Lab 4 sprint 1

Alunos: Estevao de faria, Henrique Lobo
Professor: Danilo Quadros

Caracterização do Dataset

O objeto de estudo deste trabalho são os repositórios de código aberto do GitHub. A metodologia de coleta visa criar dois grupos distintos para comparação, com base na frequência de suas *releases*:

- **Rapid Release Cycle (RRC):** Repositórios com ciclos de *release* curtos, definidos como um intervalo entre 5 e 35 dias.

- **Slow Release:** Repositórios com ciclos de *release* longos, definidos como um intervalo superior a 60 dias.

A criação dos *datasets* seguiu um funil de duas etapas, executado pelo script principal (`research_automation_script.py`):

1. **Coleta Bruta:** Inicialmente, foi realizada uma busca na API do GitHub (conforme implementado em `src/get_repos_info.py`) para obter uma lista de repositórios populares, baseando-se em critérios como número de estrelas e *forks*.

2. **Filtragem e Classificação:** Os repositórios coletados foram processados pelo script `src/filterRepos.py`. Este script calcula o tempo médio entre as *releases* de cada projeto e os classifica em um dos dois grupos (RRC ou Slow Release), descartando aqueles que não se enquadram nos critérios.

Os repositórios válidos, juntamente com seus metadados (como *commits*, *issues* e *pull requests*), são então armazenados em um banco de dados PostgreSQL para facilitar a extração de métricas.

1. Contexto do Dataset

Característica	**Detalhe**
Objeto de Estudo	Repositórios de código aberto do GitHub.
Critério de Classificação	Frequência de *releases* (ciclos curtos vs. longos).

Grupos de Comparação	**Rapid Release Cycle (RRC):** Intervalo entre releases entre 5 e 35 dias. **Slow Release:** Intervalo entre releases superior a 60 dias.
Métodos de Coleta	API do GitHub (para metadados) e Análise Estática de Código (SonarQube para métricas de qualidade).
Exemplos de Repositórios Usados	**RRC:** vuejs/vue (249 releases) **Slow Release:** facebook/react (106 releases)

2. Visão Geral das Métricas Coletadas

As métricas coletadas incluem dados de popularidade do GitHub e métricas de qualidade de código do SonarQube.

Tipo de Métrica	Métricas Relevantes	Origem (Arquivo de Referência)
Popularidade/Tamanho	Estrelas (stars / stargazerCount), Forks (forks / forkCount), Linhas de Código (ncloc), Número de Releases (release_count).	sonar_analysis_20251027_022930.csv, jsonFiltrado_filtered.json
Qualidade do Código	Bugs, Vulnerabilidades, Code Smells (normalizadas por 1000 NCLOC).	sonar_analysis_20251027_022930.csv, summary_20251027_022930.json
Complexidade	Complexidade Ciclomática (complexity), Complexidade Cognitiva	sonar_analysis_20251027_022930.csv

	(cognitive_complexity).	
Outras	Densidade de Linhas Duplicadas (duplicated_lines_density), Cobertura de Testes (coverage).	sonar_analysis_20251027_022930.csv

3. Estatísticas Descritivas do Dataset

Métrica	**Média (avg)**	**Mínimo (min)**	**Máximo (max)**
Linhas de Código (ncloc)	390.038	63.139	1.116.139
Bugs	488	68	1.065
Vulnerabilities	2	0	11
Code Smells	9.814	1.539	23.046
Complexity	49.933	10.963	121.865
Duplication (%)	9,16	-	-