**QA – Teste de Mutação**

O Teste de Mutação é um teste baseado em Defeitos. Utiliza o conhecimento que há sobre os defeitos típicos que podem ocorrem ao escrever um programa. Isso nos auxilia a criar bons Casos de Teste.

Para utilizarmos o Teste de Mutação, devemos criar vários cenários de programas, com pequenas alterações ao programa a ser testado. Os programas “alterados” deverão apresentar falhas. Caso isso não ocorra, o Caso de Teste desse programa não está bem estipulado. Quando é efetuado um Teste de Mutação em um programa, e o teste apresenta erros, diz-se que o mutante foi “morto”.

Um bom Caso de Teste consegue identificar erros na maioria dos Testes de Mutação realizados. Quanto mais testes de mutação forem realizados, melhor será a qualidade do Caso de Teste.

**Exemplo:**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Operadores de Mutação:** Regras que definem quais alterações devem ser feitas para que os mutantes sejam criados:

Diagrama

Descrição gerada automaticamente

No exemplo abaixo, iremos testar mutações no operador em destaque:

Texto

Descrição gerada automaticamente

Devemos alterar o operador “>” em destaque para “>=”, “<”, “<=”, “==” e “!=”. Cada mutação irá resultar em algum tipo de erro.

No exemplo acima, quando alteramos o operador “>” em destaque para “>=”, há grande possibilidade de que não apareçam erros. Esse tipo de mutação é chamado de **Mutante Equivalente**.

Texto, Carta

Descrição gerada automaticamente

Quanto há o Mutante Equivalente, o “Escore de Mutação” deverá ser alterado, conforme abaixo:

Texto

Descrição gerada automaticamente

**Resumo:**

Diagrama

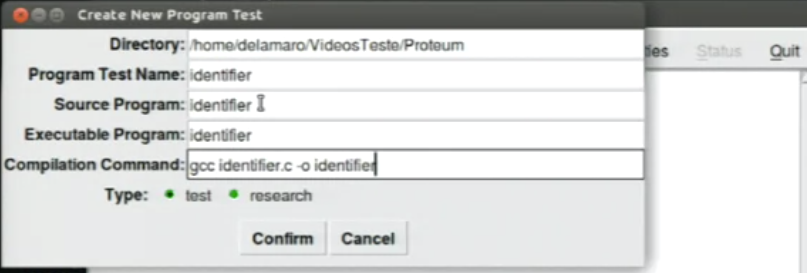
Descrição gerada automaticamente

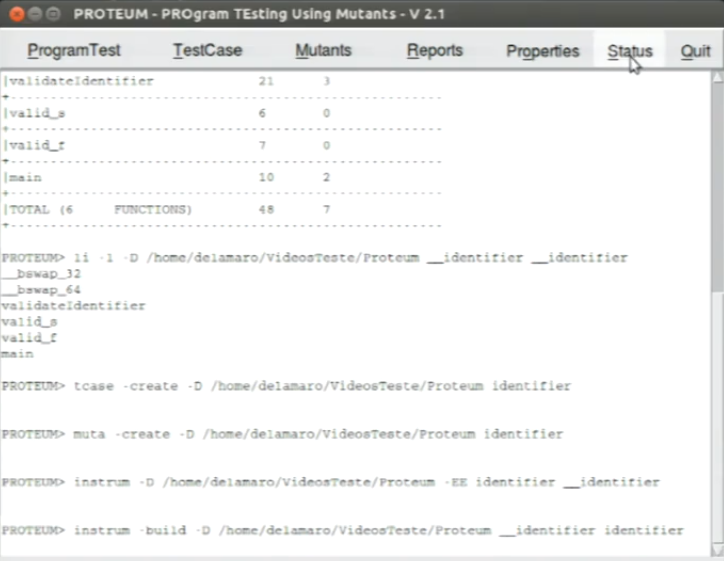
Devemos performar todos os testes, afim de “matarmos” todos os Testes Mutantes possíveis, restando, apenas, os Mutantes Equivalentes.

Há diversas ferramentas que podem nos auxiliar a performar Testes de Mutação. Iremos utilizar a ferramenta **PROTEUM**, que dá suporte à linguagem C/C++.

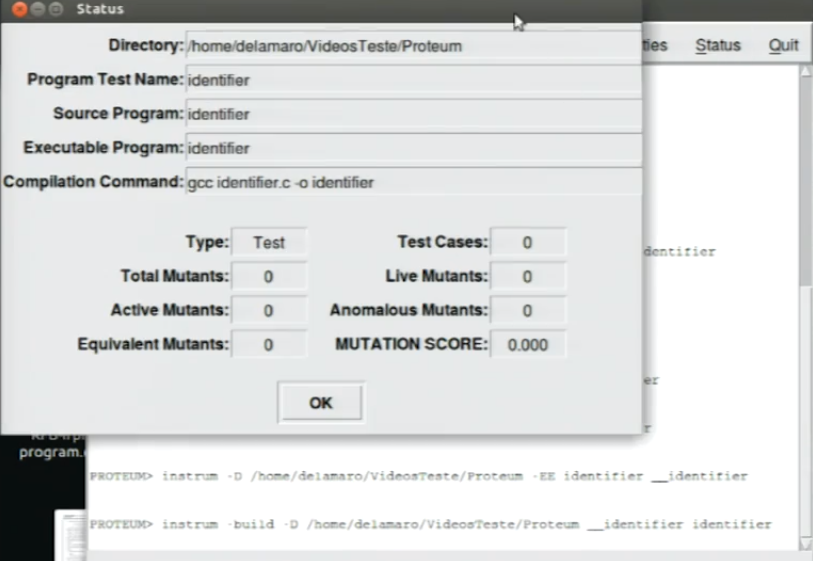
- PROTEUM – <http://ccsl.icmc.usp.br/pt-br/projects/proteum>.

**Utilizando o Sistema PROTEUM, baseado no exemplo *Identifier*:**

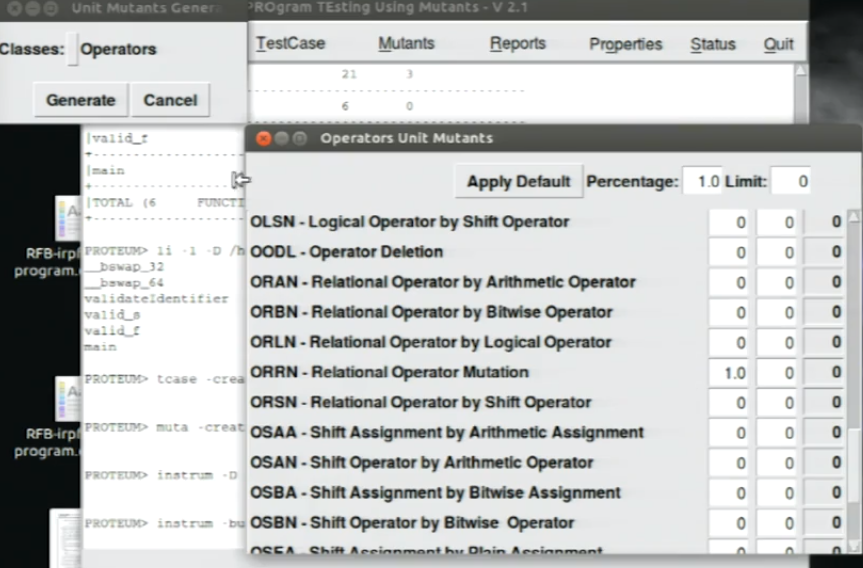




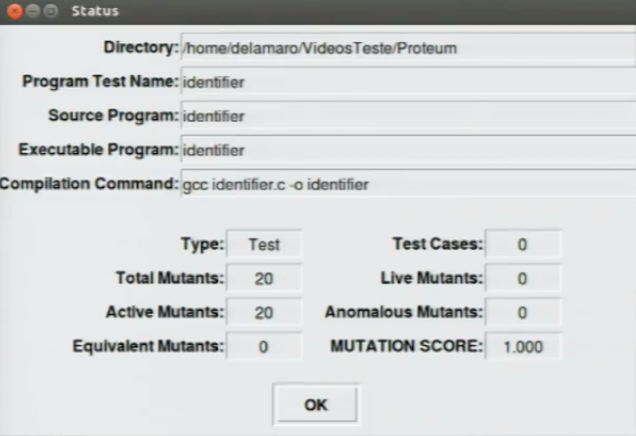
**Status – Acompanhamento dos Testes**



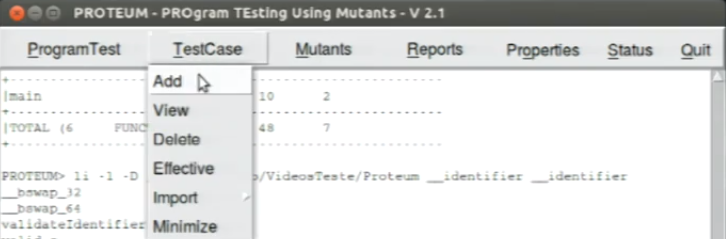
**Escolhendo o ORRN:**

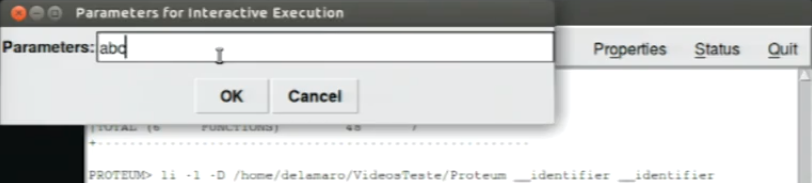


**Abaixo, verifica-se, no Status, que foram gerados 20 mutantes:**

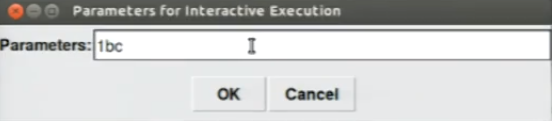


**Adicionando parâmetro válido:**

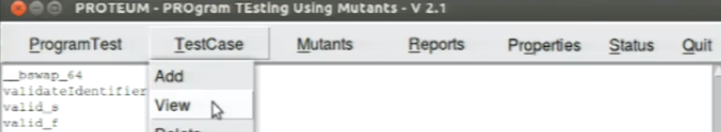
****

****

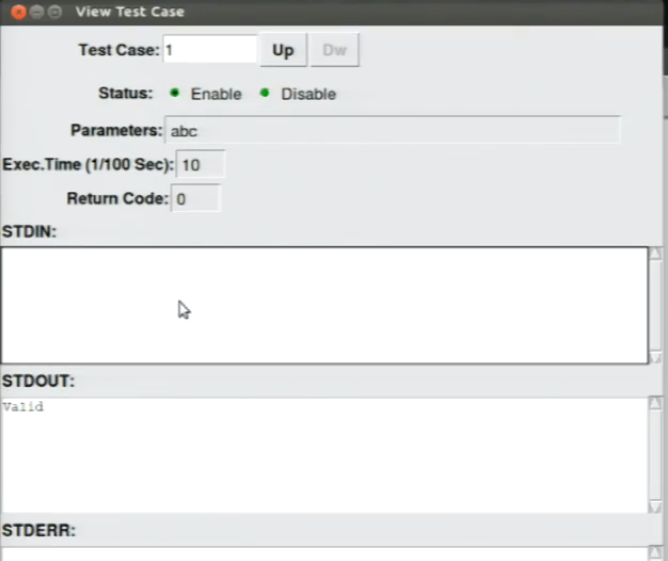
**Adicionando parâmetro inválido:**

****

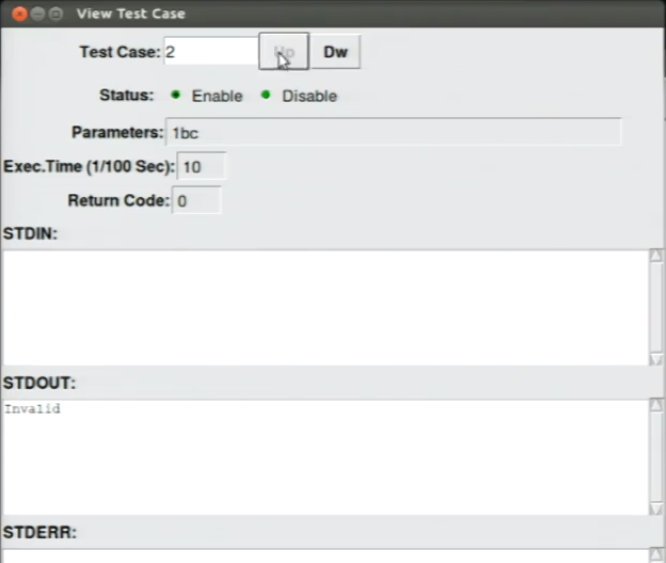
**Verificando os casos de teste no PROTEUM:**

****

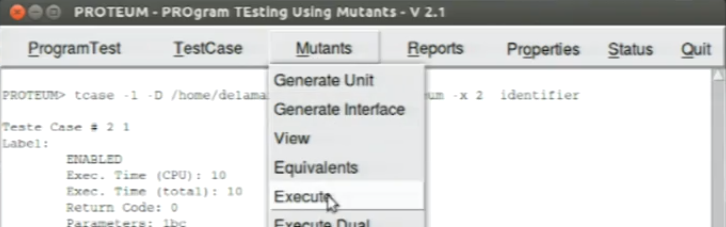
**Resultado Válido:**

****

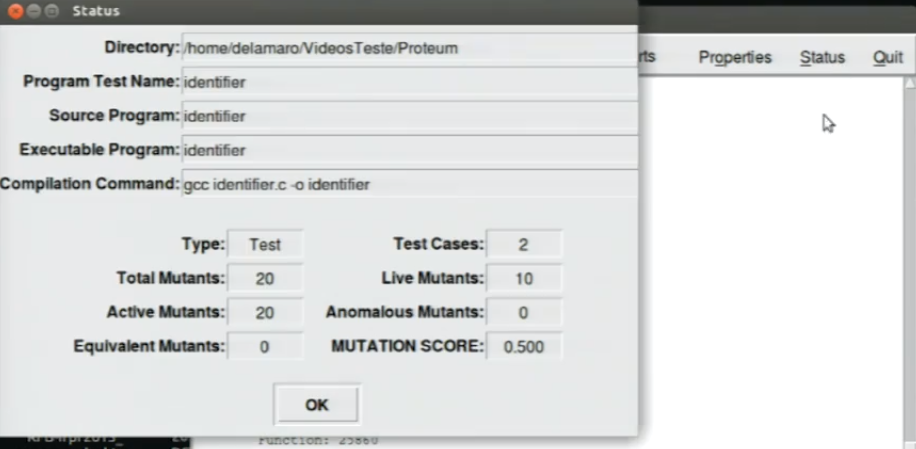
**Resultado Inválido:**

****

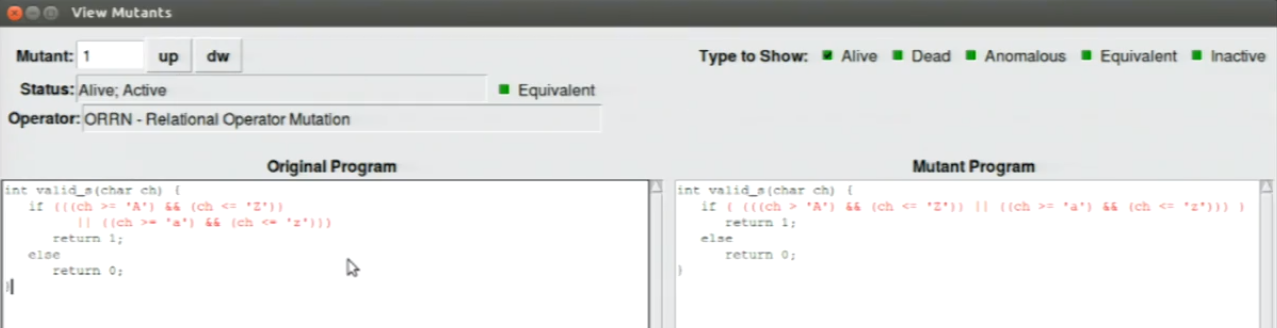
**Executando as Mutações:**

****

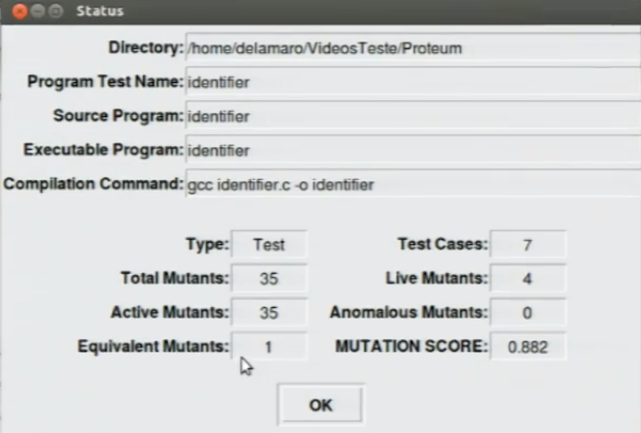
**Verificando o status após a execução das mutações:**

****

**Tela para verificar os mutantes:**

****

**Inserindo mutações equivalentes, verificamos que o status se altera:**

****

**Resumo – Esquema de Teste de Mutação:**

Diagrama

Descrição gerada automaticamente