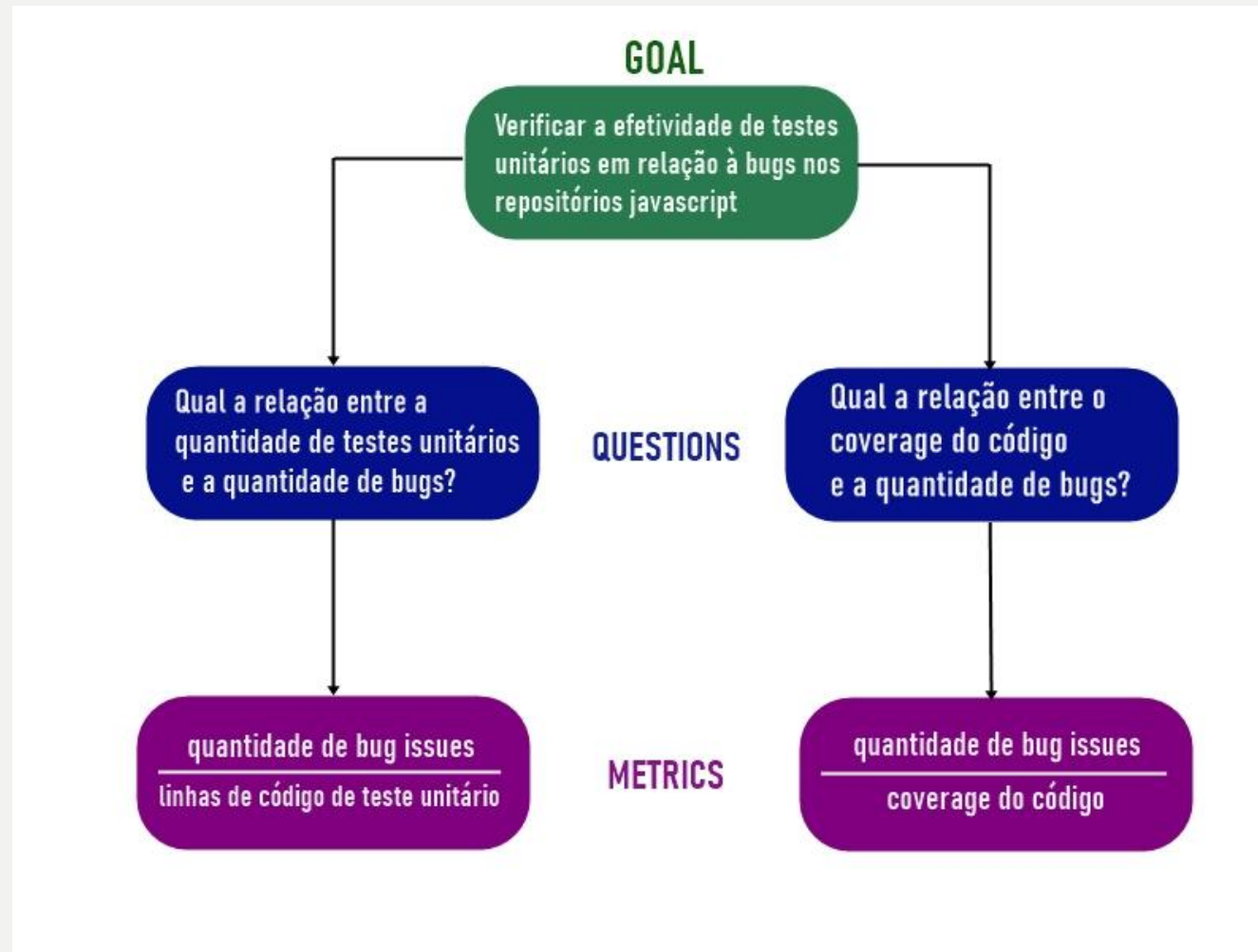




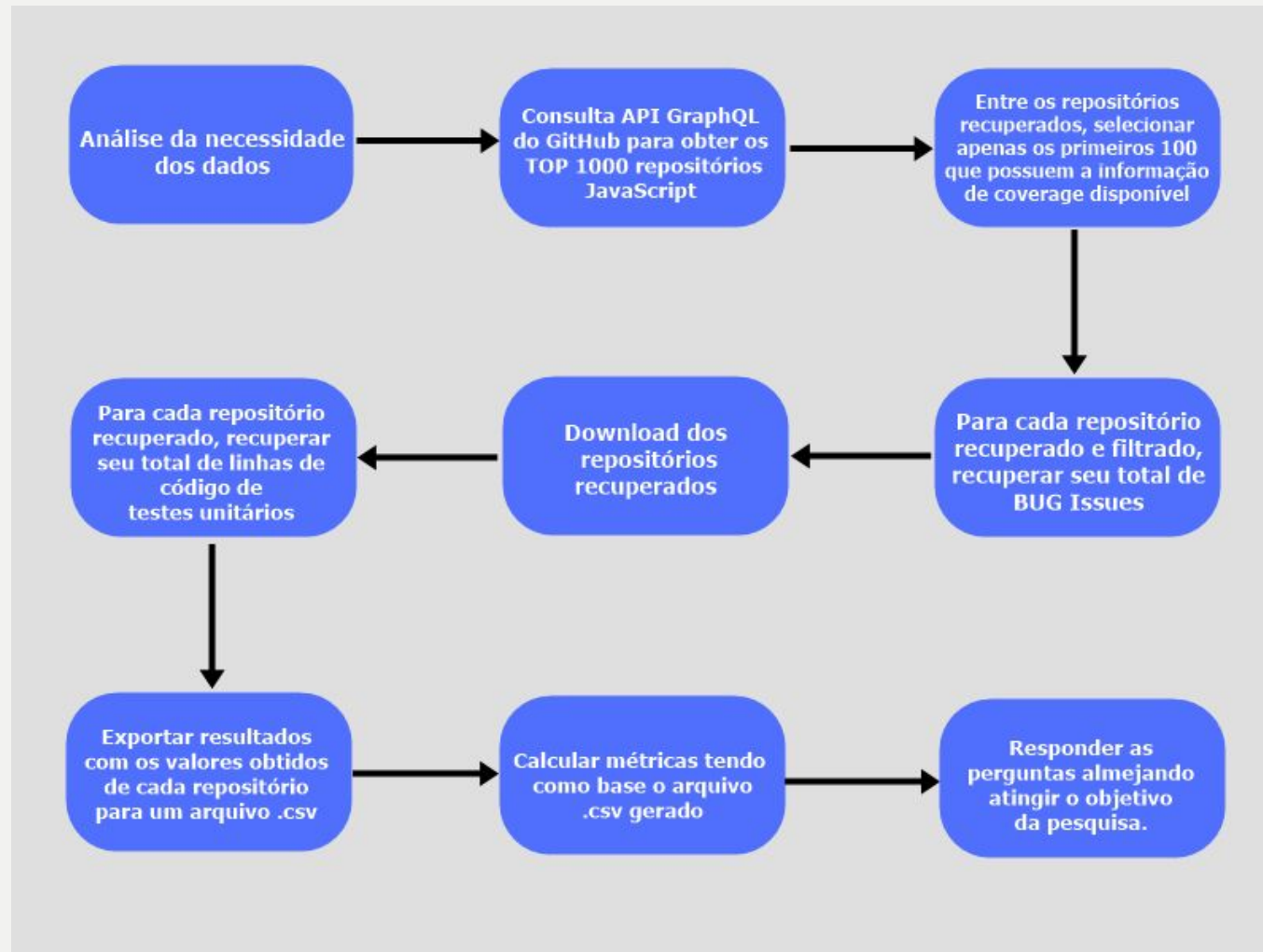
TIS VI

Análise da efetividade de testes unitários nos projetos
JavaScript mais populares do GitHub que possuem informação
de coverage

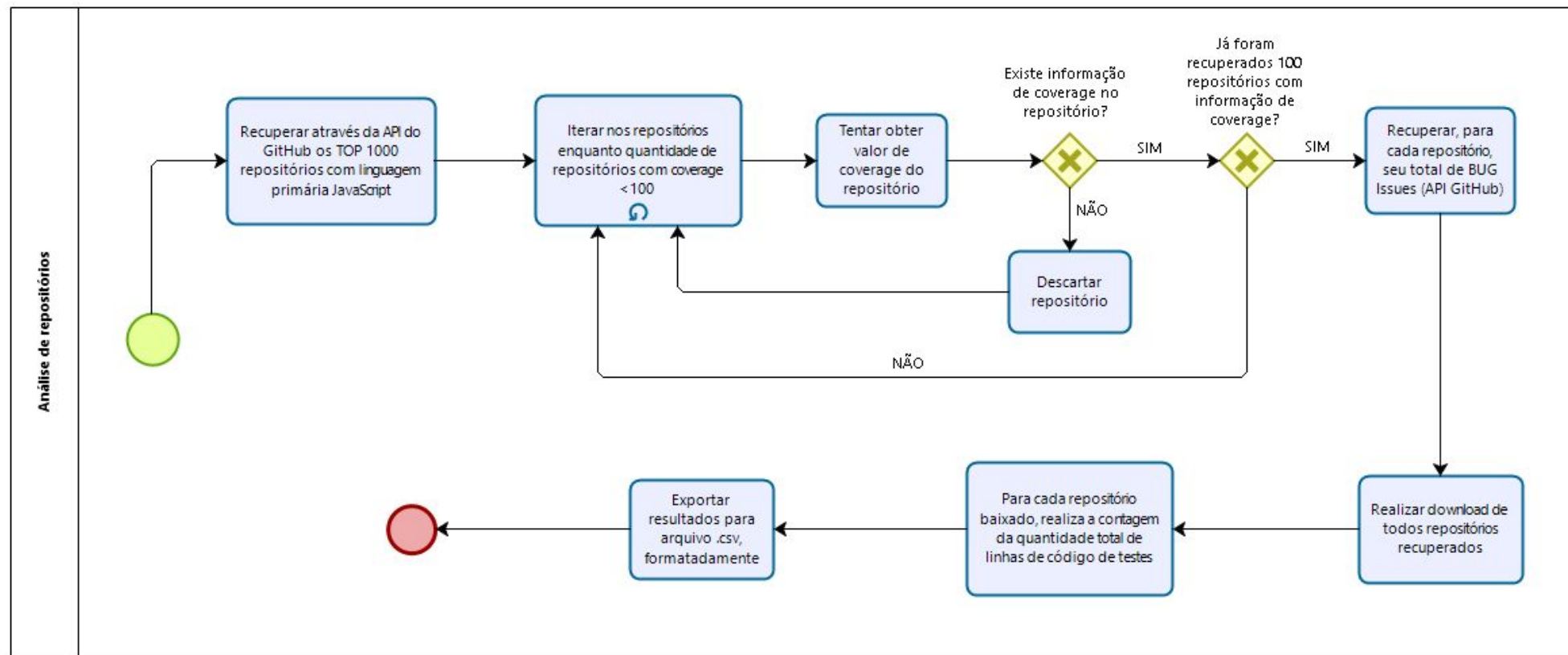
GQM



Overview da metodologia atualizada



Fluxograma do código atualizado



Resultados obtidos

- Com a análise do arquivo .csv gerado, foi possível calcular as métricas (imagem 1 no link abaixo, devido ao tamanho da tabela):
- <https://prntscr.com/sfafns>

Resultados obtidos

- Porém, como foi possível verificar na tabela, não foi possível calcular a métrica 1, uma vez que a quantidade total de LOC de teste de alguns repositórios foi igual a 0 (devido ao fato de seguir um padrão diferente de extensões, fugindo do .spec, .karma, .test e .tests, não sendo, portanto, contabilizado).
- Com a divisão não podendo ser concluída para o cálculo da métrica 1, filtramos todos repositórios cuja quantidade total contabilizada de LOCs de teste fosse igual a 0 e a retiramos da tabela. Posteriormente, refizemos os cálculos:
- <https://prntscr.com/sfagtp>

Análise dos resultados obtidos

- Não encontramos uma relação clara entre a quantidade de testes unitários e a quantidade de bugs e nem entre o coverage do código e a quantidade de bugs.
- Não há um padrão claro entre os valores obtidos, visto que a quantidade de bugs (BUG Issues), que é o valor base para nossos cálculos, é bastante discrepante entre os projetos.
- Valores muito divergentes entre média e mediana.

GitHub do projeto:

- Todos artefatos, documentos e código do projeto estão disponíveis no GitHub:
- <https://github.com/MateusSantosFonseca/TISVI-Analise-Efetividade-Testes-Unitarios>



OBRIGADO!