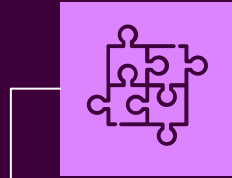


Análise de Repositórios do GitHub

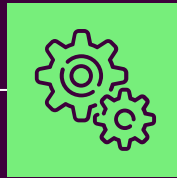
Matheus Vinícius Mota Rodrigues
Gabriel Henrique Mota Rodrigues
João Queiroga

TABLE OF CONTENTS



01

Introdução



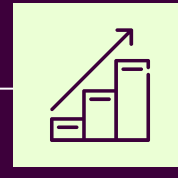
02

Hipóteses



03

Metodologia



04

Resultados

1. Introdução

Este estudo tem como objetivo analisar a qualidade interna de repositórios Java a partir de métricas coletadas com a ferramenta CK. Além das métricas de qualidade (CBO, DIT, LCOM), foram coletados dados de processo no GitHub (popularidade, atividade e maturidade).

```
query
1 q = "language:java"
2 url = f"https://api.github.com/search/repositories?q={q}&sort=stars&order=desc&per_page={PER_PAGE}&page={page}"
3 print(f"GET {url}")
4 r = requests.get(url, headers=HEADERS)
5 if r.status_code != 200:
6     print("Erro na API:", r.status_code, r.text)
7     sys.exit(1)
8 data = r.json()
9 items = data.get("items", [])
10 if not items:
11     print("Nenhum item retornado na página", page)
12     break
```

2. Hipóteses

Antes da análise, levantamos as seguintes hipóteses:

- H0: Repositórios mais populares tenderiam a apresentar maior complexidade (CBO mais alto) devido ao tamanho do código.
- H1: Projetos mais maduros deveriam ter melhor coesão (LCOM menor) por passarem por refatorações ao longo do tempo.
- H2: Projetos mais ativos (mais releases) teriam maior profundidade de herança (DIT) pela evolução contínua.

2. Hipóteses

H0 - Repositórios mais populares tenderiam a apresentar maior complexidade (CBO mais alto) devido ao tamanho do código.

- **A hipótese é rejeitada pelos dados:** A correlação entre popularidade (estrelas) e CBO foi muito fraca (Pearson ≈ -0.11 ; Spearman ≈ 0.03). Isso indica que a popularidade não está associada ao aumento de acoplamento ou complexidade.

2. Hipóteses

H1: Projetos mais maduros deveriam ter melhor coesão (LCOM menor) por passarem por refatorações ao longo do tempo.

- **A hipótese é rejeitada pelos dados:** A maturidade (idade do repositório) apresentou correlação fraca com LCOM (Spearman ≈ 0.20 , positiva e não negativa). Ou seja, projetos mais antigos não mostraram melhoria clara de coesão.

2. Hipóteses

H2: Projetos mais ativos (mais releases) teriam maior profundidade de herança (DIT) pela evolução contínua.

- **A hipótese é parcialmente confirmada pelos dados:** Releases correlacionaram mais fortemente com CBO (Spearman ≈ 0.40) do que com DIT (Spearman ≈ 0.22). Isso sugere que projetos mais ativos tendem a aumentar o acoplamento, mas não necessariamente a profundidade da herança.

3. Metodologia

1. Coleta de dados:

- Utilizamos a API REST do GitHub.
- Foram coletados 1000 repositórios mais populares em número de estrelas.
- Para cada repositório, registramos
 - Nome
 - Quantidade de estrelas
 - Tamanho
 - Idade
 - Número de releases

3. Metodologia

2. Análise CK:

○ Com a ferramenta CK, foram extraídas métricas de classes e sumarizadas por repositório

- CBO (acoplamento entre objetos)
- DIT (profundidade de herança)
- LCOM (falta de coesão)
- LOC (linhas de código)

```
run_ck_on_list.sh

1 process_repo() {
2   full_name="$1" # ex: spring-projects/spring-boot
3   owner=$(echo "$full_name" | cut -d '/' -f1)
4   repo=$(echo "$full_name" | cut -d '/' -f2)
5   target_dir="./clones/${owner}_${repo}"
6   out_dir="${OUTROOT}/${owner}_${repo}"
7
8   mkdir -p "$(dirname "$target_dir")"
9   if [ ! -d "$target_dir/.git" ]; then
10    echo "Clonando $full_name ..."
11    git clone --depth 1 "https://github.com/${full_name}.git" "$target_dir" || { echo
"Clone falhou: $full_name"; return; }
12   else
13    echo "Repositório já clonado: $full_name"
14   fi
15
16   # Verifica se há arquivos .java
17   java_count=$(find "$target_dir" -name "*.java" | wc -l)
18   if [ "$java_count" -eq 0 ]; then
19    echo "Nenhum arquivo Java encontrado em $full_name. Pulando."
20    return
21   fi
22
23   mkdir -p "$out_dir"
24   echo "Rodando CK em $full_name -> $out_dir"
25   java -jar $CK_JAR "$target_dir" false 0 false "$out_dir" "build/" "target/" || {
echo "CK falhou para $full_name"; return; }
26   echo "OK: $full_name"
27 }
```

4. Resultados

Estatísticas descritivas (Exemplo (trecho do resultado_final.csv):

- As métricas foram agregadas em média, mediana e desvio padrão por repositório.

Repositório	CBO_media	DIT_media	LCOM_media	LOC_total	Estrelas	Releases	Idade (anos)
krahets_hello-algo	1.87	1.00	4.70	7,707	117,036	9	2.88
GrowingGit_GitHub-Chinese-Top-Charts	0.00	1.00	0.00	5	102,419	0	6.05
iluwatar_java-design-patterns	3.36	1.20	1.84	35,484	92,713	0	11.12
macrozheng_mall	3.93	1.24	1110.80	100,903	81,625	3	7.47
spring-projects_spring-boot	5.37	1.29	12.61	373,644	78,446	323	12.93

4. Resultados

Correlações Estatísticas:

- Pearson

	CBO_media	DIT_media	LCOM_media	LOC_total	Estrelas	Releases	Idade_anos
CBO_media	1.00	0.29	0.07	0.17	-0.11	0.15	0.00
DIT_media	0.29	1.00	0.07	0.05	-0.11	0.04	0.19
LCOM_media	0.07	0.07	1.00	0.05	0.02	-0.01	0.03
LOC_total	0.17	0.05	0.05	1.00	0.05	0.07	0.12
Estrelas	-0.11	-0.11	0.02	0.05	1.00	0.08	-0.02
Releases	0.15	0.04	-0.01	0.07	0.08	1.00	-0.00
Idade	0.00	0.19	0.03	0.12	-0.02	-0.00	1.00

4. Resultados

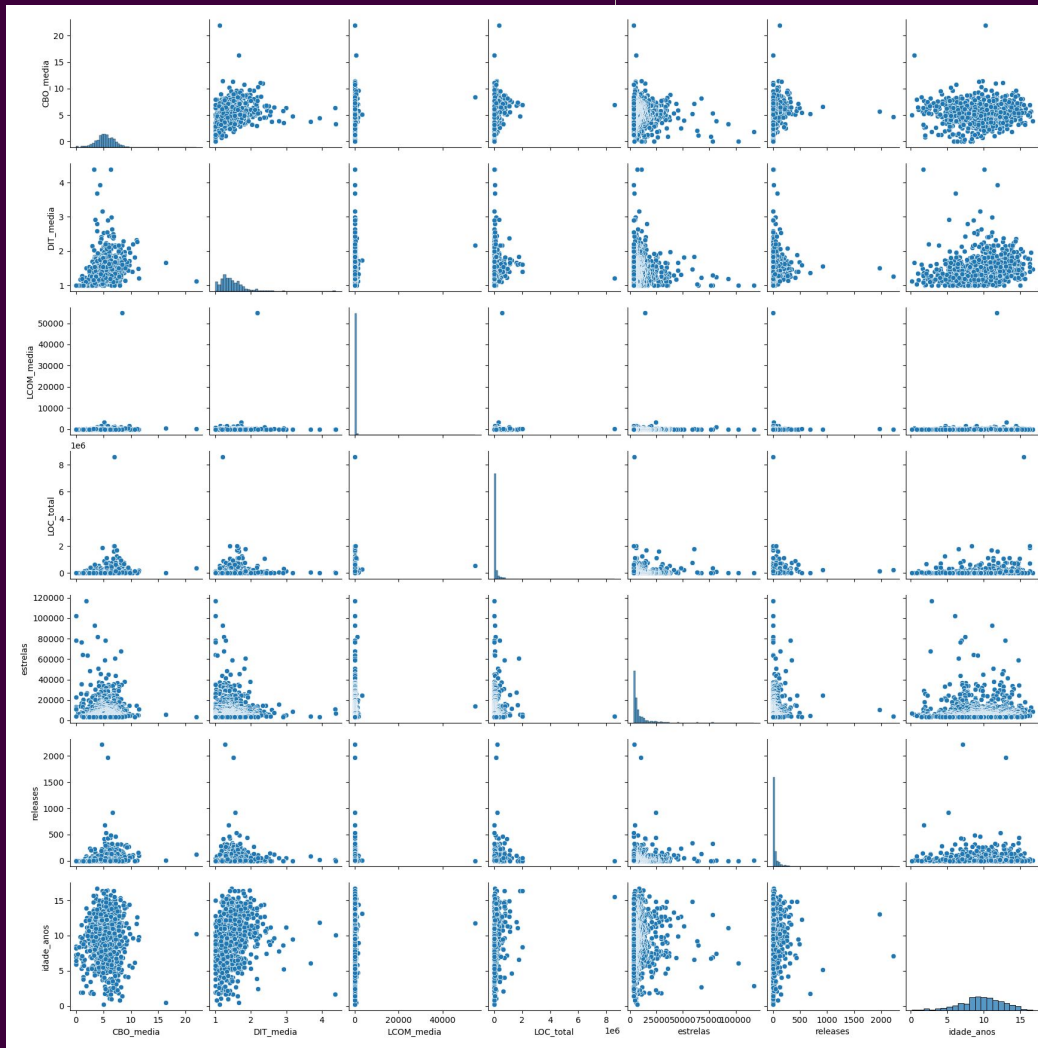
Correlações Estatísticas:

- Spearman

	CBO_media	DIT_media	LCOM_media	LOC_total	Estrelas	Releases	Idade_anos
CBO_media	1.00	0.39	0.37	0.41	0.03	0.40	0.00
DIT_media	0.39	1.00	0.36	0.29	-0.04	0.22	0.29
LCOM_media	0.37	0.36	1.00	0.49	0.04	0.32	0.20
LOC_total	0.41	0.29	0.49	1.00	0.13	0.41	0.13
Estrelas	0.03	-0.04	0.04	0.13	1.00	0.13	0.07
Releases	0.40	0.22	0.32	0.41	0.13	1.00	-0.01
Idade	0.00	0.29	0.20	0.13	0.07	-0.01	1.00

4. Resultados

Visualização dos dados:



4. Resultados

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

Resultado: As correlações foram fracas (Pearson ≈ -0.11 , Spearman ≈ 0.03). Isso mostra que a popularidade (estrelas) não tem relação direta com CBO, DIT ou LCOM.

4. Resultados

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

Resultado: A maturidade apresentou correlação baixa, mas positiva com DIT (Spearman ≈ 0.29), indicando que projetos mais antigos podem adotar hierarquias de herança mais complexas.



4. Resultados

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

Resultado: Releases correlacionaram positivamente com CBO (Spearman ≈ 0.40), sugerindo que projetos mais ativos podem ter maior acoplamento.

4. Resultados

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

Resultado: LOC teve correlação moderada com CBO e LCOM (Spearman ≈ 0.41 – 0.49), confirmando que projetos maiores tendem a apresentar mais acoplamento e menor coesão.



Conclusão

Em suma, a análise realizada indicou que a popularidade de um projeto não exerce influência direta sobre sua qualidade interna. Observou-se que a maturidade do software possui um efeito leve sobre as estruturas de herança utilizadas. Por outro lado, a atividade do projeto apresenta associação com níveis maiores de acoplamento. Finalmente, o tamanho do software mostrou-se fortemente relacionado tanto ao acoplamento quanto à coesão, indicando que projetos maiores tendem a apresentar estruturas mais complexas e interdependentes.