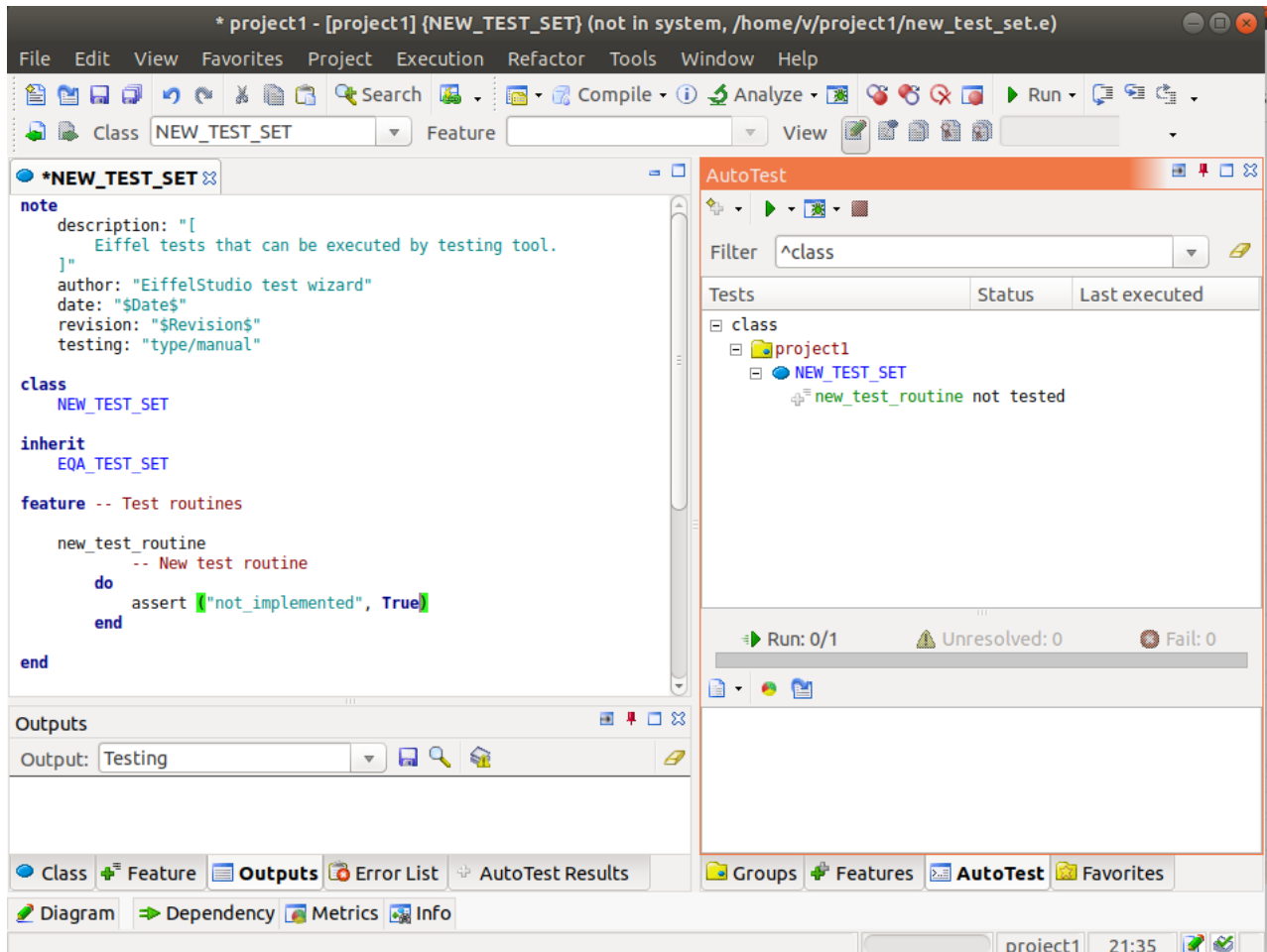


Testare manuala Eiffel

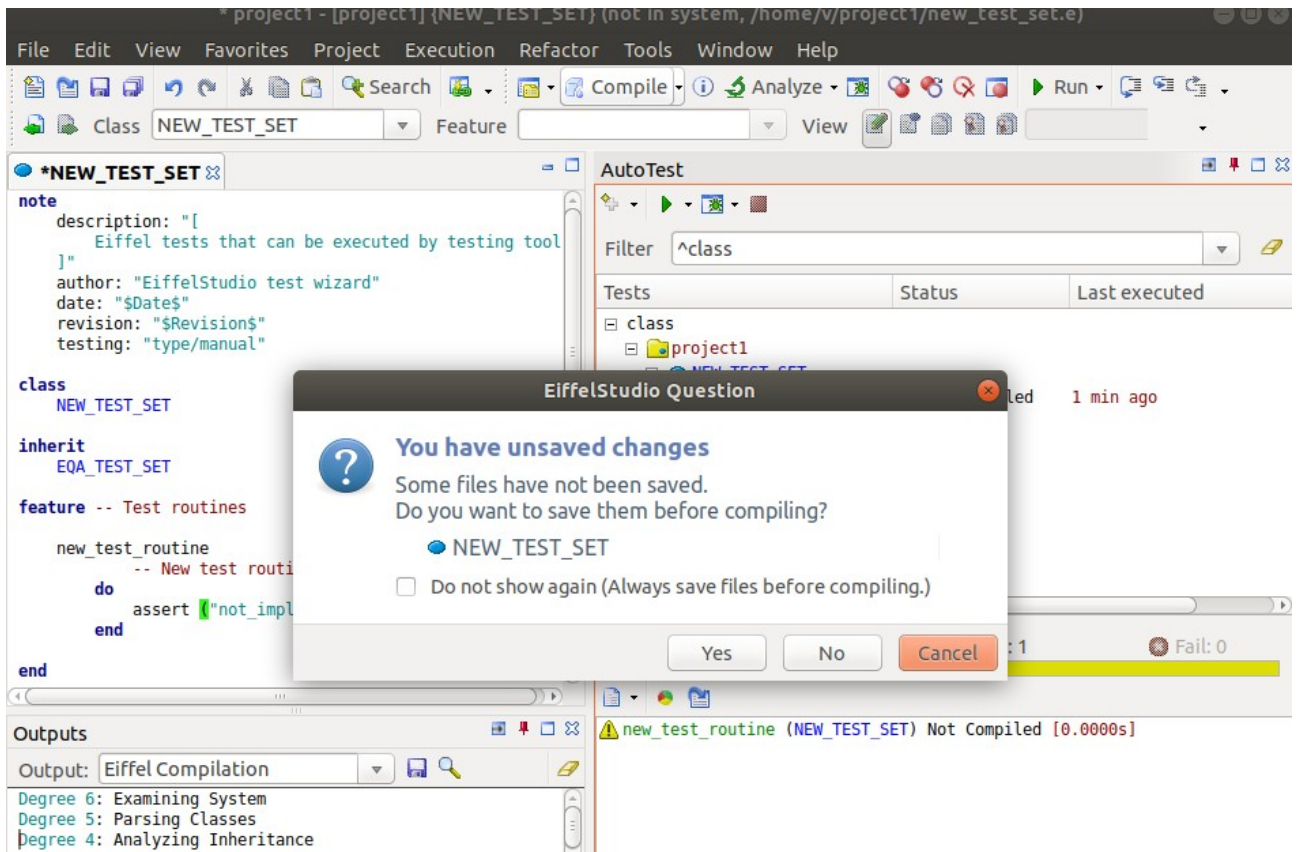
CUPRINS

creare test.....	2
compilare.....	2
executie.....	4
biblioteci.....	5
Detalii.....	6
Linii de cod.....	7
Proiect.....	7
Detaliu metrice LOC.....	8
detalii BON (Business Object Notation).....	8
detalii mostenitori.....	9

create test



compile



teste mai

executie

The screenshot displays the EiffelStudio IDE interface. The main editor window shows the source code for the `NEW_TEST_SET` class, which inherits from `EQA_TEST_SET`. A feature named `-- Test routines` contains a routine `new_test_routine` that asserts `"not_implemented", True`. The `Outputs` panel at the bottom left shows the execution of `new_test_routine (NEW_TEST_SET): pass` and the message `Execution complete`.

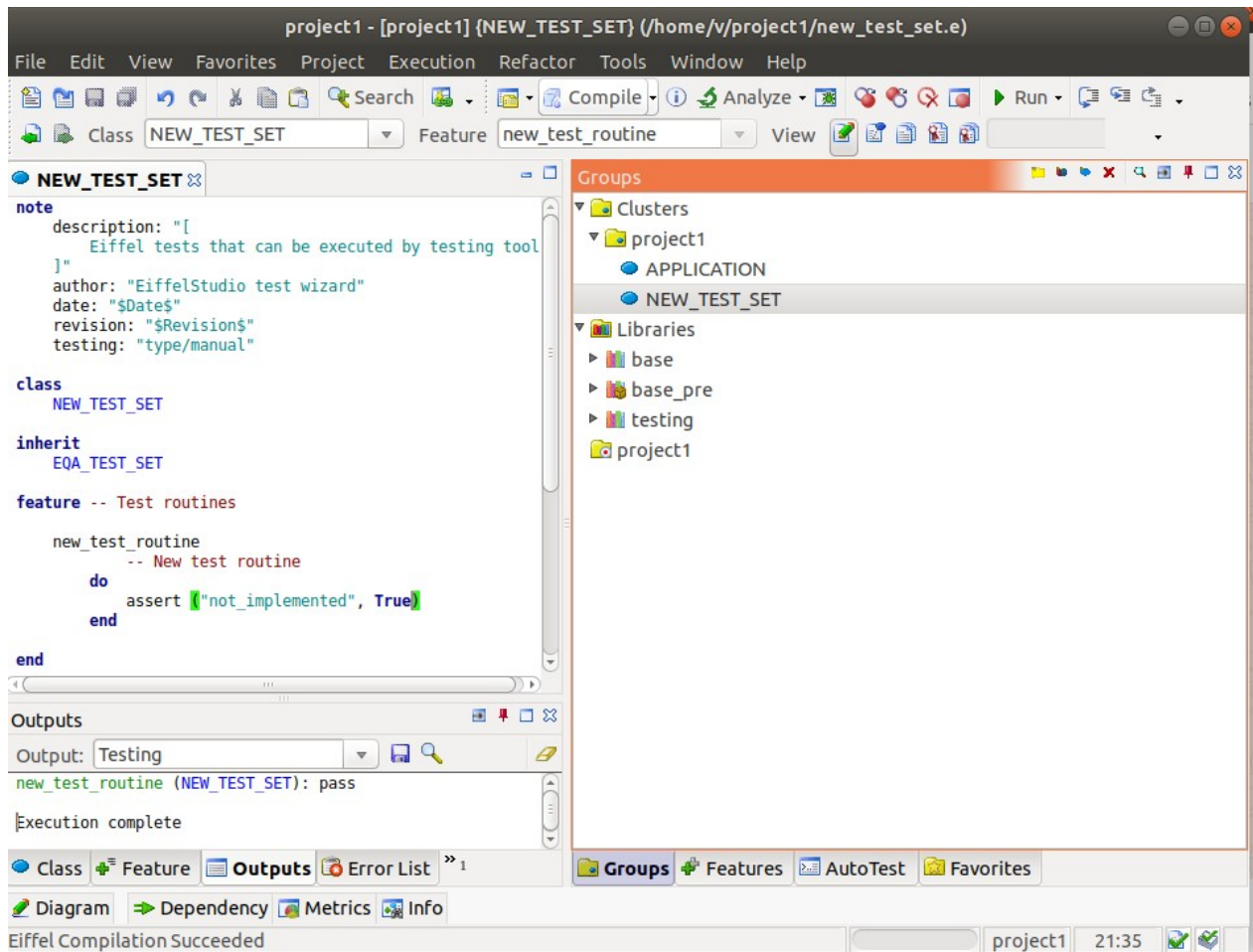
The `AutoTest` panel on the right provides a summary of the test results. It shows a table of tests with columns for `Tests`, `Status`, and `Last executed`. The test `new_test_routine` is listed with a green checkmark status and was executed `1 min ago`.

Tests	Status	Last executed
class		
project1		
NEW_TEST_SET		
new_test_routine	✓	1 min ago

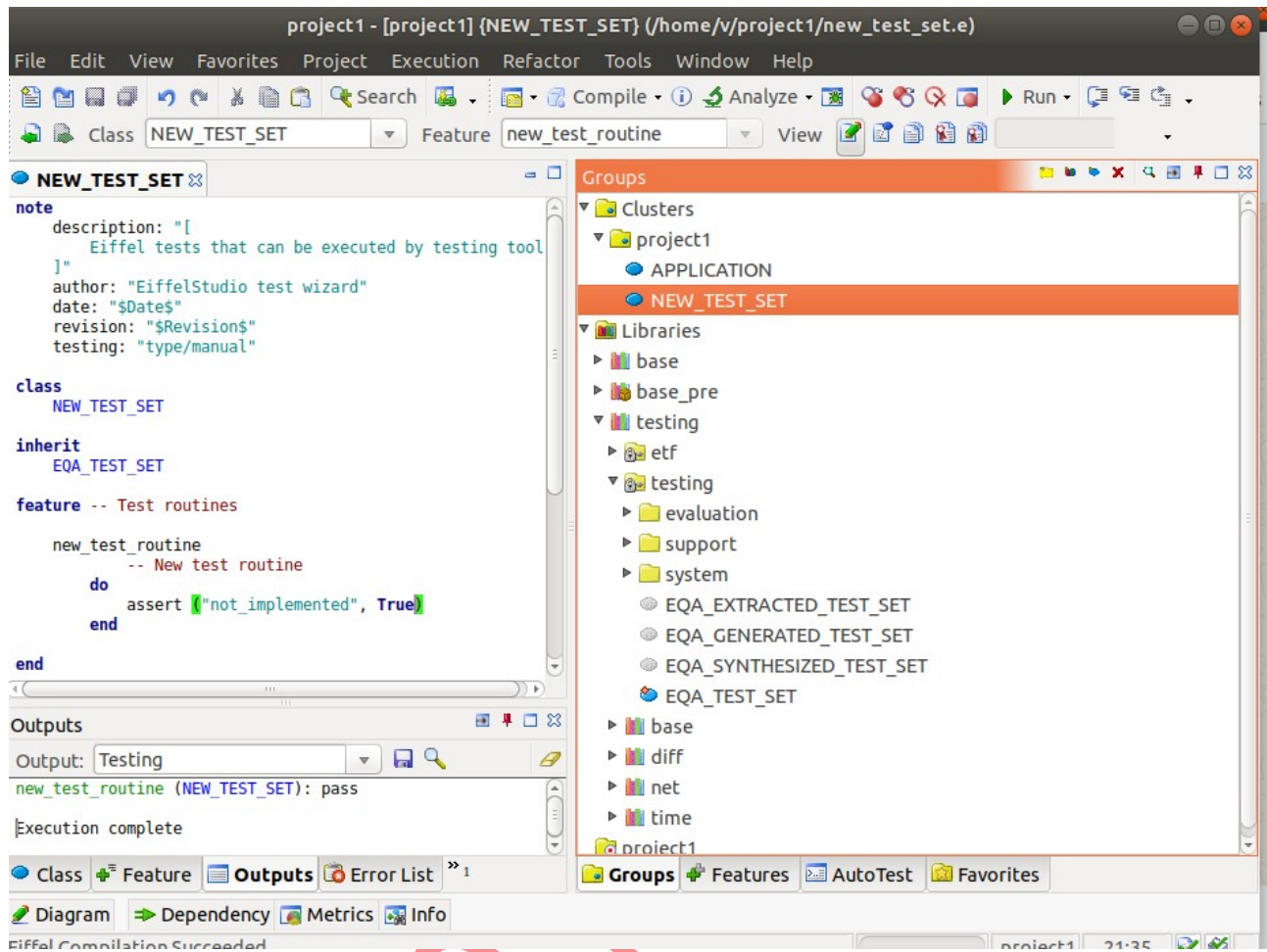
Below the table, the `Run: 1/1` status is shown, along with `Unresolved: 0` and `Fail: 0`. A green progress bar indicates successful execution. The output log shows `✓new_test_routine (NEW_TEST_SET) [0.0010s]`.

A large, semi-transparent red watermark with the word "test" is overlaid diagonally across the bottom half of the image.

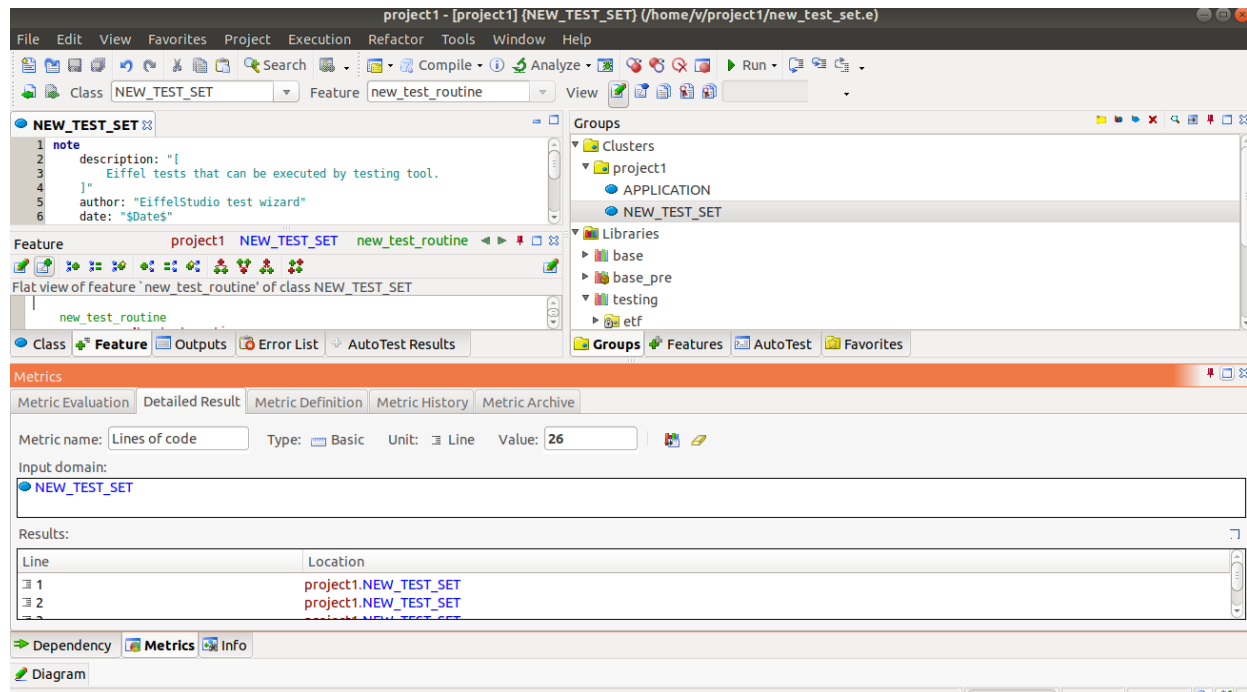
biblioteci



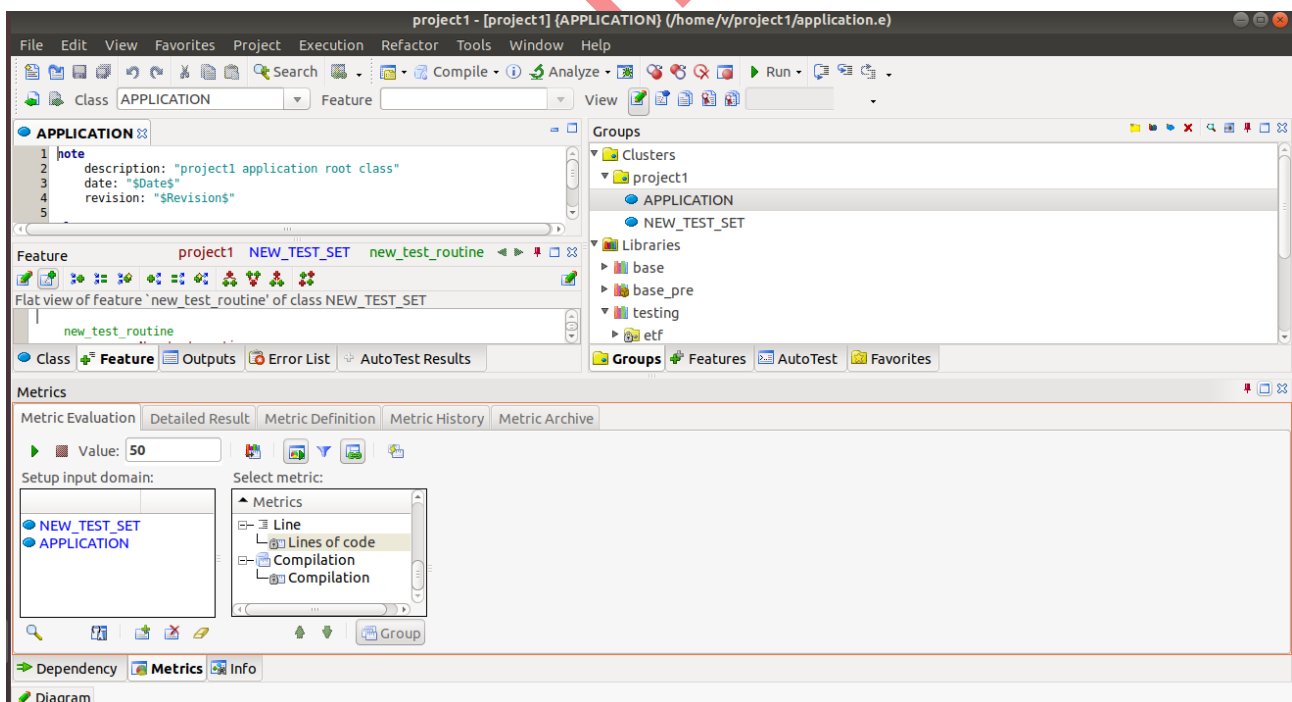
Detalii



Linii de cod



Proiect

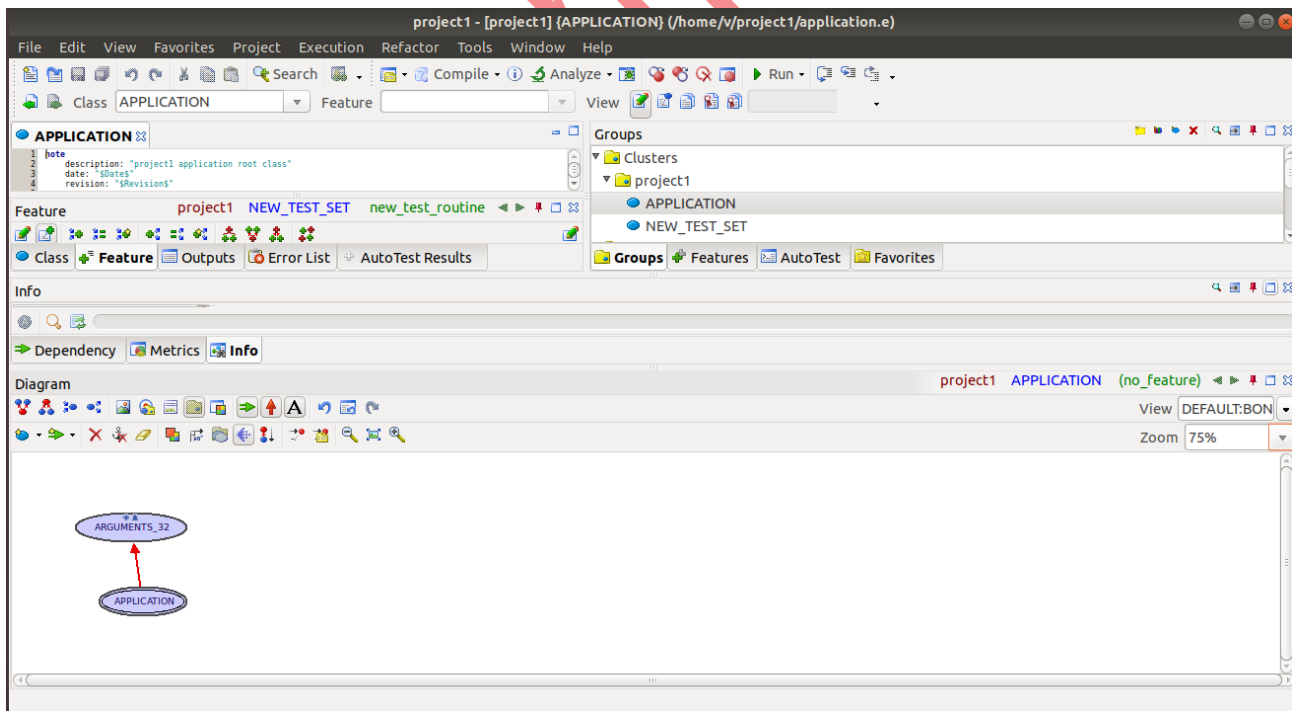


Detaliu metricei LOC

The screenshot shows the 'Metrics' window with the 'Metric Evaluation' tab selected. The metric is 'Lines of code' (Type: Basic, Unit: Line, Value: 50). The input domain is 'NEW_TEST_SET' and 'APPLICATION'. The results are displayed in a table with two columns: 'Line' and 'Location'.

Line	Location
13	project1.APPLICATION
14	project1.APPLICATION
15	project1.APPLICATION
16	project1.APPLICATION
17	project1.APPLICATION
18	project1.APPLICATION
19	project1.APPLICATION
20	project1.APPLICATION
21	project1.APPLICATION
22	project1.APPLICATION
23	project1.APPLICATION
24	project1.APPLICATION
1	project1.NEW_TEST_SET
2	project1.NEW_TEST_SET
3	project1.NEW_TEST_SET
4	project1.NEW_TEST_SET
5	project1.NEW_TEST_SET

detalii BON (Business Object Notation)



Diagram

project1 APPLICATION (no_feature) View DEFAULT:BON Zoom 50%

```
graph LR; ANY((ANY)) --> EXCEPTIONS((EXCEPTIONS)); EXCEPTIONS --> TYP_SHARED_CONSTANTS((TYP_SHARED_CONSTANTS)); TYP_SHARED_CONSTANTS -- "retrieved from medium" --> ERL_CONSTRAINTS((ERL_CONSTRAINTS)); ERL_CONSTRAINTS --> EXECUTION_ENVIRONMENT((EXECUTION_ENVIRONMENT)); EXECUTION_ENVIRONMENT --> SED_STORAGE_FACILITIES((SED_STORAGE_FACILITIES)); SED_STORAGE_FACILITIES --> EOL_EXTERNALS((EOL_EXTERNALS)); EOL_EXTERNALS --> ANY; ARGUMENTS_32((ARGUMENTS_32)) -- "arguments" --> EXECUTION_ENVIRONMENT; APPLICATION((APPLICATION)) --> ARGUMENTS_32; EIP_INTERPRETER((EIP_INTERPRETER)) --> ARGUMENTS_32; EIP_INTERPRETER --> ERL_CONSTRAINTS; EIP_INTERPRETER --> EOL_EXTERNALS;
```

teste ma