



# Testarea unei aplicații mobile

Lungu Laura-Vanesa  
Negoitǎ-Crețu Raluca-Marina



# Introducere în Testarea Sistemelor Software

Testarea sistemelor software este procesul de **verificare și validare** a funcționării corecte a aplicațiilor **înainte de lansare**, pentru a detecta defecte și a asigura **calitatea finală**.

## Clasificare tipuri de testare:

- **după accesul la cod**
  - **White-Box Testing:** Testarea logicii interne (ex: acoperire cod, teste pe ramuri/căi)
  - **Black-Box Testing:** Testarea funcționalității fără a cunoaște codul (ex: interfețe, API-uri)
  - **Grey-Box Testing:** Combină cele două — cunoaștere parțială a structurii interne
- **după tipul de testare**
  - **Testare Funcțională**
    - **Unit Testing:** testează componentele mici și independente
    - **Integration Testing:** verifică integrarea modulelor
    - **System Testing:** testează întregul sistem, ca un tot unitar
    - **End-to-End Testing:** simulează un utilizator real de la început până la final
  - **Testare Non-Funcțională**
    - **Performance Testing:** evaluează viteză și stabilitate sub sarcină
    - **Usability Testing:** măsoară ușurința în utilizare
    - **Compatibility Testing:** testează funcționarea pe diverse dispozitive, sisteme și browsere

# Arhitectura aplicației testate - MovieMingle

MovieMingle este o aplicație mobilă care recomandă filme pe baza preferințelor utilizatorilor, folosind tehnici de învățare automată. Oferă o interfață prietenoasă și un design modern, adaptat pasionaților de film.

## Adaptare Backend:

- Backend reutilizat dintr-un proiect web anterior
- Framework: **Java Spring Boot**
- Include logica de recomandare și endpoint-uri REST
- Adaptat pentru funcționalitatea mobilă

## Arhitectură Client-Server:

- **Frontend (React Native):** gestionează UI/UX și trimite cereri HTTP
- **Backend (Spring Boot):** procesează logica aplicației, autentificare, recomandări, conectare la MySQL
- **Securitate:** autentificare cu JWT, protecție CSRF
- **Integrare externă:** API TMDb – furnizează informații actualizate despre filme (descrieri, actori, ratinguri)



# Ghost Game

2024 | 1h 26min | Directed by: Jill Gevargizian

3.33 Horror Thriller

As part of an internet challenge to live undetected in a stranger's home, a daring couple target an infamous haunted house and endure a series of chilling incidents as they witness a family descending into madness.

See More

## Most Popular Today

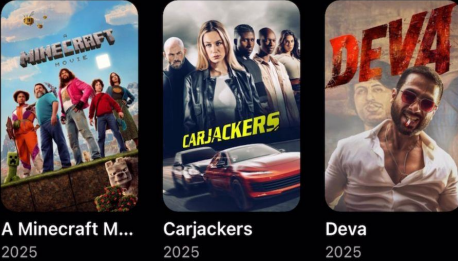


## Movie Explorer

Find the perfect movie to match your mood

Search for movies...

Action Adventure Animation Comedy



## HomeScreen



## Dead Poets Society

1989 | 2h 8min | Directed by: Peter Weir

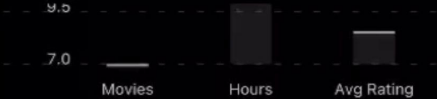
8.30 Drama

At an elite, old-fashioned boarding school in New England, a passionate English teacher inspires his students to rebel against convention and seize the potential of every day, courting the disdain of the stern headmaster.

Add to Favorite Watchlist Mark as Watched

How would you rate this movie?  
☆☆☆☆☆☆☆☆☆☆

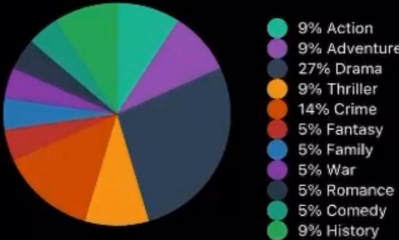
## Your Profile



## Monthly Activity



## Genre Distribution



Home

Search

MovieList

Profile

Home

Search

MovieList

Profile

Home

Search

MovieList

Profile

Home

Search

MovieList

Your Profile

# Framework-uri de testare

**Obiectiv:** Utilizarea framework-urilor potrivite pentru testarea componentelor frontend și backend ale aplicației.

## Frontend:

- **Jest + React Native Testing Library**  
Testare unitară și de integrare pentru componente și funcționalitate de bază (*JavaScript*)
- **Maestro**  
Testare end-to-end a fluxurilor reale de utilizator în aplicația mobilă (*YAML scripts + Android Emulator*)

## Backend:

- **JUnit 5 + Mockito**  
Testare unitară a serviciilor backend prin simularea dependențelor externe (*Java + Spring Boot*)
- **JUnit 5 cu SpringBootTest**  
Testare de integrare între servicii, repo-uri și baza de date in-memory (H2) (*Java + Spring Boot*)
- **TestNG**  
Testare funcțională cu scenarii complexe și organizare avansată (*Java + Spring Boot*)
- **JaCoCo**  
Măsurarea gradului de acoperire a codului de către testele automate (*Java + Gradle/Maven*)

# Testare Frontend - Jest & RNTL

**Jest** - Framework de testare JavaScript dezvoltat de Facebook, folosit pentru teste unitare, de integrare și snapshot. Rulează rapid, local, fără emulator, și suportă mock-uri automate.

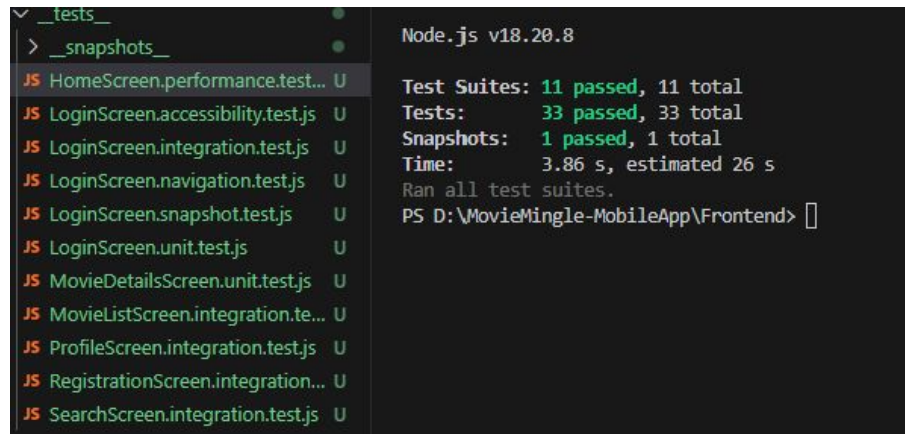
**React Native Testing Library (RNTL)** - Extinde Jest pentru testarea componentelor React Native prin simularea comportamentului real al utilizatorilor (ex: apăsări, completare formulare).

## De ce Jest + RNTL?

- Testare rapidă, fără emulator
- Integrare ușoară cu Expo
- Suport complet pentru mock-uri
- Ideal pentru teste unitare și funcționale

## Teste implementate în proiect:

- **Unitare:** formulare, input-uri, validări
- **Funcționale:** încărcare date, afișare mesaje, erori
- **Integrare:** login, navigare, căutare filme
- **Performanță:** randare rapidă HomeScreen
- **Snapshot:** verificare UI stabil
- **Accesibilitate:** label-uri și interacțiuni accesibile



```
Node.js v18.20.8

Test Suites: 11 passed, 11 total
Tests:       33 passed, 33 total
Snapshots:   1 passed, 1 total
Time:        3.86 s, estimated 26 s
Ran all test suites.
PS D:\MovieMingle-MobileApp\Frontend>
```

# Testare Frontend cu Maestro

## Ce este Maestro?

Framework de testare end-to-end pentru aplicații mobile (iOS/Android), bazat pe YAML. Simplu de configurat, ideal pentru aplicații dezvoltate cu **React Native + Expo**.

## De ce Maestro?

- Scriere rapidă a testelor în YAML
- Nu necesită cod JavaScript sau setup complicat
- Ideal pentru simularea fluxurilor reale de utilizator
- Integrare ușoară cu aplicații Expo (`npx expo run:android`)

## Testare în MovieMingle:

Au fost realizate 4 teste E2E:

- **Autentificare** – login cu date reale
- **Profil utilizator** – deschidere modal „Change Avatar”
- **HomeScreen** – interacțiune cu filme, favorite & watched
- **Search** – căutare filme după cuvinte-cheie și gen

```
Frontend > maestro > ! test-login.yaml
1  appId: com.ncraluca.Frontend
2  ---
3  - launchApp
4  - tapOn: "Log In"
5  - tapOn: "Enter your email"
6  - inputText: "ralucanegoita13@gmail.com"
7  - tapOn: "Enter your password"
8  - inputText: "1213.Ralu"
9  - tapOn: "Log in"
10 - tapOn: "Log in"
11 - assertVisible: "You're now logged in!"
12 - tapOn: "OK"
13
```

Maestro a oferit o soluție eficientă pentru testarea end-to-end rapidă și stabilă pe Android, fără emulator complex.

# Studiu Comparativ: Jest + RNTL + Maestro

## Jest + RNTL

- Testare: unitară & de integrare
- Nivel: componente, logică internă
- Limbaj: JavaScript
- Rulează local, fără emulator
- Control complet (mocks, spies)
- Execuție rapidă
- Ideal pentru: testarea logicii & UI

## Maestro

- Testare: end-to-end (E2E)
- Nivel: flow-uri reale de utilizator
- Limbaj: YAML (declarativ)
- Necesită emulator sau device fizic
- Control limitat (doar interfața)
- Execuție mai lentă
- Ideal pentru: validare comportament real

Jest + RNTL sunt ideale pentru teste precise și rapide la nivel de componentă. Maestro e potrivit pentru verificarea completă a fluxurilor reale într-un mediu cât mai apropiat de utilizatorul final.



# Testare Backend - JUnit 5 + Mockito

Testare unitară, de integrare și funcțională pentru serviciile Java Spring Boot

**JUnit 5** - Framework standard pentru testare unitară în Java (adnotări moderne, suport parametrizat, integrare cu IDE-uri și build tools).

**Mockito** - Bibliotecă de mocking pentru simularea dependențelor (ex: repo-uri, API-uri). Permite izolarea logicii testate.

## De ce JUnit + Mockito?

- Structură clară a testelor
- Izolare completă a logicii testate
- Fără baze de date reale
- Rulare rapidă, testare precisă

## Teste implementate în proiect:

- **Unitare:** validare înregistrare, codificare parolă, token-uri
- **Integrare:** autentificare, recomandări, dashboard, API + DB
- **Funcționale:** resetare parolă, adăugare rating, recomandări filme

## Exemple:

- *testRegister\_UserAlreadyExists()* – verifică excepția la user duplicat
- *testGetUserDashboardStats()* – testează generarea completă a statisticilor
- *testForgottenPassword\_Success()* – simulează resetarea parolei prin email

```
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.150 s --
[INFO] Running com._errors.MovieMingle.service.DefaultSecureTokenServiceTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.254 s --
[INFO] Running com._errors.MovieMingle.service.RatingServiceTest
[INFO] Tests run: 13, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.101 s --
[INFO] Running com._errors.MovieMingle.service.user.DefaultAppUserAccountServiceT
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.169 s --
[INFO] Running com._errors.MovieMingle.service.user.UserFavouritesServiceTest
[INFO] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.082 s --
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 44, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.729 s
[INFO] Finished at: 2025-05-14T20:48:03:00
[INFO] -----
PS D:\MovieMingle-MobileApp\Backend>
```

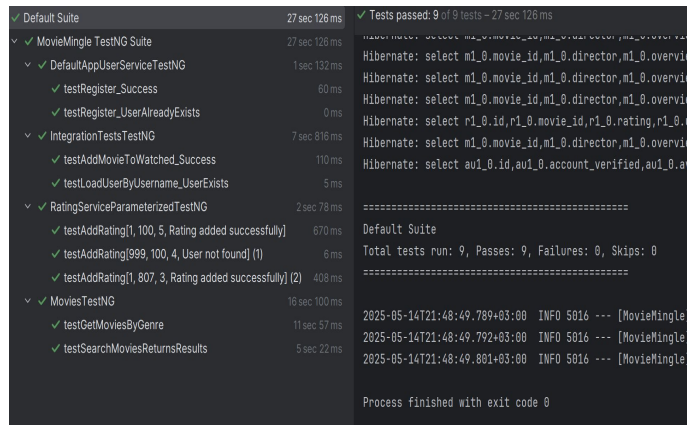
# Testare Backend - TestNG

Framework avansat pentru testare unitară, de integrare și funcțională în Java, inspirat din JUnit, cu funcționalități extinse:

- Teste parametrizate și grupate
- Control asupra ordinii de execuție
- Suport pentru paralelizare
- Annotări avansate: `@BeforeMethod`, `@Test(groups=...)`

## Teste implementate în proiect:

- **Teste unitare**
  - `testRegister_UserAlreadyExists` – verifică excepția la înregistrare duplicată
  - `testRegister_Success` – confirmă salvarea și trimiterea e-mailului
- **Teste de integrare**
  - `testAddMovieToWatched_Success` – validează adăugarea filmului la vizionate
  - `testLoadUserByUsername` – încarcă utilizator real din DB
- **Teste funcționale (parametrizate)**
  - `testAddRating` – acoperă scenarii pentru utilizator/film valid sau inexistent
  - `testSearchMoviesReturnsResults` – căutare filme după cuvânt-cheie
  - `testGetMoviesByGenre` – filtrează corect filmele pe gen



The screenshot shows the TestNG output in a terminal. It lists the execution of various test suites and individual tests with their durations. At the bottom, it shows the 'Default Suite' summary: 'Total tests run: 9, Passes: 9, Failures: 0, Skips: 0'. It also includes a timestamp and log level for each test run, and finally states 'Process finished with exit code 0'.

```
✓ Default Suite 27 sec 126 ms ✓ Tests passed: 9 of 9 tests - 27 sec 126 ms
✓ MovieMingle TestNG Suite 27 sec 126 ms
  ✓ DefaultAppUserServiceTestNG 1 sec 132 ms
    ✓ testRegister_Success 60 ms
    ✓ testRegister_UserAlreadyExists 0 ms
  ✓ IntegrationTestsTestNG 7 sec 816 ms
    ✓ testAddMovieToWatched_Success 110 ms
    ✓ testLoadUserByUsername_UserExists 5 ms
  ✓ RatingServiceParameterizedTestNG 2 sec 78 ms
    ✓ testAddRating[1, 100, 5, Rating added successfully] 670 ms
    ✓ testAddRating[999, 100, 4, User not found] (1) 6 ms
    ✓ testAddRating[1, 807, 3, Rating added successfully] (2) 408 ms
  ✓ MoviesTestNG 16 sec 100 ms
    ✓ testGetMoviesByGenre 11 sec 57 ms
    ✓ testSearchMoviesReturnsResults 5 sec 22 ms

Default Suite
Total tests run: 9, Passes: 9, Failures: 0, Skips: 0

2025-05-14T21:48:49.789+03:00 INFO 5016 --- [MovieMingle]
2025-05-14T21:48:49.792+03:00 INFO 5016 --- [MovieMingle]
2025-05-14T21:48:49.801+03:00 INFO 5016 --- [MovieMingle]

Process finished with exit code 0
```

TestNG oferă o soluție robustă și flexibilă pentru testarea avansată a aplicațiilor Java Spring Boot, cu suport pentru scenarii complexe și testare multiplă prin `@DataProvider`.

# Studiu comparativ: JUnit + Mockito + TestNG

## JUnit + Mockito

- Simplu, intuitiv, integrat nativ cu Spring Boot
- Suport limitat pentru parametrizare (@ParameterizedTest)
- Necesită extensii pentru rulare paralelă
- Ideal pentru servicii, REST APIs și mock-uri
- Cod clar, focus pe testare logică

## TestNG

- Mai configurabil, cu suport nativ pentru grupuri și priorități
- Parametrizare avansată (@DataProvider, @Factory)
- Suport nativ pentru testare paralelă
- Ideal pentru testare funcțională și data-driven
- Flexibil, potrivit pentru proiecte mari și enterprise

JUnit + Mockito e excelent pentru testare rapidă, clară și izolată. TestNG oferă putere și flexibilitate pentru scenarii complexe și rulări paralele.

# Grad de Acoperire Backend - JaCoCo

## Evaluarea calității testelor unitare și de integrare în MovieMingle

- Acoperire instrucțiuni: **53%**
- Acoperire ramuri decizie: **41%**
- Clase acoperite: **91%** (63 din 69)
- Linii netestate: **1457 din 2119**

### Zone bine acoperite:

- seeder – 92%
- recommendation – 80%
- service.email.context – 71%

### Acoperire moderată:


- service – 65%
- model – 56%
- dto – 48%

### Zone slab acoperite:

- controller – 9%
- security – 18%
- service.email – 5%

Logica de business și recomandările sunt bine testate.

Lipsesc teste automate pe API-uri (controller), securitate și e-mail — zone critice care necesită atenție pentru stabilitate și siguranță în producție.



# MovieMingle

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes			
<a href="#">com_errors.MovieMingle.controller</a>	<div><div></div></div>	0%	<div><div></div></div>	0%	83	83	233	233	61	61	14	14			
<a href="#">com_errors.MovieMingle.service</a>	<div><div></div></div>	42%	<div><div></div></div>	37%	42	75	124	237	15	36	1	4			
<a href="#">com_errors.MovieMingle.service.user</a>	<div><div></div></div>	30%	<div><div></div></div>	25%	53	72	155	219	21	31	4	6			
<a href="#">com_errors.MovieMingle.dto</a>	<div><div></div></div>	17%	<div><div></div></div>	10%	102	133	165	211	89	119	10	13			
<a href="#">com_errors.MovieMingle.seeder</a>	<div><div></div></div>	0%	<div><div></div></div>	0%	17	17	81	81	6	6	3	3			
<a href="#">com_errors.MovieMingle.model</a>	<div><div></div></div>	41%		n/a	57	102	99	175	57	102	1	7			
<a href="#">com_errors.MovieMingle.config</a>	<div><div></div></div>	0%	<div><div></div></div>	0%	16	16	41	41	12	12	2	2			
<a href="#">com_errors.MovieMingle.security</a>	<div><div></div></div>	0%	<div><div></div></div>	0%	12	12	49	49	7	7	2	2			
<a href="#">com_errors.MovieMingle.service.email.context</a>	<div><div></div></div>	44%	<div><div></div></div>	50%	22	32	38	65	21	31	1	3			
<a href="#">com_errors.MovieMingle.recommendation</a>	<div><div></div></div>	80%	<div><div></div></div>	66%	12	37	19	108	3	19	0	4			
<a href="#">com_errors.MovieMingle.service.email</a>	<div><div></div></div>	0%		n/a	2	2	13	13	2	2	1	1			
<a href="#">com_errors.MovieMingle.exception</a>	<div><div></div></div>	22%		n/a	7	9	14	18	7	9	1	3			
<a href="#">com_errors.MovieMingle</a>	<div><div></div></div>	0%		n/a	3	3	7	7	3	3	1	1			
Total		3,962 of 5,480		27%	232 of 310		25%	428	593	1,038	1,457	304	438	41	63