

INFORMAÇÕES SOBRE A AVALIAÇÃO

LAB02	Laboratório 02 - 20 pontos
-------	----------------------------

INFORMAÇÕES DOCENTE						
CURSO: ENGENHARIA DE SOFTWARE	DISCIPLINA: LABORATÓRIO DE EXPERIMENTAÇÃO DE SOFTWARE	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA: 6º
					x	
PROFESSOR (A): João Paulo Carneiro Aramuni						

Um estudo das características de qualidade de sistema java

Introdução

No processo de desenvolvimento de sistemas open-source, em que diversos desenvolvedores contribuem em partes diferentes do código, um dos riscos a serem gerenciados diz respeito à evolução dos seus atributos de qualidade interna. Isto é, ao se adotar uma abordagem colaborativa, corre-se o risco de tornar vulnerável aspectos como modularidade, manutenibilidade, ou legibilidade do software produzido. Para tanto, diversas abordagens modernas buscam aperfeiçoar tal processo, através da adoção de práticas relacionadas à revisão de código ou à análise estática através de ferramentas de CI/CD.

Neste contexto, o objetivo deste laboratório é analisar aspectos da qualidade de repositórios desenvolvidos na linguagem Java, correlacionando-os com características do seu processo de desenvolvimento, sob a perspectiva de métricas de produto calculadas através da ferramenta [CK](#).

Metodologia

1. Seleção de Repositórios

Com o objetivo de analisar repositórios relevantes, escritos na linguagem estudada, coletaremos os **top-1.000** repositórios Java mais populares do GitHub, calculando cada uma das métricas definidas na Seção 3.

2. Questões de Pesquisa

Desta forma, este laboratório tem o objetivo de responder às seguintes questões de pesquisa:

RQ 01. Qual a relação entre a popularidade dos repositórios e as suas características de qualidade?

RQ 02. Qual a relação entre a maturidade do repositórios e as suas características de qualidade ?

RQ 03. Qual a relação entre a atividade dos repositórios e as suas características de qualidade?

RQ 04. Qual a relação entre o tamanho dos repositórios e as suas características de qualidade?

3. Definição de Métricas

Para cada questão de pesquisa, realizaremos a comparação entre as características do processo de desenvolvimento dos repositórios e os valores obtidos para as métricas definidas nesta seção. Para as métricas de **processo**, define-se:

- Popularidade: número de estrelas
- Tamanho: linhas de código (LOC) e linhas de comentários
- Atividade: número de releases
- Maturidade: idade (em anos) de cada repositório coletado

Por métricas de **qualidade**, entende-se:

- CBO: Coupling between objects
- DIT: Depth Inheritance Tree
- LCOM: Lack of Cohesion of Methods

4. Coleta e Análise de Dados

Para análise das métricas de popularidade, atividade e maturidade, serão coletadas informações dos repositórios mais populares em Java, utilizando as APIs REST ou GraphQL do GitHub. Para medição dos valores de qualidade, utilizaremos uma ferramenta de análise estática de código (por exemplo, o [CK](#)).

Importante: a ferramenta CK gera diferentes arquivos .csv com os resultados para níveis de análise diferentes. É importante que você sumarie os dados obtidos.

Relatório Final:

Para cada uma das questões de pesquisa, faça uma sumarização dos dados obtidos através de valores de medida central (mediana, média e desvio padrão), por **repositório**.

Mesmo que de maneira informal, elabore hipóteses sobre o que você espera de resposta e tente analisar a partir dos valores obtidos.

Elabore um documento que apresente (i) uma introdução simples com hipóteses informais; (ii) a metodologia que você utilizou para responder às questões de pesquisa; (iii) os resultados obtidos para cada uma delas; (iv) a discussão sobre o que você esperava como resultado (suas hipóteses) e os valores obtidos.

Na aula de entrega, os grupos deverão apresentar os seus resultados.

Bônus (+1 ponto):

Para melhor analisar a correlação entre os valores obtidos em cada questão de pesquisa, gere gráficos de correlação que permitam visualizar o comportamento dos dados obtidos. Adicionalmente, utilize um teste estatístico que forneça confiança nas análises apresentadas (por exemplo, teste de correlação de **Spearman** ou de **Pearson**).

Processo de Desenvolvimento:

Sprints e Entregas

Lab02S01: Lista dos 1.000 repositórios Java + Script de Automação de clone e Coleta de Métricas + Arquivo .csv com o resultado das medições de 1 repositório (5 pontos)

Lab02S02: Arquivo .csv com o resultado de todas as medições dos 1.000 repositórios + hipóteses (5 pontos) Análise e visualização de dados + elaboração do relatório final (10 pontos)

Prazo final (Acesse o cronograma): <https://github.com/joaopauloaramuni/laboratorio-de-experimentacao-de-software/tree/main/CRONOGRAMA>

Valor total: 20 pontos