

# Laboratório 3 – Análise Comparativa de Repositórios Python e JAVA

- Comparação de Métricas de Qualidade de Repositórios de Software
- Prática de Laboratório

# Introdução

- No desenvolvimento de software, simplicidade e expressividade são características importantes na escolha de uma linguagem de programação. No contexto em que sistemas precisam ser desenvolvidos de forma ágil, linguagens de simples prototipação e com uma baixa curva de aprendizado ganham cada vez mais espaço no mercado. Entretanto, a balança entre agilidade e qualidade é uma questão relevante neste cenário, uma vez que os desenvolvedores tendem a priorizar entregas rápidas e releases constantes em detrimento de padrões de qualidade mais alto (Prof. Laerte Xavier, 2020).

# Objetivo

- O objetivo deste laboratório é analisar a qualidade de repositórios desenvolvidos na linguagem Python comparando-os com repositórios desenvolvidos na linguagem JAVA sob a perspectiva de características levantadas através de ferramentas como Radon, CodeFactor ou SonarQube.

# Metodologia

1. Escolher a ferramenta de análise estática de código Java e Python. Sugere-se avaliar o conteúdo do repositório <https://github.com/analysis-tools-dev/static-analysis>
2. Com o objetivo de analisar repositórios relevantes, escritos nas linguagens estudadas (Python e Java), coletaremos, ou seja, faremos o *clone*, dos *top-100* repositórios Python e dos *top-100* repositórios Java mais populares do GitHub, para calcular cada uma das métricas definidas a seguir.
3. Produzir um relatório comparando os repositórios mais populares das linguagens Python e Java de acordo com as métricas definidas.

# Questões de pesquisa

- **RQ 01:** Quais as características dos *top-100* repositórios Java mais populares?
- **RQ 02:** Quais as características dos *top-100* repositórios Python mais populares?
- **RQ 03:** Repositórios Java e Python populares possuem características de “boa qualidade” semelhantes?
- **RQ 04:** A popularidade influencia nas características dos repositórios Java e Python?

# Métricas

Utilizaremos como fatores de qualidade métricas associadas à quatro dimensões:

- **Popularidade:** número de estrelas, número de *watchers*, número de *forks* dos repositórios coletados
- **Tamanho:** linhas de código (LOC e SLOC) e linhas de comentários
- **Atividade:** número de releases, frequência de publicação de releases (número de releases / dias)
- **Maturidade:** idade (em anos) de cada repositório coletado

# Coleta e Análise das Métricas (1/2)

- Para análise das métricas de popularidade, atividade e maturidade, serão coletadas informações dos repositórios Java e Python utilizando as APIs REST ou GraphQL do GitHub.
- Para medição dos valores de tamanho, utilizaremos uma ferramenta de análise estática de código (vide passo 1 da metodologia).
- As questões de pesquisa (RQs) 1 e 2 serão respondidas a partir da análise quantitativa de cada uma métricas (recomenda-se a utilização de valores medianos).
- Para as RQs 03 e 04, os valores obtidos nas RQs 01 e 02 devem ser comparados e discutidos.

# Relatório Final

## (proposta do Prof. Laerte Xavier)

- Para cada uma das questões de pesquisa, faça uma sumarização dos dados obtidos através de valores medianos. Mesmo que de forma 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.



# Relatório Final – *bônus (+2 pontos)*

- Apresente uma extensão do comparativo dos repositórios Java e Python considerando as métricas de (i) *complexidade ciclomática*, (ii) *índice de manutenibilidade* e (iii) *Halstead*. Veja como referência como é feito pela ferramenta “radon” em <https://radon.readthedocs.io/en/latest/intro.html>

# *Sprints* de Trabalho Individual (1/3)

- **Sprint 01:** *arquivo .csv com a lista dos top-100 repositórios Java e os top-100 repositórios Python, bem como os scripts de coleta utilizados para mineração e análise dos repositórios.*
  - Valor: 5 pontos
  - Entrega em 30/09/2020 até às 18:30 no Canvas e no SGA

# *Sprints* de Trabalho Individual (2/3)

- **Sprint 02:** *mineração dos repositórios populares Python (.csv com os valores de todas as métricas dos top-100 repositórios, bem como os scripts de coleta utilizados para mineração e análise dos repositórios)*
  - Valor: 7 pontos
  - Entrega em 07/10/2020 até às 18:30 no Canvas e no SGA

# *Sprints* de Trabalho Individual (3/3)

- **Sprint Final:** *Análise de dados + elaboração do relatório final*
  - Valor: 8 pontos
  - Entrega em 14/10/2020 até às 18:30 no Canvas e no SGA

OBS.: Desconto de 0,5 por dia de atraso de entrega