

Enunciado do Trabalho da Disciplina DCC168 Teste de Software

Primeiramente, selecione um dos sistemas abaixo para aplicar as Técnicas de Teste Funcional, Estrutural e Baseada em Erros e as ferramentas relacionadas, abordadas nas aulas em laboratório:

- Pong Game: <https://github.com/ArthurK12/PongGame>;
- Sistema de Gestão de Biblioteca: <https://github.com/pk7856/LibraryManagementSystem>;
- Sistema de Gestão de Recursos Humanos:
<https://github.com/Only1Ryu/HumanResourceManagementSystem>
- JHotDraw: <https://sourceforge.net/projects/jhotdraw/>;
- BVR Tool: <https://github.com/SINTEF-9012/bvr>;

O trabalho da disciplina está dividido em três fases: *i*) Teste Funcional, *ii*) Teste Estrutural e *iii*) Teste baseado em Defeitos. As técnicas e critérios de teste deverão ser aplicados de forma incremental, seguindo a ordem: 1º - Teste Funcional, 2º - Teste Estrutural e 3º - Teste baseado em Defeitos.

Fase I: Aplicação do Teste Funcional

As atividades de Teste Funcional deverão ser realizadas com base na Especificação do Sistema (documentação) e na Execução do Sistema, ou seja, com base na interação do usuário com a interface gráfica para exercitar cada função fornecida pelo sistema. Considere os critérios de Particionamento por Classes de Equivalência e Análise do Valor Limite para testar funcionalidades que possuam variáveis de entrada que incluam restrições.

Na Tabela 1 são mostrados exemplos de classes de equivalência válidas e inválidas para o primeiro elemento de um identificador do programa **Identifier**. Recomenda-se especificar as classes de equivalência relacionadas a cada variável de entrada de uma função do sistema da forma como descrito no exemplo da Tabela 1.

Tabela 1 – Classes de equivalência.

Variáveis de Entrada	Classes de Eq. Válidas	Classes de Eq. Inválidas
Primeiro elemento	A-Z (V1), a-z (V2)	0-9 (I1), caracteres especiais, e.g., ?, !, *, \$, # (I2)

Especifique casos de teste para a cobertura de cada classe de equivalência identificada. Utilize o modelo da Tabela 2 para especificar os casos de teste e suas relações com as classes de equivalência.

Tabela 2 – Casos de teste e classes de equivalência.

Variáveis de Entrada			Saída Esperada	Classes de Eq. Cobertas
Var1	Var2	Var3		

Aplique o Grafo de Causa e Efeito para identificar casos de teste relacionados a potenciais restrições de valores de duas ou mais variáveis que possuem alguma relação. Por exemplo, as variáveis “dias” e “mês” do programa Calendário são fortemente relacionadas. Dessa forma, o valor da variável “mês” implica no número de “dias”, e.g., “mês” igual a 6 implica em número de “dias” igual a 30, “mês” igual 7 implica em número de “dias” igual a 31. Especifique os casos de teste e suas relações com os caminhos de execução do programa identificados com a aplicação do Grafo de Causa e Efeito para uma funcionalidade do sistema considerada complexa. Utilize o modelo da Tabela 2 para especificar os casos de teste e suas relações com caminhos de execução do programa cobertos pelos casos de teste.

Os casos de teste identificados, chamados “**TestSet-Func**” (Tabela 3), deverão ser implementados utilizando o JUnit. Ao final da execução dos testes no JUnit efetue a análise de cobertura dos casos de teste utilizando a ferramenta EclEmma e anote as informações sobre a cobertura em uma tabela (exemplo Tabela 3).

Tabela 3 – Casos de Teste para JUnit para o Teste Funcional.

Programa/ Método	Tamanho TestSet-Func.	Cobertura TestSet no EclEmma	Tamanho TestSet- Estrutural	Cobertura TestSet- Estrutural	Tamanho TestSet-Mut	Cobertura TestSet-Mut

Obs: a Tabela 3 pode ser estendida com informações por método e com informações sobre cada critério na ferramenta EclEmma (analise a tabela de resultados de análise de cobertura gerada pela ferramenta EclEmma). Não é permitido acrescentar casos de teste que não sejam os funcionais nesta fase. Caso sejam identificados defeitos no código, descrever no relatório quais são os defeitos e como podem ser corrigidos.

Fase II: Aplicação do Teste Estrutural

Aplique os critérios de Teste Estrutural baseados em fluxo de dados e de controle para testar ao menos três métodos considerados complexos. Utilize a ferramenta de apoio ao teste estrutural EclEmma para realizar a análise de cobertura após a execução dos casos de teste funcional anteriormente identificados.

Elabore o(s) grafo(s) de fluxo de controle do(s) programa(s)/método(s) testado(s) anteriormente e derive **novos** casos de teste para melhorar a cobertura do estrutural, ou seja, em relação aos critérios Todos-Nós, Todos-Arcos, Todos os caminhos, Todas-Definições, Todos-Usos, gerando o conjunto de casos de teste “**TestSet-Estrutural**”. O objetivo desta fase é obter 100% de cobertura para os critérios baseados em fluxo de dados e baseados em

fluxo de controle. O relatório de análise de cobertura gerado pela ferramenta EcIEmma deve ser apresentado no relatório.

Fase III: Aplicação do Teste Baseado em Defeitos

Utilize a ferramenta PITest (disponível em <http://pitest.org/>), de apoio ao Teste de Mutação, execute os casos de teste gerados nas **Fases I e II**, ou seja, casos de teste adequados aos critérios do Teste Funcional e Estrutural e avalie a cobertura. Em seguida, adicione novos casos de teste para melhorar a cobertura em relação ao critério de Teste de Mutação, gerando um novo conjunto de casos de teste “**TestSet-Mut**”. O objetivo desta última fase é definir casos de teste necessários para atingir um escore de 100% para o Teste de Mutação.

Obs: Considere os operadores de mutação disponíveis na ferramenta PITest. Gere o relatório da ferramenta PITest com os resultados obtidos e inclua no relatório do trabalho. Caso sejam identificados defeitos no código nesta fase, descrever no relatório quais são os defeitos e como podem ser corrigidos.

Cada grupo deve elaborar uma apresentação de no máximo 4 slides, dois com a descrição do processo de desenvolvimento do projeto e outros 2 com comentários sobre os resultados obtidos, dificuldades e lições aprendidas.

O que entregar: um arquivo “**.zip**” contendo o relatório, os fontes do projeto com os respectivos casos de teste que atendam aos critérios funcionais, estruturais e baseado em defeitos e a apresentação de slides.

Data de Entrega: 02/12/2019 via Moodle.