

# Visualização de flaky tests em projetos de interface do usuário

Rafael Rampim Soratto<sup>1</sup>  
Matheus Batistela<sup>1</sup>  
Alan Patriarca<sup>1</sup>

<sup>1</sup>Universidade Tecnológica Federal Do Paraná (UTFPR) - Brasil

6 de novembro de 2021

- 1 Introdução
- 2 Primeira Visualização
- 3 Segunda Visualização
- 4 Referências

- Tipo de informação do conjunto de dados são identificadores de código do Github em projetos open source. O autor filtrou commits, issues, e pull requests que possuam a string 'flaky' em sua descrição de projetos front-end. Portanto, cada linha do csv trata-se de um identificador de bugs de teste no código fonte (ROMANO et al., 2021).

A categoria deste dataset é textual: suas colunas resembram informações no formato de string. O conjunto de dados apresenta 235 instâncias de flaky tests em projetos web e mobile que utilizam JavaScript e Android. Todos os dados estão disponíveis no link <https://ui-flaky-test.github.io/>.

As informações que os atributos registram são:

- Título;
- URL(LINK);
- Root Cause of Flakiness;
- Root Cause Category;
- Tests/Code Affected (URL);
- Testing Environment/Runner;
- Manifestation Category;
- How Fixed in Code;
- Fix Category;
- Link to Pull Request.

As informações exploradas ou confirmadas por meio das visualizações da sua base de dados são: Causa raíz, efeitos, características em comum e metodologias de resolução para testes fragmentados de interface, e link para a a issue / pull request / commit.

Conforme o slide anterior, entre as colunas do dataset existem algumas categorias que se repetem.

- Root Cause Category;
- Testing Environment/Runner;
- Manifestation Category;
- Fix Category;

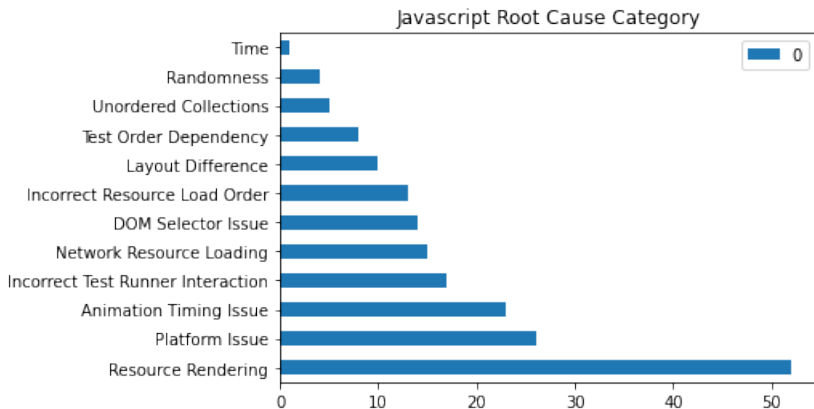
- Para conhecer melhor as categorias de causa e correção de testes fragmentados foi utilizada a técnica de visualização de barras e pizza.
- Foi utilizada a linguagem python em conjunto com as bibliotecas pandas e matplotlib.
- Cada unidade na barra ou na porcentagem da pizza, representa o número de repetições daquela categoria. Desta forma é possível visualizar quais são as categorias mais comuns neste dataset.
- Essa informação é importante para verificar se as principais causas, efeitos e correções de flakys se repetem frequentemente, indicando os ambientes propícios para identificar erros nos testes.

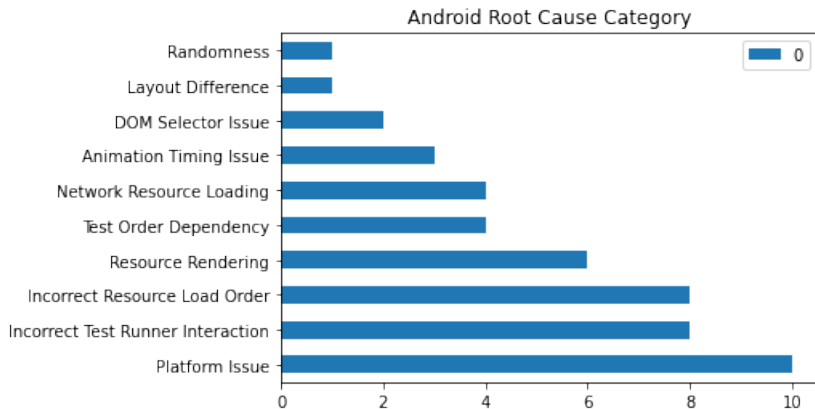
De acordo com as visualizações, foi possível identificar que as seguintes categorias são mais comuns:

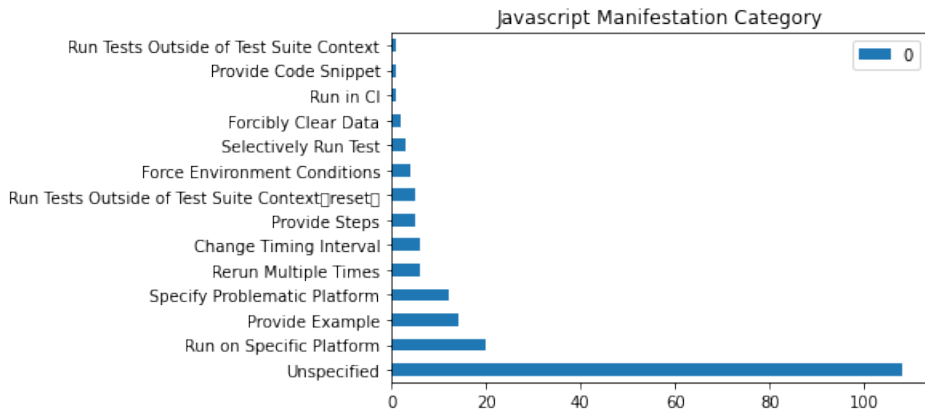
- A raiz da causa mais comum de flakies neste dataset está relacionada a renderização de recursos;
- A forma mais comum do flaky se manifestar no código é desconhecida. A segunda e terceiras formas mais comuns estão diretamente relacionadas com a plataforma utilizada para o teste.
- As maneiras mais comuns de se corrigir um flaky é refatorando o código, alterando a lógica de assíncronicidade ou removendo o teste.
- O ambiente de teste mais comum presente nos flakies é o Circleci.

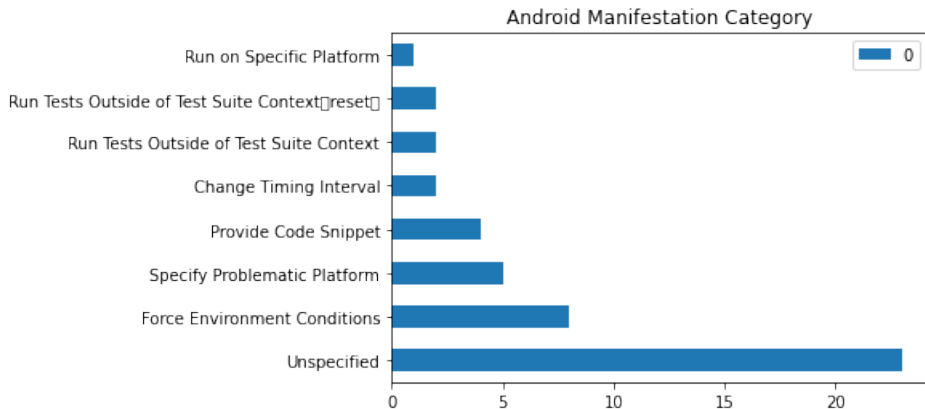
- A primeira visualização foi replicada porém realizou-se um pré-processamento do dataset, dividindo-o em duas partes.
- Uma parte refere-se aos projetos android, e a outra javascript.
- Desta maneira é possível comparar as causas mais comuns em ambas linguagens e maneira separada.

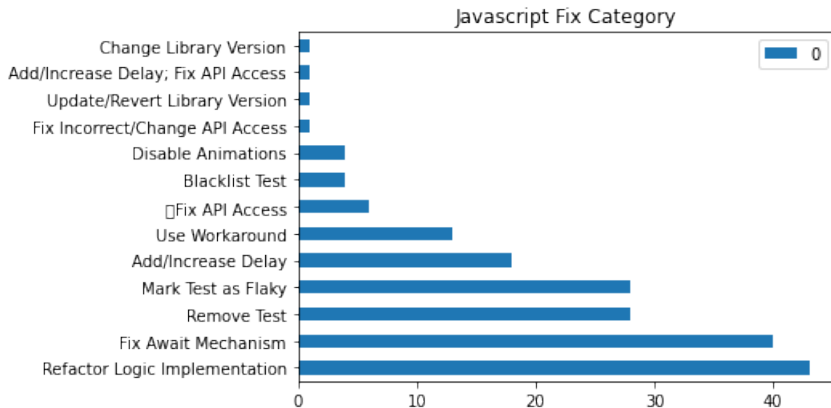


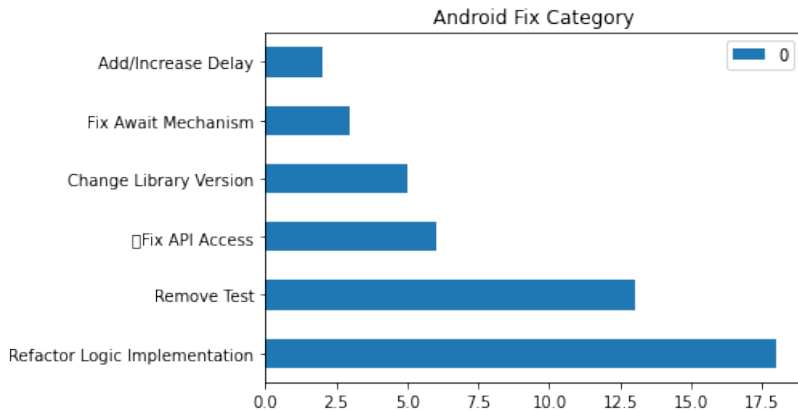






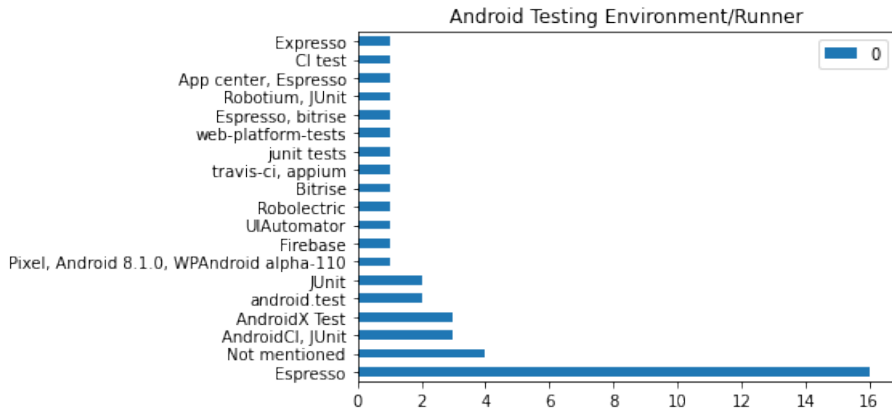






Os ambientes mais comuns do javascript são:

- 1 CircleCi;
- 2 WEb;
- 3 Cypress;
- 4 Jest;
- 5 TravisCI.






De acordo com os gráficos é possível visualizar os seguintes padrões no dataset:

- Problemas na plataforma que executa o teste é um motivo comum tanto em javascript quando no android. Porém, em javascript problemas com renderização de recursos são ainda mais comuns.
- Problemas na plataforma que executa o teste é um motivo comum tanto em javascript quando no android. Porém, em javascript problemas com renderização de recursos são ainda mais comuns.
- Durante a escrita de issues, realização de commits e pull requests a manifestação da falha do teste geralmente não é apresentada na maioria das vezes em ambas as linguagens. Porém, duas manifestações mais comuns são: execução em plataformas específicas e forçar condições de teste.
- As três maneiras mais comuns de se corrigir um flaky são:
  - 1 Refatorando o teste;
  - 2 Corrigindo assíncronicidade do teste;
  - 3 Removendo o teste.



 ROMANO, A. et al. An empirical analysis of ui-based flaky tests. In: IEEE. *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. 2021. p. 1585–1597. Disponível em: <[www.google.com.br](http://www.google.com.br)>.