



**ENACOM**  
DA CIÊNCIA AO PRODUTO

**Sexta of Python - fluxo de testes**  
**18/02/2021**



# Cronograma

- Objetivos;
- Testes do science;
- Testes do QA;
- Fluxos de teste;
  - Feature science;
  - Feature QA;
  - Feature com mudança do JSON.



## Objetivos

- Detalhar sobre o fluxo de testes utilizado para garantir qualidade das entregas;
- Exemplos apresentados focam no que é atualmente utilizado pelo Gêmeo (Gerda).



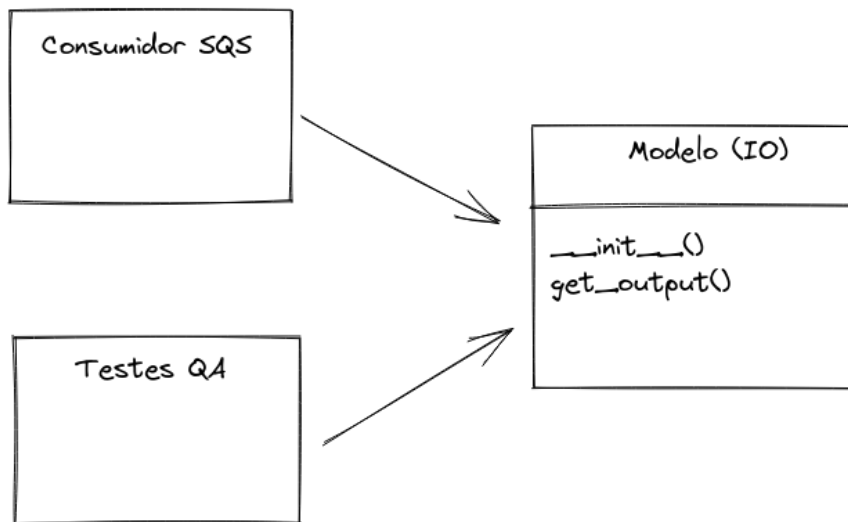
## Testes do Science

- Testes unitários, sempre que possível utilizando TDD;
- Criação e manutenção realizada pelos DEVs;
- Todos são impeditivos para fechar a feature;
- Casos de testes desatualizados, skipar para subir e corrigir posteriormente (não ideal);
- Criar testes para cobrir eventuais bugs.



## Testes do QA

- Testes do caminho completo:
  - *Inputs e outputs* iguais os recebidos pelo SQS





## Testes do QA

- Criação e manutenção por parte da equipe de QA;
- Dado a entrada e a saída de cada CT, são realizadas algumas verificações, que podem ser de dois tipos:
  - Testes de factibilidade:
    - Quando o comportamento esperado é bem definido;
    - Em geral são impeditivos;
  - Testes de performance:
    - Em geral possuem um valor de referência obtido a partir de um ponto estável do código;
    - Necessita definição de uma métrica;
    - Pode ser uma verificação numérica ou gráfica.



## Fluxo feature science

- Usado para qualquer alteração do science que não altere os contratos (JSONs de entrada e saída).
  - Nova funcionalidade;
  - Correção de bugs.

[Fluxograma](#)



## Fluxo feature QA

- Usado para:
  - Criação de novos testes;
  - Mudança na estrutura de execução do QA.

[Fluxograma](#)





## Fluxo feature com mudança do JSON

- Usado para funcionalidade que envolva alteração dos contratos (JSONs de entrada e saída).
  - Desenvolvedor e QA fazem as alterações em paralelo.
    - Alteração *input*:
      - Obs: interessante definir novo JSON de forma a não alterar o comportamento atual dos CTs.
    - Alteração *output*.

[Fluxograma](#)



## Fluxo feature com mudança do JSON

- Usado para funcionalidade que envolva alteração dos contratos (JSONs de entrada e saída).
  - Discutir e definir o JSON com o back / front, e documentar no notion;
  - Alteração *input*:
    - Enviar input\_model atualizado para o QA; - dev
    - Alterar JSON de todos os CTs de acordo; - qa
    - Executar pytest QA (feature);
    - Subir as duas features juntas;
    - Obs: interessante definir novo JSON de forma a não alterar o comportamento atual dos CTs;
  - Alteração *output*:
    - Enviar output atualizado para o QA; - dev
    - Alterar verificações dos testes de acordo (quando necessário);
    - Executar pytest QA (feature);
  - Subir as duas features juntas;