

Felipe Cadar Chamone - 2016006417

## Lab 1 - Clean Code no GitHub

---

1. Quais sistemas adotam práticas de melhoria de código? Para cada sistema, liste as principais labels associadas as práticas (ex: `cleaning_code`, `refactoring`, `quality`, etc).

- Vue: é mais comum usar a tag `ready to merge`, mas a maioria só escreve `refactor(type)` no título do PR, sem adicionar nenhuma tag.
- Django: Possui apenas 6 labels no total mas não usa elas nos PRs de refactor.
- Spring Boot: é mais comum o uso de tags `type: task`, `type: enhancement`. Parece ser o repositório com mais PRs de refactors.

2. Quais os principais problemas resolvidos em issues/commits relacionados a melhoria de código? Apresente três problemas típicos (ex: remover duplicação, remover complexidade, limpar código, etc) juntamente com exemplos concretos

- Problema 1 (Spring Boot): Moving Packages, Issue #10261 #12061
- Problema 2 (Vue): Duplicate Code, Issue #9469 #9006
- Problema 3 (Vue): Dead Code, Issue #10434 #9109 #8348 #8359

3. Selecione e discuta dois exemplos concretos de melhoria de código que julgar interessante dentre os commits/issues analisados.

1. Fix #32317: Refactor loaddata [#13842](#)

Esse refactor é bem interessante pois é extremamente simples e é um passo essencial para, mais tarde, trabalhar na extração de método mais tarde. Ele trata de substituir o uso de um objeto externo a esse método por um local, facilitando a extração de método.

2. Refactor. Stop using memory for list copies. [#2509](#)

Esse PR entra no refactor de Substitute Algorithm. Ele troca o estilo de verificação de objetos em um lista de python para que não precisemos usar list comprehension e assim evitar de copiar memória sem necessidade no python, um ambiente que já é próprio a gastar mais memória do que o necessário.

```
# From:
if base_lang(lang) in [base_lang(trans) for trans in
list(_translations)]:
# To:
if any(base_lang(lang) == base_lang(trans) for trans in
_translations):
```