

Especificação do Projeto da Disciplina

Para ser desenvolvido em dupla.

O projeto envolve a realização da atividade de teste considerando a aplicação das 3 técnicas de teste (funcional, estrutural e baseada em defeitos), por meio de uma estratégia incremental de aplicação e usando ferramentas de teste.

1ª parte: Escolha do projeto de software

A primeira atividade do projeto envolve a escolha de um projeto de software, de preferência que você tenha desenvolvido (projeto pessoal ou feito no escopo de alguma disciplina). Caso não tenha um projeto adequado, no final deste documento deixamos link de projetos que podem ser escolhidos. O importante é que o projeto contenha algumas características:

1. Dê um nome ao seu projeto;
2. Desenvolvido na linguagem Java (importante para uso das ferramentas de teste);
3. Seja um software com um bom propósito (por exemplo, um jogo feito, aplicação, etc). Não seria adequado um sistema que só implemente uma árvore binária por exemplo e que não tenha uma finalidade para existência;
4. Tenha um grau de complexidade razoável para que seja interessante testar. Neste sentido, é interessante analisar métricas como complexidade ciclomática, LOC (linhas de código) dentre outras. Indicamos o uso do plugin do eclipse - Metrics, informações de uso do plugin podem ser encontradas no link a seguir: <http://metrics2.sourceforge.net/>.

O projeto escolhido será analisado pelos responsáveis da disciplina para definir se estão adequados. Para isso, a dupla deve entregar um texto que especifica detalhes do projeto, como: quando foi desenvolvido, contexto de desenvolvimento (p.ex., atividade de disciplina) e as funcionalidades do projeto. Incluir informações referentes às métricas do código (complexidade ciclomática e LOC) e outras que julgar adequadas, como: número de métodos, número de classes, as quais são retornadas pela ferramenta que tenha sido utilizada para coleta das métricas de código.

Data de entrega (1ª parte): 20/05 (pdf a ser enviado no edisciplinas).

2ª Parte: Aplicação Incremental das técnicas e critérios de teste

O teste será aplicado de maneira incremental, seguindo a ordem Funcional -> Estrutural -> Baseado em Defeitos.

Um documento pdf deverá ser entregue com informações sobre essa aplicação, anotando-se as coberturas parciais dos casos de teste. Nesse documento uma tabela deve ser incluída com informações sobre as coberturas dos testes gerados:

Programa ou Método	Tamanho TestSet-Func	Cobertura TestSet-Func na EclEmma	Tamanho TestSet-Estr	Cobertura TestSet-Estr PIT	Tamanho TestSet-Mut	Cobertura Final TesSet-Mut PIT
xx						

A tabela pode ser estendida com informações por métodos e com informações de cada critério da EclEmma.

2.1) Aplicação do Teste Funcional

Aplicar os critérios de teste funcionais Particionamento em Classes de Equivalência e Análise do Valor Limite no software escolhido. Essa aplicação deve ser feita com base na especificação do software. Preparar um documento com as informações da tabela de equivalência gerada e da aplicação dos dois critérios. Escrever os casos de teste gerados neste documento, os quais correspondem ao conjunto de teste TestSet-Func.

Implementar os casos de testes gerados utilizando a Junit. Ao final meça a cobertura, utilizando a ferramenta EclEmma, anotando a informação de cobertura na tabela.

Importante: Não é permitido o acréscimo de outros casos de teste nesta fase.

Se defeitos forem identificados, apresentar (no documento) quais e como podem ser corrigidos.

2.2) Aplicação do Teste Estrutural

Por meio do uso da ferramenta de teste estrutural EclEmma, execute os casos de teste gerados anteriormente (casos de teste adequados ao teste funcional) e avalie a cobertura. A seguir, adicione novos casos de teste de modo a melhorar a cobertura do teste estrutural, gerando o conjunto de teste TestSet-Estr. O objetivo é definir casos de teste para conseguir atingir 100% de cobertura para os critérios estruturais.

Considerar os critérios disponíveis na EclEmma para avaliar a cobertura (ver slides com informações sobre EclEmma).

Gerar relatório da EclEmma com os resultados obtidos para entregar junto com a atividade.

Se defeitos forem identificados, apresentar (no documento) quais e como podem ser corrigidos.

2.3) Aplicação do Teste Baseado em Defeitos

Por meio do uso da ferramenta de teste de mutação PIT, execute os casos de teste gerados anteriormente (casos de teste adequados ao teste funcional e ao teste estrutural) e avalie a cobertura. A seguir, adicione novos casos de teste de modo a melhorar a cobertura do teste de mutação, gerando o conjunto de teste TestSet-Mut. O objetivo é definir casos de teste para conseguir atingir escore de 100% para o teste de mutação.

Considerar os operadores de mutação disponíveis na PIT para avaliar a cobertura (<http://pittest.org/>).

Gerar relatório da PIT com os resultados obtidos para entregar junto com a atividade.

Se defeitos forem identificados, apresentar (no documento) quais e como podem ser corrigidos.

Data de entrega (2ª parte): 14/06 (enviar zip com arquivos da atividade no edisciplinas).

Apresentação da atividade: 17/06 e 24/06

- **12 equipes - 5 min de apresentação + 5 min de perguntas;**
- Preparar uma apresentação com 2 slides:
- (slide1) Mostrar como foi o processo de desenvolvimento do projeto;
- (slide2) Comentar os resultados as dificuldades e lições aprendidas;

Projetos Sugeridos no Github:

- <https://github.com/ArthurK12/PongGame>

- <https://github.com/pk7856/LibraryManagementSystem>

- <https://github.com/Only1Ryu/HumanResourceManagementSystem>

- <https://github.com/Akpandita/Student-management-system/tree/master/Student%20portal>

-

<https://github.com/vwadhwa3/bank-management-system/tree/master/src/bank%20management%20System>

- <https://github.com/tonoyadmit/Hospital-Management-System>

- https://github.com/wxqSnowing/Diary_Management