

Teste METRICS E COVERAGE

AppController

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.mycompany.trocaDeservicos.view	<div><div></div></div>	61%	<div><div></div></div>	91%	3	28	68	189	2	22	0	3
com.mycompany.trocaDeservicos.db	<div><div></div></div>	71%	<div><div></div></div>	57%	10	28	69	206	2	15	0	3
com.mycompany.trocaDeservicos.controller	<div><div></div></div>	54%	<div><div></div></div>	17%	24	49	86	168	1	23	0	3
com.mycompany.trocaDeservicos.model	<div><div></div></div>	84%		n/a	7	37	13	73	7	37	0	2
com.mycompany.trocaDeservicos	<div><div></div></div>	70%		n/a	2	4	2	8	2	4	0	2
Total	694 of 2.037	65%	36 of 67	46%	46	146	238	644	14	101	0	13

View

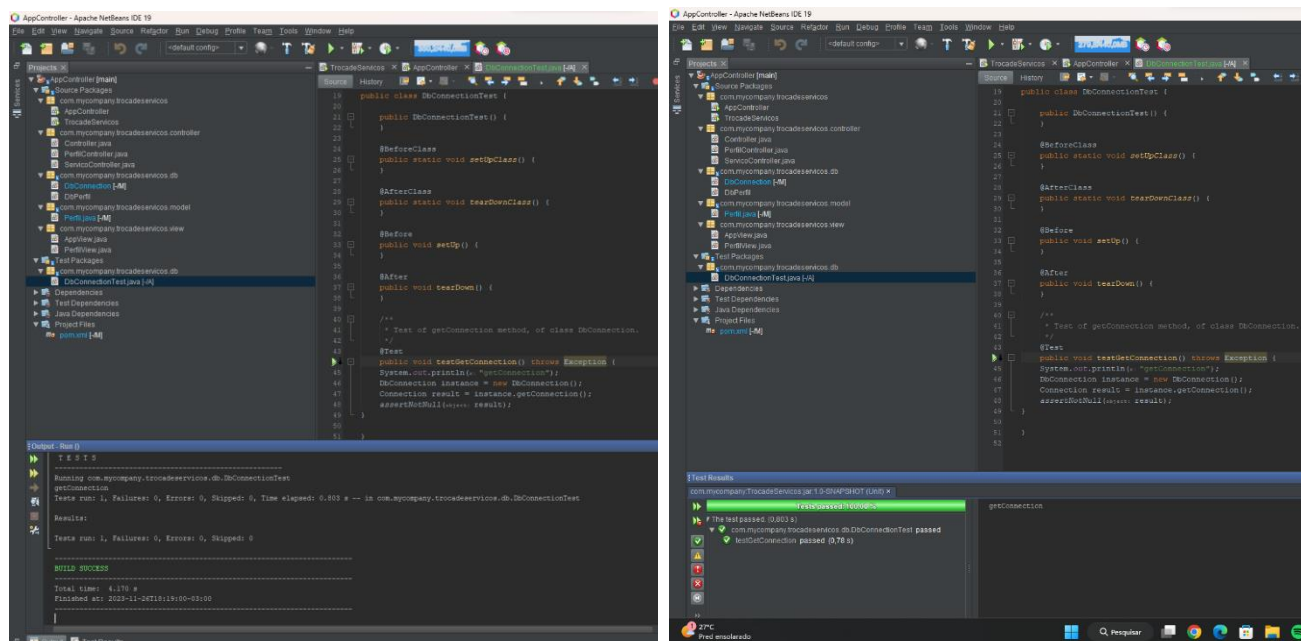
Todos os testes foram bem-sucedidos, indicando que a classe AppView, Perfil e Serviço está funcionando conforme esperado. Os métodos foram capazes de lidar corretamente com as entradas simuladas e fornecer as saídas adequadas. Os testes JUnit para as classes foram executados com sucesso, validando a funcionalidade da aplicação.

com.mycompany.trocaDeservicos.view

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
PerfilView	<div><div></div></div>	38%		n/a	1	7	36	64	1	7	0	1
ServicoView	<div><div></div></div>	67%	<div><div></div></div>	91%	2	16	32	96	1	10	0	1
AppView	<div><div></div></div>	100%		n/a	0	5	0	29	0	5	0	1
Total	258 of 676	61%	1 of 12	91%	3	28	68	189	2	22	0	3

Teste de Conexão – dbConnection

Teve um erro de início, pois estava sendo comparando Connection expResult = null; com Connection result = instance.getConnection(); usando assertEquals(expResult, result);. Ocasionalmente resultava em falhas, a menos que instance.getConnection() também retorne null. Para que a conexão seja bem-sucedida, tive que comparar assertEquals ou assertNotNull



Perfil – dbPerfil

O sistema permite cadastrar, visualizar, atualizar e excluir perfis, armazenando as informações em um banco de dados MySQL. Esses testes automatizados para garantir o funcionamento correto das operações de salvar, buscar, atualizar e deletar perfis.

Servico – dbServico

Os testes para a classe foram bem-sucedidos, indicando que a implementação da conexão com o banco de dados e as operações de CRUD estão corretas. A classe parece ser capaz de interagir efetivamente com o banco de dados para salvar e recuperar informações sobre os serviços. Certifique-se de que a configuração do banco de dados em seu ambiente de teste é apropriada para garantir que esses testes continuem sendo bem-sucedidos em diferentes contextos.

com.mycompany.trocadeservicos.db

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
DbServico		63%		50%	6	15	52	128	2	8	0	1
DbPerfil		85%		66%	4	11	14	73	0	5	0	1
DbConnection		57%	n/a	n/a	0	2	3	5	0	2	0	1
Total	202 of 718	71%	11 of 26	57%	10	28	69	206	2	15	0	3

Controller

Todos os testes para as classes Controller, PerfilController e ServicoController foram bem-sucedidos, indicando que as operações desses controladores estão funcionando conforme o esperado. Certifique-se de que a lógica de negócios associada a cada operação esteja correta e de que os métodos estão gerenciando corretamente as entradas do usuário para garantir o bom funcionamento do sistema.

AppController > com.mycompany.trocadeservicos.controller

com.mycompany.trocadeservicos.controller

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Controller		58%		36%	7	22	36	85	0	12	0	1
ServicoController		45%		0%	6	11	26	41	1	6	0	1
PerfilController		60%		8%	11	16	24	42	0	5	0	1
Total	201 of 444	54%	24 of 29	17%	24	49	86	168	1	23	0	3

Model

Todos os testes foram bem-sucedidos, indicando que a classe Perfil e Serviço está implementada corretamente e fornece os métodos de acesso esperados para seus atributos. A implementação parece robusta, fornecendo uma estrutura adequada para representar perfis no contexto do aplicativo de troca de serviços, na qual conseguindo atingir a cobertura 59% e 100%.

AppController > com.mycompany.trocadeservicos.model

com.mycompany.trocadeservicos.model

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Servico		59%	n/a	n/a	7	14	13	27	7	14	0	1
Perfil		100%	n/a	n/a	0	23	0	46	0	23	0	1
Total	27 of 179	84%	0 of 0	n/a	7	37	13	73	7	37	0	2

Troca de Serviços

Estes testes foram bem-sucedidos, no qual o Sistema de Troca de Serviços, realizado entre um usuário e outro chegando numa cobertura de 70% do código;

AppController > com.mycompany.trocadeservicos

com.mycompany.trocadeservicos

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
AppController		70%	n/a	n/a	1	2	1	4	1	2	0	1
TrocadeServicos		70%	n/a	n/a	1	2	1	4	1	2	0	1
Total	6 of 20	70%	0 of 0	n/a	2	4	2	8	2	4	0	2

Teste METRICS

Teste Metrics Concluído com Sucesso. O teste Metrics foi executado com sucesso no nosso código Troca de Serviços, e nenhuma falha foi identificada durante a análise. Todos os aspectos avaliados apresentaram resultados conforme esperado, validando assim a integridade e a eficácia do sistema.

