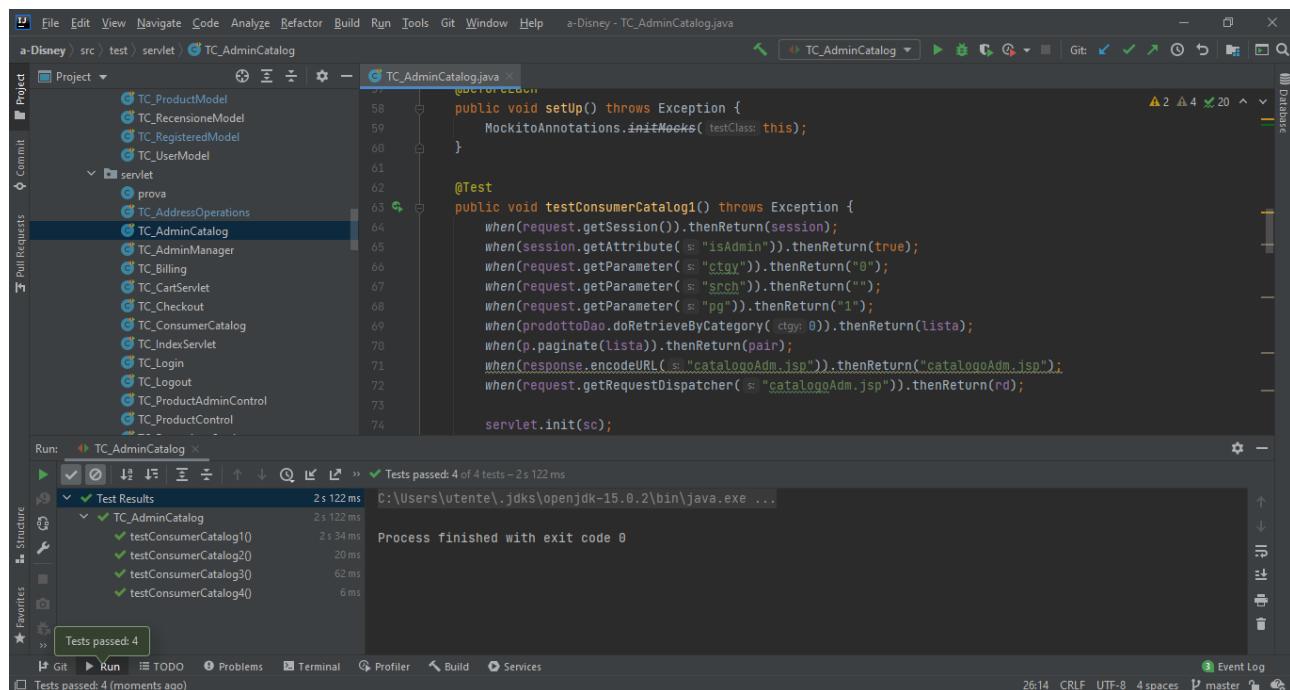
The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays a Java test class for AddressOperations. The run tool window at the bottom shows the test results for TC_AdressModel, indicating 4 tests passed in 1s 462 ms. The command line output shows the results and ends with "Process finished with exit code 0".

```
when(request.getParameter("operation")).thenReturn("j");
when(request.getParameter("cap")).thenReturn("84061");
when(request.getParameter("citta")).thenReturn("Felitto");
when(request.getParameter("stato")).thenReturn("Italia");
when(request.getParameter("ncv")).thenReturn("7");
when(request.getParameter("provincia")).thenReturn("Salerno");
when(request.getParameter("via")).thenReturn("via Tommasini");
when(session.getAttribute("user")).thenReturn(r);
when(session.getAttribute("addresses")).thenReturn(ad);

StringWriter stringWriter = new StringWriter();
PrintWriter writer = new PrintWriter(stringWriter);
when(response.getWriter()).thenReturn(writer);

when(addressDao.doSave(bean, user));
servlet doGet(request, response);
```

ADMINCATALOG



The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays a Java test class for AdminCatalog. The run tool window at the bottom shows the test results for TC_AdminCatalog, indicating 4 tests passed in 2s 122 ms. The command line output shows the results and ends with "Process finished with exit code 0".

```
MockitoAnnotations.initMocks(testClass);
}

@Test
public void testConsumerCatalog1() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute("isAdmin")).thenReturn(true);
    when(request.getParameter("ctgy")).thenReturn("0");
    when(request.getParameter("srch")).thenReturn("");
    when(request.getParameter("pg")).thenReturn("1");
    when(prodottoDao.doRetrieveByCategory(ctgy)).thenReturn(lista);
    when(p.paginate(lista)).thenReturn(pair);
    when(response.encodeURL("catalogoAdm.jsp")).thenReturn("catalogoAdm.jsp");
    when(request.getRequestDispatcher("catalogoAdm.jsp")).thenReturn(rd);
    servlet.init(sc);
}
```

ADMINMANAGER

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_AdminManager.java` containing test code for the `TC_AdminManager` class. The test uses the `Mockito` framework to verify interactions with `fatturaDao`, `paginator`, and `response`. The test passes with a duration of 2ms.

```
when(request.getParameter("da")).thenReturn("2020-8-1");
when(request.getParameter("a")).thenReturn("2020-8-4");
when(request.getParameter("pg")).thenReturn("1");
when(fatturaDao.retrieveInvoices(da, a)).thenReturn(list);
when(fatturaDao.retrieveInvoices(r, da, a)).thenReturn(list);
when(paginator.paginate(list)).thenReturn(pair1);
when(response.encodeURL("Ordini.jsp")).thenReturn("/Ordini.jsp");
when(request.getRequestDispatcher("/Ordini.jsp")).thenReturn(rd);

ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
servlet.doGet(request, response);
verify(rd).forward(request, response);
verify(response).encodeURL(captor.capture());
assertEquals(expected: "/Ordini.jsp", captor.getValue());
```

Test Results: 1 of 1 test - 2ms

TC_AdminManager
testAdminManager() 2ms

Process finished with exit code 0

BILLING

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_Billing.java` containing test code for the `TC_Billing` class. The test uses the `Mockito` framework to verify interactions with `request`, `response`, and `session`. The test passes with a duration of 1s 633ms.

```
verify(response).encodeURL(captor.capture());
assertEquals(expected: "Login.jsp", captor.getValue());
```

```
@Test
public void testBilling2() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(request.getSession().getAttribute("cart")).thenReturn(cart);
    when(request.getSession().getAttribute("user")).thenReturn(user);
    when(request.getRequestDispatcher("AddressOperations?operation=0")).thenReturn(rd1);
    when(response.encodeURL("Checkout.jsp")).thenReturn("Checkout.jsp");
    when(request.getRequestDispatcher("Checkout.jsp")).thenReturn(rd2);

    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    servlet doGet(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "Checkout.jsp", captor.getValue());}
```

Test Results: 2 of 2 tests - 1s 633 ms

TC_Billing
testBilling10 1s 623 ms
testBilling20 10 ms

Process finished with exit code 0

CARTSERVLET

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the `TC_CartServlet.java` file, which contains Java code for a servlet. The run tool window at the bottom shows a single test named `testCartServlet()` has passed, taking 2s 247 ms. The terminal output shows "Process finished with exit code 0".

```
package test.servlet;
import ...
public class TC_CartServlet{
    @Mock
    HttpServletRequest request;
    @Mock
    HttpServletResponse response;
    @Mock
    HttpSession session;
    @Mock
    ProductModel prodottoDao;
```

CHECKOUT

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the `TC_Checkout.java` file, which contains Java code for a servlet. The run tool window at the bottom shows two tests named `testCheckout1()` and `testCheckout2()` have passed, taking 2s 384 ms and 1s 542 ms respectively. The terminal output shows "Process finished with exit code 0".

```
package test.servlet;
import ...
public class TC_Checkout {
    @Mock
    HttpServletRequest request;
    @Mock
    HttpServletResponse response;
    @Mock
    FatturaModel fatturaDao;
    @Mock
    FatturaBean fatturaBean;
    @Mock
```

CONSUMERCATALOG

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The left sidebar displays the project structure with a tree view of classes under 'src/test/servlet'. The main editor window shows the code for 'TC_ConsumerCatalog.java'. The code is annotated with JUnit annotations like @Test and @Mock. The run tool window at the bottom shows a successful run of the tests, with 3 tests passed in 2 seconds and 72 milliseconds.

```
a-Disney > src > test > servlet > TC_ConsumerCatalog.java

@Project Project TC_ConsumerCatalog.java

@Mock
public void testConsumerCatalog1() throws Exception {
    when(request.getParameter("ctgy")).thenReturn("0");
    when(request.getParameter("srch")).thenReturn("");
    when(request.getParameter("pg")).thenReturn("1");
    when(productoDao.doRetrieveByCategory(ctgy)).thenReturn(lista);
    when(p.paginate(lista)).thenReturn(pain);
    when(response.encodeURL("Catalogo.jsp")).thenReturn("Catalogo.jsp");
    when(request.getRequestDispatcher("Catalogo.jsp")).thenReturn(rd);

    servlet.init(sc);
    servlet doGet(request, response);
    verify(rd).forward(request, response);
    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "Catalogo.jsp", captor.getValue());
}

@Mock
public void testConsumerCatalog2() throws Exception {
    when(request.getParameter("ctgy")).thenReturn("0");
    when(request.getParameter("srch")).thenReturn("");
    when(request.getParameter("pg")).thenReturn("1");
    when(productoDao.doRetrieveByCategory(ctgy)).thenReturn(lista);
    when(p.paginate(lista)).thenReturn(pain);
    when(response.encodeURL("Catalogo.jsp")).thenReturn("Catalogo.jsp");
    when(request.getRequestDispatcher("Catalogo.jsp")).thenReturn(rd);

    servlet.init(sc);
    servlet doGet(request, response);
    verify(rd).forward(request, response);
    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "Catalogo.jsp", captor.getValue());
}

@Mock
public void testConsumerCatalog3() throws Exception {
    when(request.getParameter("ctgy")).thenReturn("0");
    when(request.getParameter("srch")).thenReturn("");
    when(request.getParameter("pg")).thenReturn("1");
    when(productoDao.doRetrieveByCategory(ctgy)).thenReturn(lista);
    when(p.paginate(lista)).thenReturn(pain);
    when(response.encodeURL("Catalogo.jsp")).thenReturn("Catalogo.jsp");
    when(request.getRequestDispatcher("Catalogo.jsp")).thenReturn(rd);

    servlet.init(sc);
    servlet doGet(request, response);
    verify(rd).forward(request, response);
    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "Catalogo.jsp", captor.getValue());
}
```

Run: TC_ConsumerCatalog x
Test Results
TC_ConsumerCatalog
testConsumerCatalog1() 2 s 72 ms
testConsumerCatalog2() 2 s 24 ms
testConsumerCatalog3() 16 ms
Process finished with exit code 0

Tests passed: 3 (moments ago)

INDEXSERVLET

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The left sidebar displays the project structure with a tree view of classes under 'src/test/servlet'. The main editor window shows the code for 'TC_IndexServlet.java'. The code is annotated with JUnit annotations like @Test and @Mock. The run tool window at the bottom shows a successful run of the tests, with 2 tests passed in 2 seconds and 132 milliseconds.

```
a-Disney > src > test > servlet > TC_IndexServlet.java

@Project Project TC_IndexServlet.java

@Mock
public void testIndexServlet() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(productDao.doRetrieveByDiscount(amount: 30, acquista: true)).thenReturn(list);

    assertThrows(Exception.class, ()>servlet doPost(request, response));
}

@Mock //TCS
public void testIndexServlet2() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(productDao.doRetrieveByDiscount(amount: 30, acquista: true)).thenReturn(list);
    when(request.getRequestDispatcher("/contentJSP/ProductCard.jsp")).thenReturn(rd);

    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    servlet doPost(request, response);
    verify(request).getRequestDispatcher(captor.capture());
    assertEquals(expected: "/contentJSP/ProductCard.jsp", captor.getValue());
}
```

Run: TC_IndexServlet x
Test Results
TC_IndexServlet
testIndexServlet() 2 s 132 ms
testIndexServlet2() 2 s 104 ms
Process finished with exit code 0

Tests passed: 2 (moments ago)

LOGIN

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_Login.java` containing several test methods using Mockito annotations. The run tool window at the bottom shows the results of a run named 'TC_Login'. It lists 5 tests all passed, with a total execution time of 2s 168ms. The terminal output shows the command used to run the tests and the message 'Process finished with exit code 0'.

```
a-Disney > src > test > servlet > TC_Login.java

public void testLogin1() throws Exception {
    when(request.getParameter("username")).thenReturn("rosa@libero.it");
    when(request.getParameter("password")).thenReturn("rosa1234");
    when(request.getSession()).thenReturn(session);
    when(userDao.login(username: "rosa@libero.it", password: "rosa")).thenReturn(userBean);

    assertThrows(Exception.class, ()-> servlet.doPost(request, response));
}

public void testLogin2() throws Exception {
    when(request.getParameter("username")).thenReturn("");
    when(request.getParameter("password")).thenReturn("rosa1234");
    when(request.getSession()).thenReturn(session);
    when(userDao.login(username: "", password: "rosa")).thenReturn(userBean);
}
```

Run: TC_Login >

Test Results

- TC_Login
 - testLogin1() 2s 108 ms
 - testLogin2() 10 ms
 - testLogin3() 34 ms
 - testLogin4() 9 ms
 - testLogin5() 7 ms

Process finished with exit code 0

Tests passed: 5 (moments ago)

LOGOUT

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_Logout.java` containing a single test method. The run tool window at the bottom shows the results of a run named 'TC_Logout'. It lists 1 test passed, with a total execution time of 1s 236ms. The terminal output shows the command used to run the tests and the message 'Process finished with exit code 0'.

```
a-Disney > src > test > servlet > TC_Logout.java

public void setUp() throws Exception {
    MockitoAnnotations.initMocks(testClass: this);
}

public void testLogout() throws Exception {
    when(request.getSession()).thenReturn(session);

    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet doGet(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "index.jsp", captor.getValue());
}
```

Run: TC_Logout >

Test Results

- TC_Logout
 - testLogout() 1s 236 ms

Process finished with exit code 0

Tests passed: 1 (moments ago)

PRODUCTADMINCONTROL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_ProductAdminController.java` containing a test method `testPAC26()`. The test uses the Mockito library to simulate various parameters and session attributes. The test results in the bottom panel show 5 tests passed in 2 seconds.

```
@Test //TCS I cambi sono vuoti
public void testPAC26() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute("isAdmin")).thenReturn(true);
    when(request.getParameter("act")).thenReturn("insert");
    when(request.getParameter("name")).thenReturn("");
    when(request.getParameter("description")).thenReturn("");
    when(request.getParameter("personaggio")).thenReturn("Aurora");
    when(request.getParameter("url")).thenReturn("images/auroracuscino.PNG");
    when(request.getParameter("tipo")).thenReturn("cuscino");
    when(request.getParameter("categoria")).thenReturn("0");
    when(request.getParameter("qty")).thenReturn("3");
    when(request.getParameter("price")).thenReturn("");
    when(request.getParameter("iva")).thenReturn("5.00");
    when(request.getParameter("sconto")).thenReturn("35.00");
}
```

Run: TC_ProductAdminController

Tests Results: 5 Tests passed: 5 of 5 tests – 2 s 794 ms

Test	Time	Result
✓ TC_ProductAdminController	2 s 794 ms	35.0
✓ testPACDelete()	2 s 576 ms	7.00
✓ testPAC24()	21 ms	5.0
✓ testPAC25()	10 ms	ProductBean [qty=3, code=0, photo=images/auroracuscino3.PNG, name=Cuscino, description=Cuscino Aurora, character=Aurora]
✓ testPAC26()	13 ms	
✓ testPAC27()	174 ms	

Process finished with exit code 0

PRODUCTCONTROL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_ProductControl.java` containing a test method `testAddC()`. The test uses the Mockito library to simulate session attributes and request parameters. The test results in the bottom panel show 3 tests passed in 2 seconds.

```
@Test //TCS
public void testAddC() throws Exception {
    when(request.getSession()).thenReturn(session);
    Cart c = new Cart();
    when((Cart)request.getSession().getAttribute("cart")).thenReturn(c);
    when(request.getParameter("act")).thenReturn("addC");
    when(request.getParameter("id")).thenReturn("3");
    when(request.getParameter("qty")).thenReturn("2");

    StringWriter stringWriter = new StringWriter();
    PrintWriter writer = new PrintWriter(stringWriter);
    when(response.getWriter()).thenReturn(writer);
    ProductBean b = new ProductBean();
    when(productDao.doRetrieveByKey( code: 3,  acquistabile: true)).thenReturn(b);

    servlet doGet(request, response);
}
```

Run: TC_ProductControl

Tests Results: 3 Tests passed: 3 of 3 tests – 2 s 283 ms

Test	Time	Result
✓ TC_ProductControl	2 s 283 ms	
✓ testAddC()	2 s 257 ms	
✓ testView()	12 ms	
✓ testDelete()	14 ms	

Process finished with exit code 0

RECENSIONESERVLET

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Path:** a-Disney > src > test > servlet > TC_RecensioneServlet
- Code Editor:** The code for `TC_RecensioneServlet.java` is displayed. It contains a test method `testRecensioneView()` which asserts that the response string equals the expected value.
- Run Tab:** Shows the results of the run: "Tests passed: 2 of 2 tests – 2 s 633 ms". The test `testRecensioneInsert()` took 2 s 594 ms and `testRecensioneView()` took 39 ms.
- Project Tree:** Shows various test classes like `TC_Billing`, `TC_CartServlet`, etc., under the `servlet` package.
- Status Bar:** Shows "Tests passed: 2 (moments ago)" and "Event Log" with "master" selected.

REGISTRATION

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Path:** a-Disney > src > test > servlet > TC_Registration
- Code Editor:** The code for `TC_Registration.java` is displayed. It uses Mockito annotations to mock request and response objects for a registration test.
- Run Tab:** Shows the results of the run: "Tests passed: 1 of 1 test – 2 s 644 ms". The test `testRegistration6()` took 2 s 644 ms.
- Project Tree:** Shows various test classes like `TC_Administrator`, `TC_Billing`, etc., under the `servlet` package.
- Status Bar:** Shows "Tests passed: 1 (moments ago)" and "Event Log" with "master" selected.

SUGGESTER

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the `TC_Suggerster.java` file, which contains Java code for testing a suggerster service. The test method `testSuggerster()` uses the Mockito framework to simulate a search request and verify the response. The test passes with a duration of 208 ms.

```
53     }
54
55     @Test //TCS
56     public void testSuggerster() throws Exception {
57         when(request.getParameter( :: "srch")).thenReturn("rosalia@libero.com");
58         when(model.doRetrieveBySearch( search: "rosalia@libero.com",  acquistabile: true)).thenReturn(list);
59
60         StringWriter stringWriter = new StringWriter();
61         PrintWriter writer = new PrintWriter(stringWriter);
62         when(response.getWriter()).thenReturn(writer);
63         servlet.init();
64         servlet.service(request, response);
65         String result = stringWriter.getBuffer().toString().trim();
66         assertNotNull(result);
67     }
68
69 }
70 }
```

Run: TC_Suggerster > Tests Results > TC_Suggerster > testSuggerster
Tests passed: 1 of 1 test - 2s 208 ms
Process finished with exit code 0

SUGGESTERUSER

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the `TC_SuggersterUser.java` file, which contains Java code for testing a suggerster user service. The test method `testSuggersterUser()` uses the Mockito framework to simulate a search request and verify the response. The test passes with a duration of 378 ms.

```
56
57
58     @Test //TCS
59     public void testSuggersterUser() throws Exception {
60         when(request.getParameter( :: "srch")).thenReturn("rosalia@libero.it");
61         when(request.getSession()).thenReturn(session);
62         when(registeredDao.doRetrieveBySearch("rosalia@libero.it")).thenReturn(list);
63
64         StringWriter stringWriter = new StringWriter();
65         PrintWriter writer = new PrintWriter(stringWriter);
66         when(response.getWriter()).thenReturn(writer);
67         servlet.init();
68         servlet.service(request, response);
69         String result = stringWriter.getBuffer().toString().trim();
70         assertNotNull(result);
71     }
72 }
```

Run: TC_SuggersterUser > Tests Results > TC_SuggersterUser > testSuggersterUser
Tests passed: 1 of 1 test - 2s 378 ms
Process finished with exit code 0

USERMANAGER

The screenshot shows the IntelliJ IDEA interface for the 'USERMANAGER' project. The code editor displays the `TC_UserManager.java` file, which contains a JUnit test method `test1()`. The test uses Mockito annotations to simulate session and parameter values, then asserts equality between the expected and actual values of `RegisteredModel` attributes. The test passes with a duration of 2s 489ms.

```
@Test //TCS5
public void test1() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(request.getParameter("op")).thenReturn("modCred");
    when(request.getSession().getAttribute("user")).thenReturn(userBean);
    when(request.getParameter("name")).thenReturn("rosalia");
    when(request.getParameter("cognome")).thenReturn("capozzolo");
    when(request.getParameter("login")).thenReturn("rosalia@libero.it");
    when(request.getParameter("pass")).thenReturn("rosalia");

    servlet.doPost(request, response);
    RegisteredModel registeredDao= new RegisteredModel();
    Registered registrato= registeredDao.doRetrieveByKey( user: "rosalia@libero.it");
    assertEquals( expected: "rosalia",registrato.getName());
    assertEquals( expected: "rosalia",registrato.getPassword());
    assertEquals( expected: "capozzolo",registrato.getCognome());
}
```

The run results show 2 tests passed in 2s 489ms. The terminal output shows the process finished with exit code 0. The bottom status bar indicates the code was saved at 27:14 and is on the master branch.

VISUALIZZAINDIRIZZI

The screenshot shows the IntelliJ IDEA interface for the 'VISUALIZZAINDIRIZZI' project. The code editor displays the `TC_Visualizzaindirizzi.java` file, which contains a JUnit test method `testConsumerCatalog1()`. The test uses Mockito annotations to simulate session and attribute values, then asserts equality between the expected and actual URLs. The test passes with a duration of 2s 224ms.

```
public void setUp() throws Exception {
    MockitoAnnotations.initMocks( testClass: this);
}

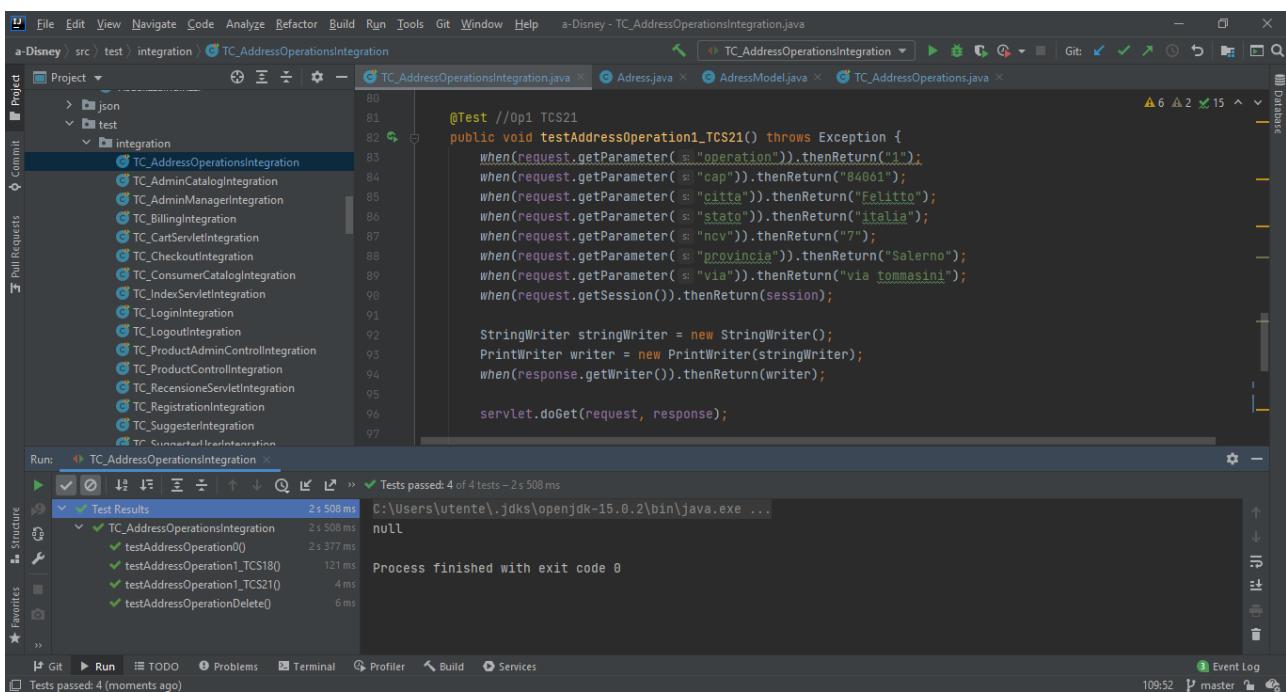
@Test
public void testConsumerCatalog1() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute( s: "user")).thenReturn(user);
    when(response.encodeURL( s: "/protected-page.jsp")).thenReturn("/protected-page.jsp");
    when(request.getRequestDispatcher( s: "/protected-page.jsp")).thenReturn(rd);

    servlet doGet(request, response);
    verify(rd).forward(request, response);
    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    verify(response).encodeURL(captor.capture());
    assertEquals( expected: "/protected-page.jsp",captor.getValue());
}
```

The run results show 1 test passed in 2s 224ms. The terminal output shows the process finished with exit code 0. The bottom status bar indicates the code was saved at 62:42 and is on the master branch.

TEST DI INTEGRAZIONE

ADDRESSOPERATIONSINTEGRATION



The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it, the code editor displays a Java file named `TC_AddressOperationsIntegration.java`. The code contains several test methods for an integration test class. The run tool window at the bottom shows the test results for `TC_AddressOperationsIntegration`, indicating 4 tests passed in 2 seconds and 508 ms. The log output shows the command `C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...` followed by `null`. The status bar at the bottom right shows the time as 10:52 and the branch as master.

```
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
```

```
@Test //Op1 TCS21
public void testAddressOperation1_TCS21() throws Exception {
    when(request.getParameter("operation")).thenReturn("1");
    when(request.getParameter("cap")).thenReturn("84061");
    when(request.getParameter("citta")).thenReturn("Felitto");
    when(request.getParameter("stato")).thenReturn("Italia");
    when(request.getParameter("ncv")).thenReturn("7");
    when(request.getParameter("provincia")).thenReturn("Salerno");
    when(request.getParameter("via")).thenReturn("via Tommasini");
    when(request.getSession()).thenReturn(session);

    StringWriter stringWriter = new StringWriter();
    PrintWriter writer = new PrintWriter(stringWriter);
    when(response.getWriter()).thenReturn(writer);

    servlet doGet(request, response);
}
```

Run: TC_AddressOperationsIntegration x

Test Results: 2 s 508 ms

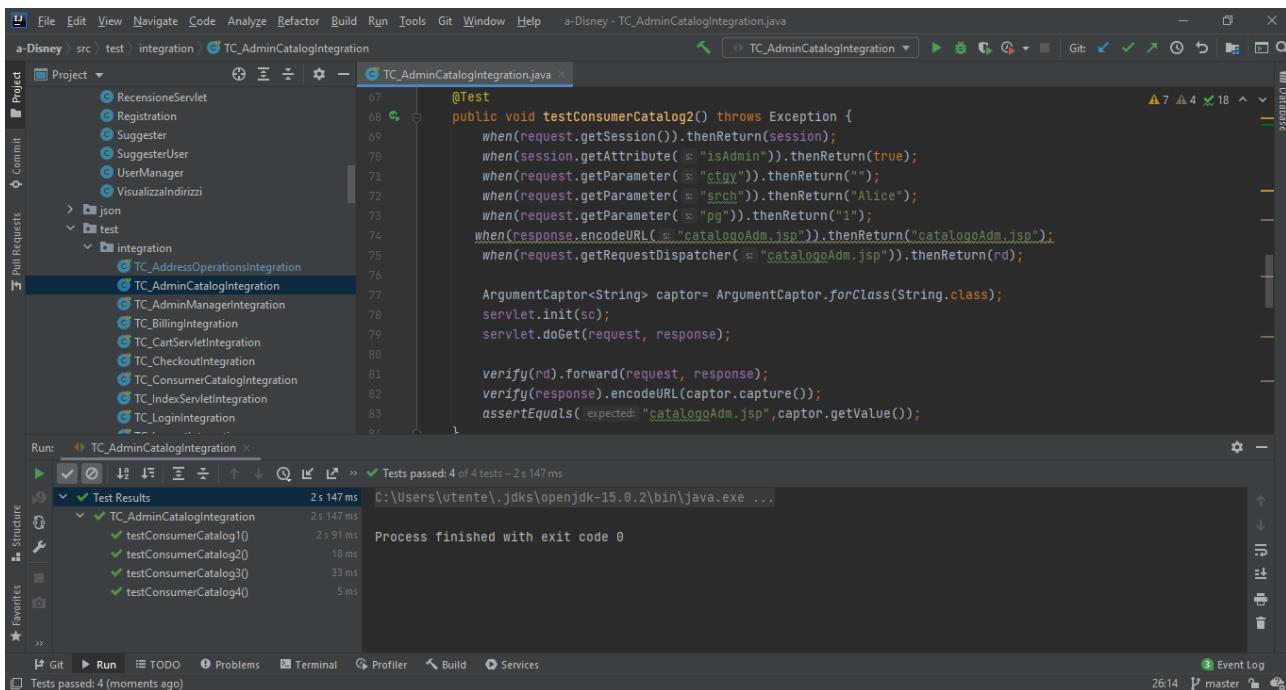
- TC_AddressOperationsIntegration
 - testAddressOperation0 2 s 508 ms
 - testAddressOperation1_TCS180 121 ms
 - testAddressOperation1_TCS210 4 ms
 - testAddressOperationDelete 6 ms

Process Finished with exit code 0

Git Run TODO Problems Terminal Profiler Build Services

Tests passed: 4 (moments ago)

ADMINCATALOGINTEGRATION



The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it, the code editor displays a Java file named `TC_AdminCatalogIntegration.java`. The code contains several test methods for an integration test class. The run tool window at the bottom shows the test results for `TC_AdminCatalogIntegration`, indicating 4 tests passed in 2 seconds and 147 ms. The log output shows the command `C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...`. The status bar at the bottom right shows the time as 26:14 and the branch as master.

```
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
```

```
@Test
public void testConsumerCatalog2() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute("isAdmin")).thenReturn(true);
    when(request.getParameter("ctgy")).thenReturn("");
    when(request.getParameter("srch")).thenReturn("Alice");
    when(request.getParameter("pg")).thenReturn("1");
    when(response.encodeURL("catalogoAdm.jsp")).thenReturn("catalogoAdm.jsp");
    when(request.getRequestDispatcher("catalogoAdm.jsp")).thenReturn(rd);

    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet.init(sc);
    servlet doGet(request, response);

    verify(rd).forward(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected, "catalogoAdm.jsp", captor.getValue());
}
```

Run: TC_AdminCatalogIntegration x

Test Results: 2 s 147 ms

- TC_AdminCatalogIntegration
 - testConsumerCatalog1() 2 s 91 ms
 - testConsumerCatalog2() 18 ms
 - testConsumerCatalog3() 33 ms
 - testConsumerCatalog4() 5 ms

Process finished with exit code 0

Git Run TODO Problems Terminal Profiler Build Services

Tests passed: 4 (moments ago)

ADMINMANAGERINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the file 'TC_AdminManagerIntegration.java'. The class contains a single test method, `testAdminManager()`, which uses the Mockito framework to simulate session and request attributes. The test passes quickly, taking 2 seconds and 37 milliseconds. The run results show one test passed.

```
a-Disney > src > test > integration > TC_AdminManagerIntegration.java
@Text
public void testAdminManager() throws Exception
{
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute( :: "isAdmin")).thenReturn(true);
    when(request.getParameter( :: "search")).thenReturn("rosalia@libero.it");
    when(request.getParameter( :: "da")).thenReturn("2020-8-1");
    when(request.getParameter( :: "a")).thenReturn("2020-8-4");
    when(request.getParameter( :: "pg")).thenReturn("1");
    when(response.encodeURL( :: "/Ordini.jsp")).thenReturn("/Ordini.jsp");
    when(request.getRequestDispatcher( :: "/Ordini.jsp")).thenReturn(rd);

    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet doGet(request, response);
    verify(rd).forward(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals( expected: "/Ordini.jsp", captor.getValue());
}
```

Run: TC_AdminManagerIntegration > Test Results: 2 s 37 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
Process finished with exit code 0

BILLINGINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the file 'TC_BillingIntegration.java'. It contains two test methods, `testBilling1()` and `testBilling2()`. The first test, `testBilling1()`, involves creating a cart, adding a product, and then checking session attributes. The second test, `testBilling2()`, involves similar steps. Both tests pass quickly, taking 1 second and 301 milliseconds. The run results show two tests passed.

```
a-Disney > src > test > integration > TC_BillingIntegration.java
@Text
public void testBilling1() throws Exception {
    Cart c= new Cart();
    ProductBean prodotto= new ProductBean();
    prodotto.setCode(3);
    c.addOrder(new Order(prodotto, quantity: 4));
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute( :: "cart")).thenReturn(c);
    when(session.getAttribute( :: "user")).thenReturn(null);
    when(response.encodeURL( :: "Login.jsp")).thenReturn("Login.jsp");

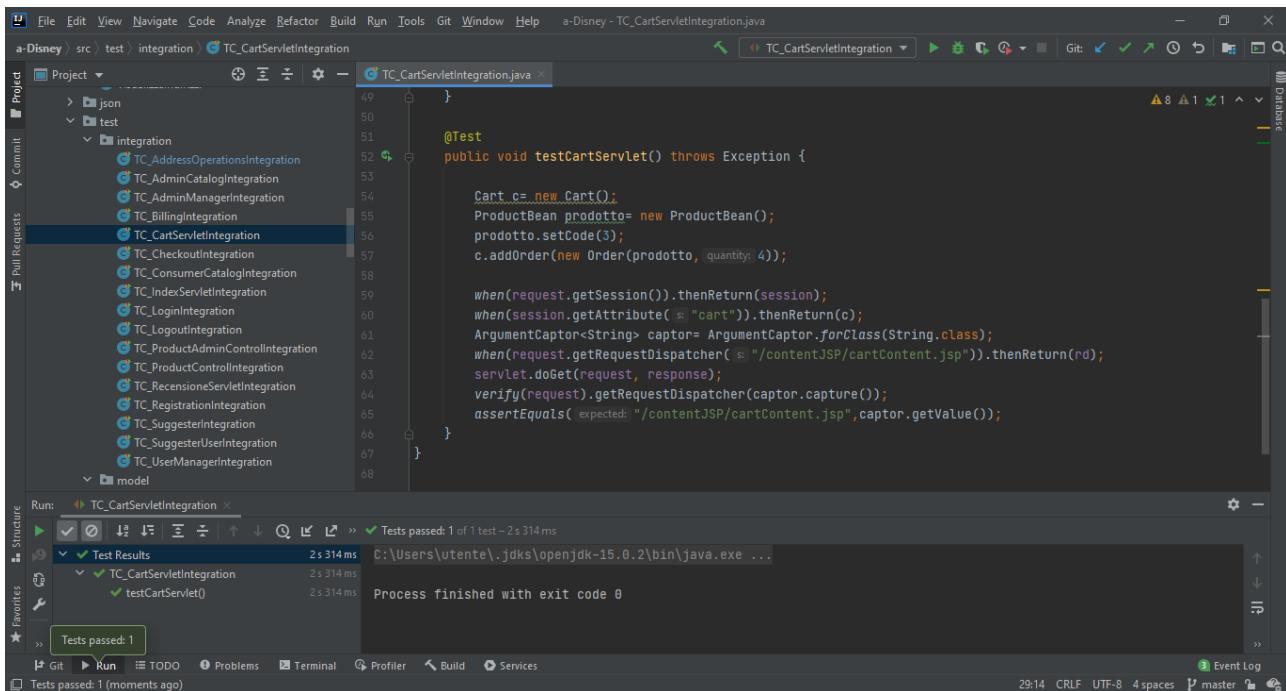
    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet doGet(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals( expected: "Login.jsp", captor.getValue());
}

@Text
public void testBilling2() throws Exception {
    Cart c= new Cart();
    ProductBean prodotto= new ProductBean();
    prodotto.setCode(3);
    c.addOrder(new Order(prodotto, quantity: 4));
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute( :: "cart")).thenReturn(c);
    when(session.getAttribute( :: "user")).thenReturn(null);
    when(response.encodeURL( :: "Login.jsp")).thenReturn("Login.jsp");

    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet doGet(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals( expected: "Login.jsp", captor.getValue());
}
```

Run: TC_BillingIntegration > Test Results: 1 s 310 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
Process finished with exit code 0

CARTSERVLETINTEGRATION

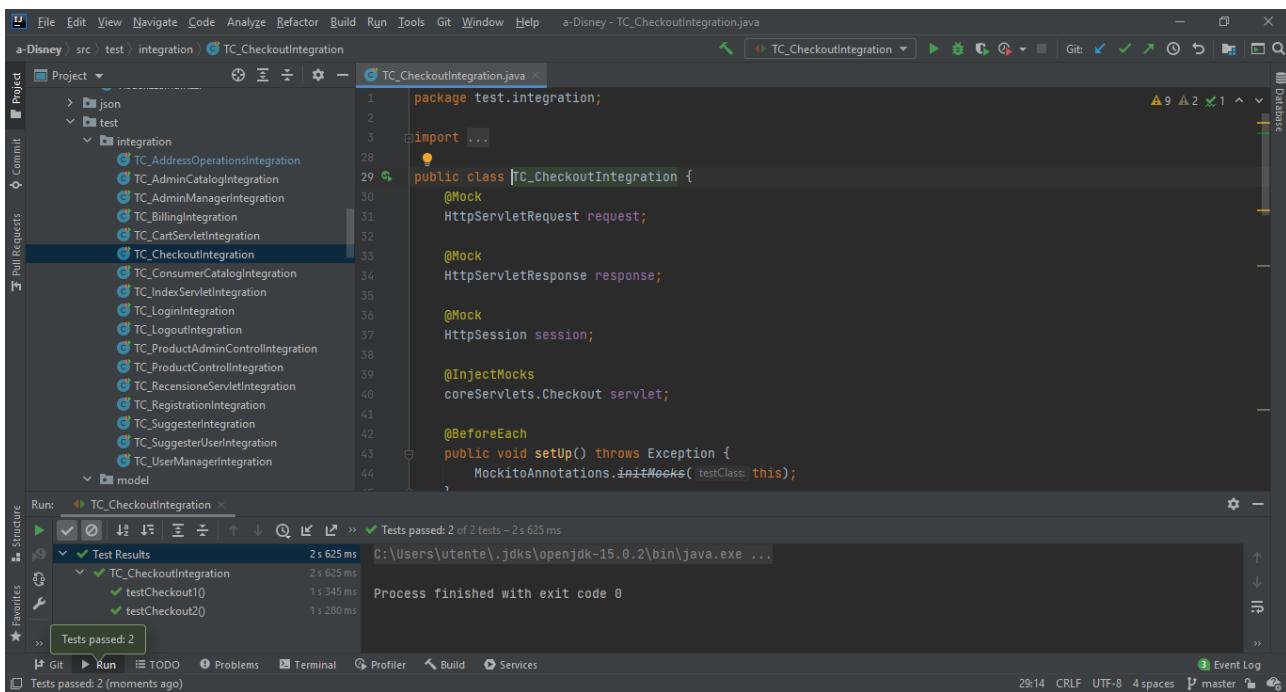


The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_CartServletIntegration.java`, which contains a single test method `testCartServlet()`. The test sets up a `Cart` object, adds a product, and then checks the session attribute `cart` to verify it contains the expected product. The run tool window at the bottom shows the test passed in 2s 314ms.

```
a-Disney > src > test > integration > TC_CartServletIntegration.java
...
@Test
public void testCartServlet() throws Exception {
    Cart c = new Cart();
    ProductBean prodotto= new ProductBean();
    prodotto.setCode(3);
    c.addOrder(new Order(prodotto, quantity: 4));
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute( "cart")).thenReturn(c);
    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    when(request.getRequestDispatcher( "/contentJSP/cartContent.jsp")).thenReturn(rd);
    servlet doGet(request, response);
    verify(request).getRequestDispatcher(captor.capture());
    assertEquals( expected: "/contentJSP/cartContent.jsp",captor.getValue());
}
}

Run: TC_CartServletIntegration > Test Results
  ✓ Test Results 2s 314 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
    ✓ TC_CartServletIntegration 2s 314 ms
      ✓ testCartServlet() 2s 314 ms
Process finished with exit code 0
Tests passed: 1 (moments ago)
```

CHECKOUTINTEGRATION



The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_CheckoutIntegration.java`, which contains a test class `TC_CheckoutIntegration` with two methods: `testCheckout()` and `testCheckout2()`. The test uses Mockito to mock `HttpServletRequest`, `HttpServletResponse`, and `HttpSession`, and injects a `coreServlets.Checkout` servlet. The run tool window at the bottom shows both tests passing in 2s 625ms.

```
a-Disney > src > test > integration > TC_CheckoutIntegration.java
...
package test.integration;
import ...
public class TC_CheckoutIntegration {
    @Mock
    HttpServletRequest request;
    @Mock
    HttpServletResponse response;
    @Mock
    HttpSession session;
    @InjectMocks
    coreServlets.Checkout servlet;
    @BeforeEach
    public void setUp() throws Exception {
        MockitoAnnotations.initMocks( this);
    }
}

Run: TC_CheckoutIntegration > Test Results
  ✓ Test Results 2s 625 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
    ✓ TC_CheckoutIntegration 2s 625 ms
      ✓ testCheckout() 1s 345 ms
      ✓ testCheckout2() 1s 280 ms
Process finished with exit code 0
Tests passed: 2 (moments ago)
```

CONSUMERCATALOGINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_ConsumerCatalogIntegration.java`. The code contains two test methods: `testConsumerCatalog1()` and `testConsumerCatalog2()`. The `testConsumerCatalog1()` method uses Mockito annotations to verify parameters and responses for a Catalogo.jsp page. The `testConsumerCatalog2()` method also uses Mockito annotations to verify a request dispatcher. The run tool window at the bottom shows the results of the test run, indicating 3 tests passed in 2s 509ms.

```
a-Disney > src > test > integration > TC_ConsumerCatalogIntegration.java
@Text
public void testConsumerCatalog1() throws Exception {
    when(request.getParameter("ctgy")).thenReturn("0");
    when(request.getParameter("srch")).thenReturn("");
    when(request.getParameter("pg")).thenReturn("1");
    when(response.encodeURL("Catalogo.jsp")).thenReturn("Catalogo.jsp");
    when(request.getRequestDispatcher("Catalogo.jsp")).thenReturn(rd);

    servlet.init(sc);
    servlet doGet(request, response);
    verify(rd).forward(request, response);
    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "Catalogo.jsp", captor.getValue());
}

@Text
public void testConsumerCatalog2() throws Exception {
```

Run: TC_ConsumerCatalogIntegration

Test Results: 2s 509 ms

- TC_ConsumerCatalogIntegration 2s 509 ms
 - testConsumerCatalog1() 2s 448 ms
 - testConsumerCatalog2() 15 ms
 - testConsumerCatalog3() 46 ms

Process finished with exit code 0

INDEXSERVLETINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_IndexServletIntegration.java`. The code contains two test methods: `testIndexServlet()` and `testIndexServlet2()`. The `testIndexServlet()` method uses Mockito annotations to verify session and doPost methods. The `testIndexServlet2()` method also uses Mockito annotations to verify a request dispatcher. The run tool window at the bottom shows the results of the test run, indicating 2 tests passed in 2s 73ms.

```
a-Disney > src > test > integration > TC_IndexServletIntegration.java
MockitoAnnotations.initMocks(testClass: this);

@Text //TCS
public void testIndexServlet() throws Exception {
    when(request.getSession()).thenReturn(session);
    assertThrows(Exception.class, () -> servlet.doPost(request, response));
}

@Text //TCS
public void testIndexServlet2() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(request.getRequestDispatcher("/contentJSP/ProductCard.jsp")).thenReturn(rd);

    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    servlet.doPost(request, response);
    verify(request).getRequestDispatcher(captor.capture());
    assertEquals(expected: "/contentJSP/ProductCard.jsp", captor.getValue());
}
```

Run: TC_IndexServletIntegration

Test Results: 2s 73 ms

- TC_IndexServletIntegration 2s 73 ms
 - testIndexServlet() 2s 48 ms
 - testIndexServlet2() 25 ms

Process finished with exit code 0

LOGININTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The main area displays the code for `TC_LoginIntegration.java`. The code contains two test methods: `testLogin1()` and `testLogin2()`, both annotated with `@Test`. The `testLogin1()` method uses `MockitoAnnotations.initMocks(this);` and `when(request.getParameter("username")).thenReturn("rosa@libero.it");` to simulate a login attempt. The `testLogin2()` method does the same but with an empty username. Both tests assert that an exception is thrown if the session is null. Below the code editor, the "Run" tab is selected, showing the test results: "Tests passed: 5". The results table shows five green checkmarks for each test method, with execution times ranging from 7ms to 238ms.

```
45     @Test //TCS1
46     public void testLogin1() throws Exception {
47         when(request.getParameter( s: "username")).thenReturn("rosa@libero.it");
48         when(request.getParameter( s: "password")).thenReturn("rosa1234");
49         when(request.getSession()).thenReturn(session);
50
51         assertThrows(Exception.class, ()->servlet.doPost(request, response));
52     }
53
54     @Test //TCS2
55     public void testLogin2() throws Exception {
56         when(request.getParameter( s: "username")).thenReturn("");
57         when(request.getParameter( s: "password")).thenReturn("rosa1234");
58         when(request.getSession()).thenReturn(session);
59
60         assertThrows(Exception.class, ()->servlet.doPost(request, response));
61     }
```

Test	Time
TC_LoginIntegration	2s 238 ms
testLogin1()	2s 238 ms
testLogin2()	7 ms
testLogin3()	21 ms
testLogin4()	12 ms
testLogin5()	14 ms

LOGOUTINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The main area displays the code for `TC_LogoutIntegration.java`. The code contains one test method: `testLogout()`, annotated with `@Test`. This method sets up the environment by calling `MockitoAnnotations.initMocks(this);`. It then simulates a logout request by returning the session and capturing the URL. The test asserts that the response URL is "index.jsp". Below the code editor, the "Run" tab is selected, showing the test results: "Tests passed: 1". The results table shows one green checkmark for the `testLogout()` method, with an execution time of 1s 405 ms.

```
47     public void setUp() throws Exception {
48         MockitoAnnotations.initMocks( testClass: this );
49     }
50
51     @Test //TCS
52     public void testLogout() throws Exception {
53         when(request.getSession()).thenReturn(session);
54
55         ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
56         servlet.doGet(request, response);
57         verify(response).encodeURL(captor.capture());
58         assertEquals( expected: "index.jsp", captor.getValue() );
59     }
60
61 }
```

Test	Time
TC_LogoutIntegration	1s 405 ms
testLogout()	1s 405 ms

PRODUCTADMINCONTROLINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the following details:

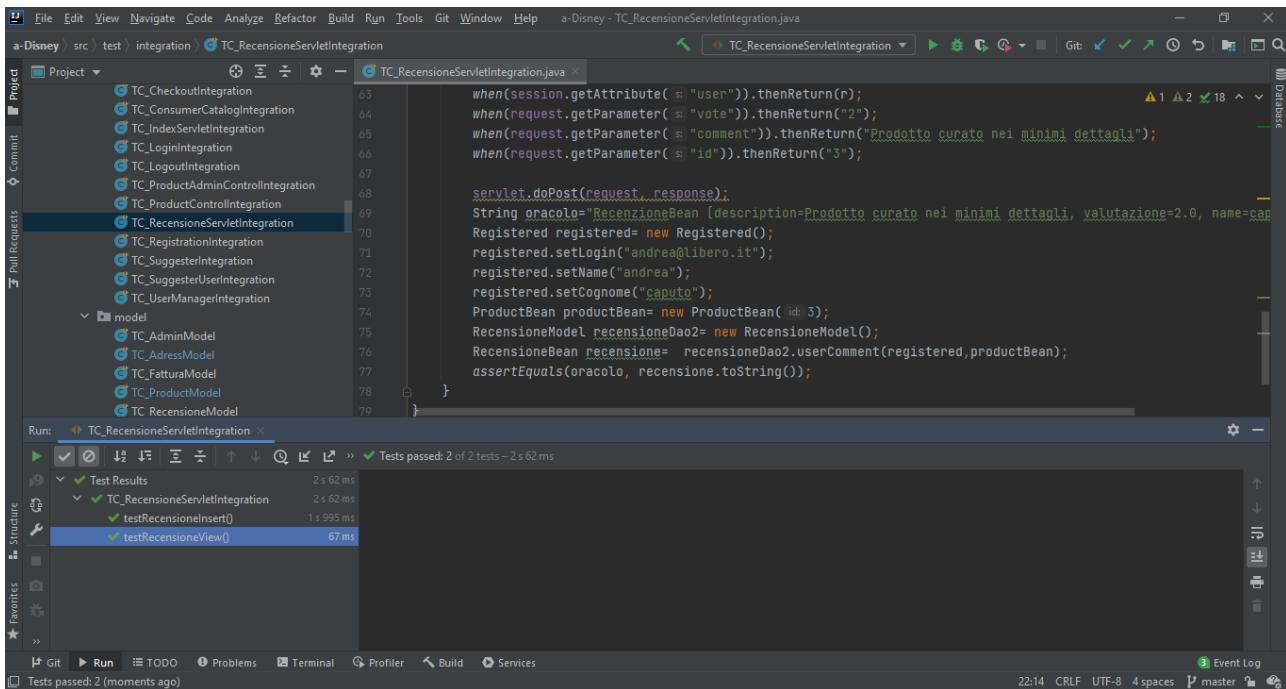
- File Path:** a-Disney > src > test > integration > TC_ProductAdminControlIntegration
- Code Editor:** The editor displays the `TC_ProductAdminControlIntegration.java` file, which contains Java code for testing a Product Admin Control integration.
- Run Tab:** The "Run" tab shows the test results for `TC_ProductAdminControlIntegration`. It lists five tests: `testPACDelete()`, `testPAC240()`, `testPAC250()` (which is highlighted in blue), `testPAC260()`, and `testPAC270()`. The total execution time is 2 seconds and 625 milliseconds.
- Event Log:** The event log at the bottom right indicates "Tests passed: 5 (moments ago)" and the current branch is "master".

PRODUCTCONTROLINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Path:** a-Disney > src > test > integration > TC_ProductControlIntegration
- Code Editor:** The editor displays the `TC_ProductControlIntegration.java` file, which contains Java code for testing a Product Control Integration using Mockito annotations.
- Run Tab:** The "Run" tab shows the test results for `TC_ProductControlIntegration`. It lists three tests: `testAddC()`, `testView()`, and `testDelete()`. The total execution time is 2 seconds and 196 milliseconds. The output window shows the command `C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...` and the message "Process finished with exit code 0".
- Event Log:** The event log at the bottom right indicates "Tests passed: 3 (moments ago)" and the current branch is "master".

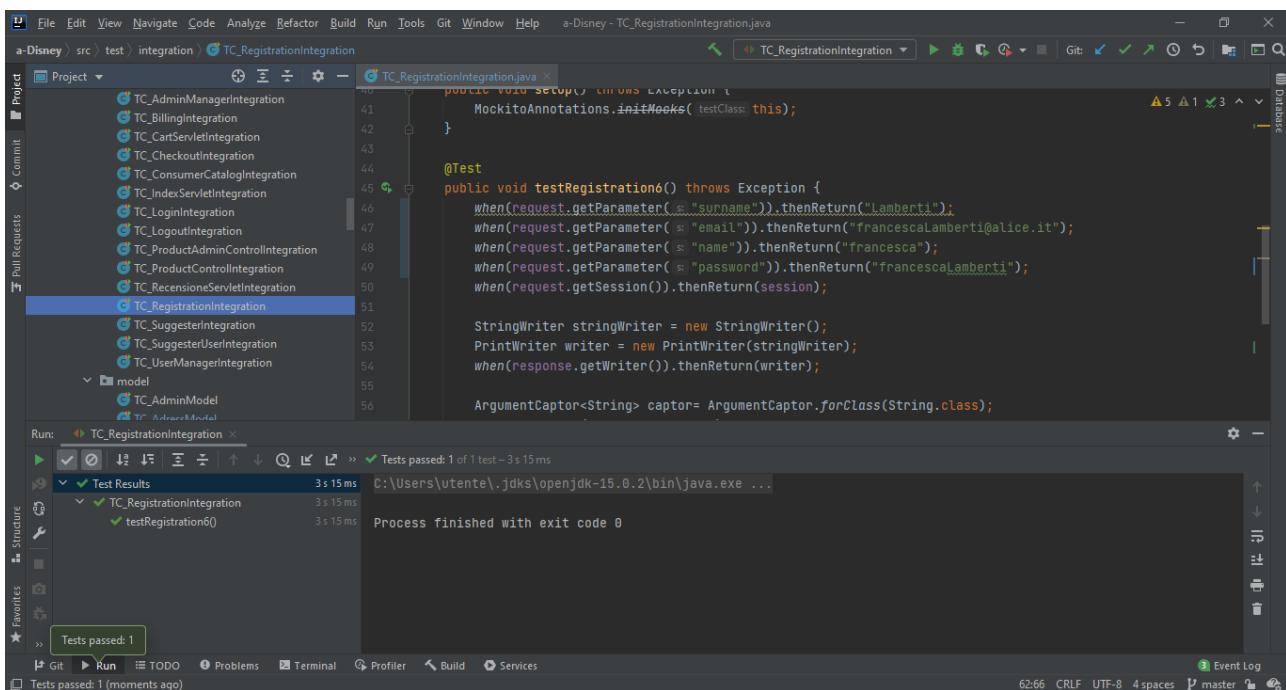
RECENSIONESERVLETINTEGRATION



```
a-Disney > src > test > integration > TC_RecensioneServletIntegration.java
public void testRecensioneView() throws Exception {
    when(session.getAttribute("user")).thenReturn(r);
    when(request.getParameter("vote")).thenReturn("2");
    when(request.getParameter("comment")).thenReturn("Prodotto curato nei minimi dettagli");
    when(request.getParameter("id")).thenReturn("3");

    servlet.doPost(request, response);
    String oracolo = RecensioneBean [description=Prodotto curato nei minimi dettagli, valutazione=2.0, name=caputo];
    Registered registered = new Registered();
    registered.setLogin("andrea@libero.it");
    registered.setName("andrea");
    registered.setCognome("caputo");
    ProductBean productBean = new ProductBean( id: 3 );
    RecensioneModel recensioneDao = new RecessioneModel();
    RecensioneBean recensione = recensioneDao.userComment(registered, productBean);
    assertEquals(oracolo, recensione.toString());
}
```

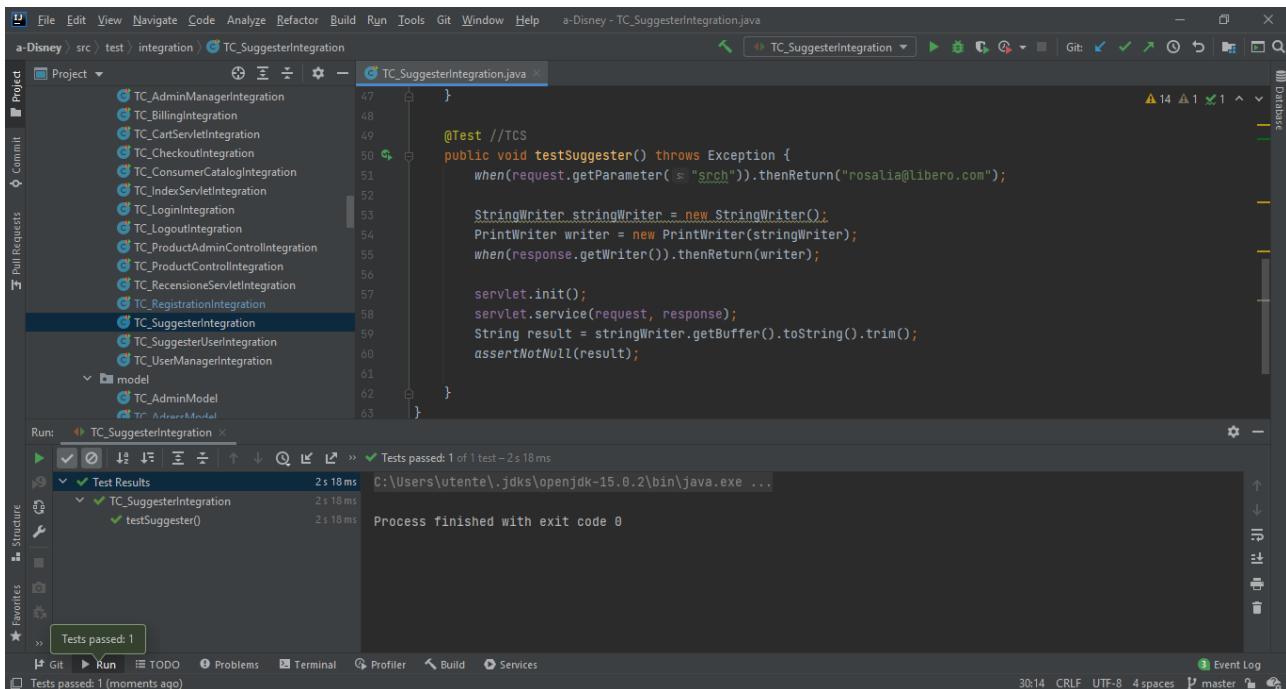
REGISTRATIONINTEGRATION



```
a-Disney > src > test > integration > TC_RegistrationIntegration.java
public void testRegistration6() throws Exception {
    when(request.getParameter("surname")).thenReturn("Lamberti");
    when(request.getParameter("email")).thenReturn("francescalamberti@alice.it");
    when(request.getParameter("name")).thenReturn("francesca");
    when(request.getParameter("password")).thenReturn("francescaLamberti");
    when(request.getSession()).thenReturn(session);

    StringWriter stringWriter = new StringWriter();
    PrintWriter writer = new PrintWriter(stringWriter);
    when(response.getWriter()).thenReturn(writer);
    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
}
```

SUGGESTERINTEGRATION



```
a-Disney > src > test > integration > TC_SuggererIntegration.java
File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help a-Disney - TC_SuggererIntegration.java
Project Commit Pull Requests Run: TC_SuggererIntegration
@Test //TCS
public void testSuggerer() throws Exception {
    when(request.getParameter("srch")).thenReturn("rosalia@libero.com");

    StringWriter stringWriter = new StringWriter();
    PrintWriter writer = new PrintWriter(stringWriter);
    when(response.getWriter()).thenReturn(writer);

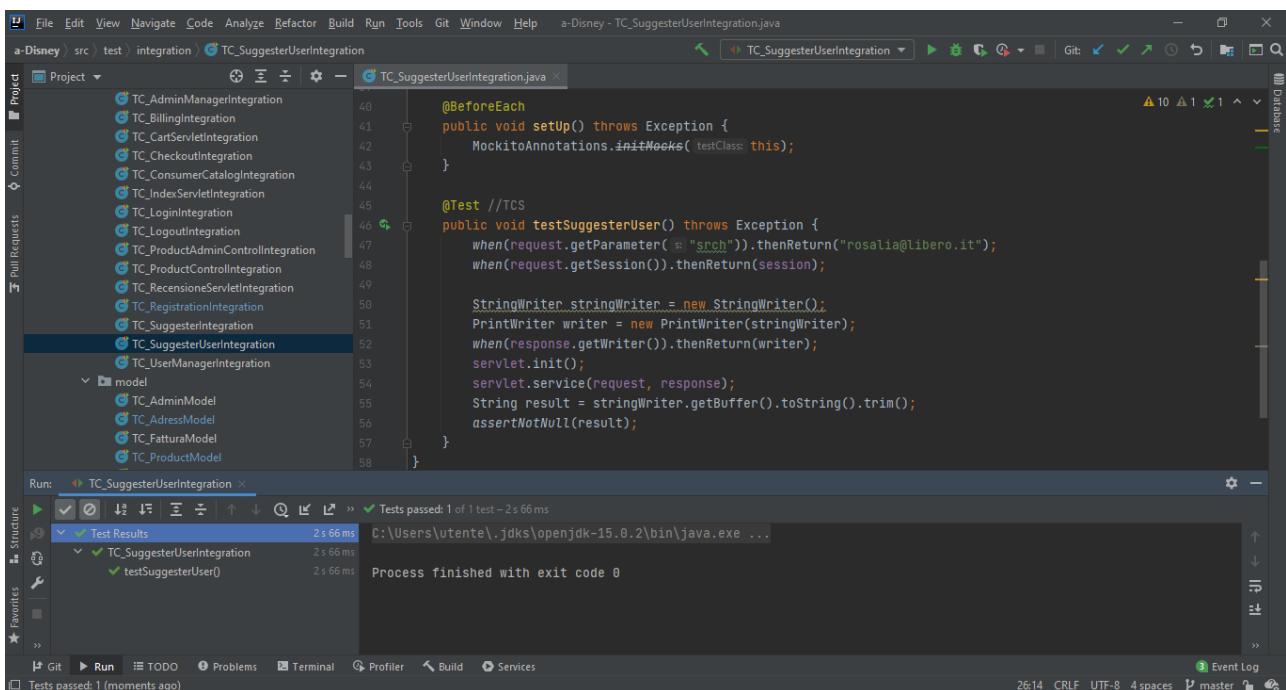
    servlet.init();
    servlet.service(request, response);
    String result = stringWriter.getBuffer().toString().trim();
    assertNotNull(result);
}

47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

Run: TC_SuggererIntegration
Tests passed: 1 of 1 test - 2 s 18 ms
Test Results
TC_SuggererIntegration
  ✓ testSuggerer()
  2 s 18 ms
Process finished with exit code 0

```

SUGGERERUSERINTEGRATION



```
a-Disney > src > test > integration > TC_SuggererUserIntegration.java
File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help a-Disney - TC_SuggererUserIntegration.java
Project Commit Pull Requests Run: TC_SuggererUserIntegration
@BeforeEach
public void setUp() throws Exception {
    MockitoAnnotations.initMocks(this);
}

@Test //TCS
public void testSuggererUser() throws Exception {
    when(request.getParameter("srch")).thenReturn("rosalia@libero.it");
    when(request.getSession()).thenReturn(session);

    StringWriter stringWriter = new StringWriter();
    PrintWriter writer = new PrintWriter(stringWriter);
    when(response.getWriter()).thenReturn(writer);
    servlet.init();
    servlet.service(request, response);
    String result = stringWriter.getBuffer().toString().trim();
    assertNotNull(result);
}

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

Run: TC_SuggererUserIntegration
Tests passed: 1 of 1 test - 2 s 66 ms
Test Results
TC_SuggererUserIntegration
  ✓ testSuggererUser()
  2 s 66 ms
Process finished with exit code 0

```

USERMANAGERINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_UserManagerIntegration.java`, which contains a test method `test1()` using the `@Test` annotation. The test sets up a registered user bean and performs various assertions based on session parameters and database retrievals. Below the code editor is the 'Run' tool window, which shows the test results: 2 tests passed in 2 seconds. The terminal output shows the command `C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...` and the message 'Process finished with exit code 0'. The bottom status bar indicates 'CRLF' and 'UTF-8' encoding.

```
@Test //TC55
public void test1() throws Exception {
    Registered userBean= new Registered( name: "rosalia", cognome: "capozzolo", login: "rosalia@libero.it", passw
    when(request.getSession()).thenReturn(session);
    when(request.getParameter( s: "op")).thenReturn("modCred");
    when(request.getSession().getAttribute( s: "user")).thenReturn(userBean);
    when(request.getParameter( s: "name")).thenReturn("rosalia");
    when(request.getParameter( s: "cognome")).thenReturn("capozzolo");
    when(request.getParameter( s: "Login")).thenReturn("rosalia@libero.it");
    when(request.getParameter( s: "pass")).thenReturn("rosalia");

    servlet.doPost(request, response);
    RegisteredModel registeredDao2= new RegisteredModel();
    Registered registrato= registeredDao2.doRetrieveByKey( user: "rosalia@libero.it");
    assertEquals( expected: "rosalia",registrato.getName());
    assertEquals( expected: "rosalia",registrato.getPassword());
    assertEquals( expected: "capozzolo",registrato.getCognome());
```

VISUALIZZAINDIRIZZIINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_VisualizzaindirizziIntegration.java`, which contains a test method `testConsumerCatalog1()` using the `@Test` annotation. The test logs in a user, sets session attributes, and verifies a forwarded response. Below the code editor is the 'Run' tool window, which shows the test results: 1 test passed in 2 seconds. The terminal output shows the command `C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...` and the message 'Process finished with exit code 0'. The bottom status bar indicates 'CRLF' and 'UTF-8' encoding.

```
@Test
public void testConsumerCatalog1() throws Exception {
    Registered user = new Registered();
    user.setLogin("rosalia@libero.it");
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute( s: "user")).thenReturn(user);
    when(response.encodeURL( s: "/protected-page.jsp")).thenReturn("/protected-page.jsp");
    when(request.getRequestDispatcher( s: "/protected-page.jsp")).thenReturn(rd);

    servlet doGet(request, response);
    verify(rd).forward(request, response);
    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    verify(response).encodeURL(captor.capture());
    assertEquals( expected: "/protected-page.jsp", captor.getValue());
```

KATALON

TEST DI SISTEMA GESTIONE UTENTE

LOGIN

login

...

- TCS4
- TCS5
- TCS3
- TCS2
- TCS1

REGISTRAZIONE

Registrazione2.0

...

- TCS6
- TCS11
- TCS10
- TCS9
- TCS8
- TCS7

CAMBIO NOME

CambioNome2.0

...

- TCS12
- TCS13

CAMBIO COGNOME

CambioCognome2.0 :

- TCS14
- TCS15

TEST DI SISTEMA GESTIONE PRODOTTO

AGGIUNTA RECENSIONE

Aggiunta recensione2.0 :

- TCS24
- TCS23

MODIFICA PRODOTTO

Modifica prodotto 2.0 :

- TCS25
- TCS26

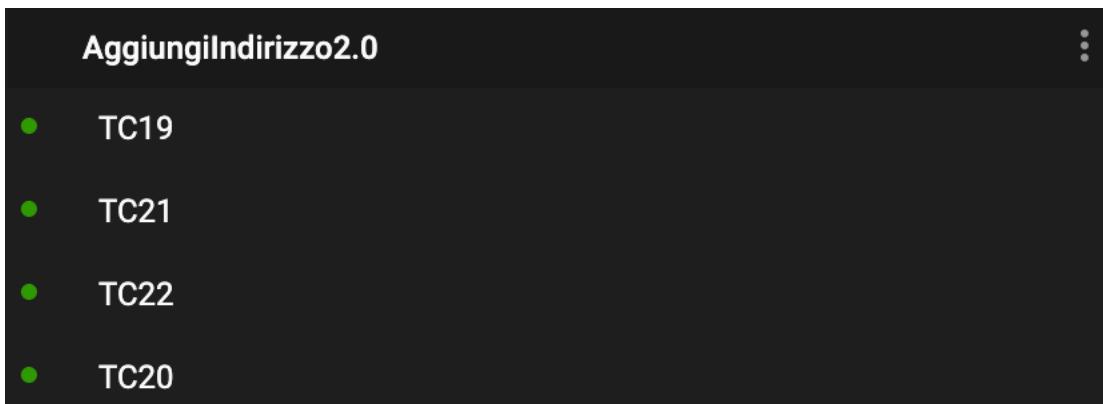
AGGIUNGI PRODOTTO

Aggiunta nuovo prodotto 2.0 :

- TCS28
- TCS27

TEST DI SISTEMA GESTIONE INDIRIZZO

AGGIUNGI INDIRIZZO



TEST DI SISTEMA GESTIONE ORDINE

PAGAMENTO

