

Atividade Módulo 31 EBAC - Métricas de Qualidade.

Estudante: Luiz Guilherme Menon.

Análise dos Relatórios disponibilizados pelo Cypress Dashboards

O Cypress Dashboards é uma ótima ferramenta intuitiva para extrair métricas relacionadas a saúde de um sistema através de diversos tipos de relatórios, os quais são gerados a partir da execução de projetos compostos por um ou mais branches, sendo possível executá-los via integração contínua.

A configuração para gerar tais relatórios é simples, sendo que existe uma documentação bem explicativa, além de uma ampla comunidade sobre assuntos relacionados as funções do Cypress no geral.

Existem relatórios que trazem a quantidade de execuções por status de sucesso ou de falha (Run status), a duração das execuções (Run duration), a quantidade de testes executados ao longo do tempo (Test suite size), as falhas que mais se repetem (Top failures), os testes mais lentos (Slowest tests) e os erros mais comuns (Most common errors).

Em contas pagas, é possível configurar para que um mesmo teste se repita de modo a tentar identificar os que passam e falham sem mudanças de código (Flaky test). Somado a isso, é possível armazenar imagens e vídeos dos testes para coletar evidências das execuções.

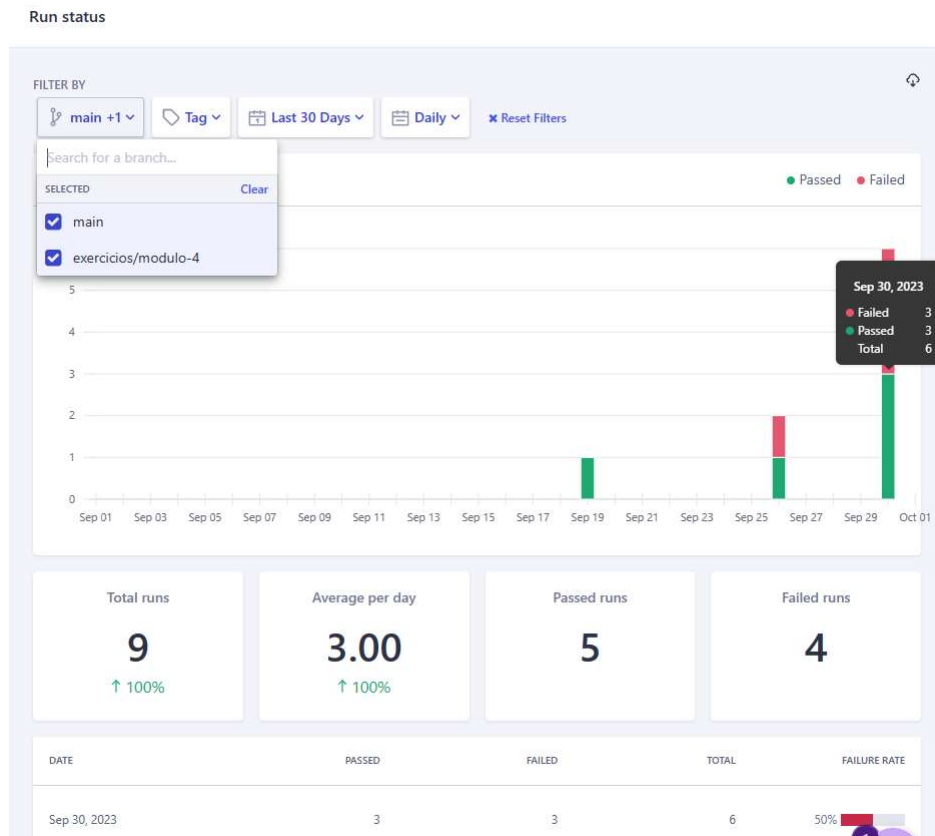
Cada um dos relatórios citados apresentam diversas métricas tais como média de execuções diárias, média de tempo das execuções, quantidades de casos de testes e de arquivos Spec, entre outras. É possível salvar os relatórios como CSV se necessário.

Além disso, existem diversas possibilidades de filtros como branch, tag, autor, versão do Cypress, sistema operacional e período. Também é possível ter uma visão geral dos últimos testes através da opção Latest runs.

Nas próximas páginas, serão apresentados alguns dos relatórios propostos para esta atividade, sendo que os mesmos foram divididos em dois projetos, um para o Módulo 21 (branch main) e outro para dois branches referentes aos Módulos 22 (branch main) e 23 (branch exercicios/modulo-4):

Data: 30/09/2023

Run status com filtros aplicados:



Neste gráfico, são apresentadas a quantidade de execuções diárias nos últimos 30 dias, sendo possível observar que foram executados dois branches, com uma taxa de falha de 50%, ou seja, metade dos testes falharam na última execução, conforme visto a seguir:

Tests for review

All tests for review (2)Failed (1)Flaky (Team)Modified (1)

Checkout.spec.js

✖ 1

✔ 0

🔄 0

•

778ms

•

Windows 10.0.22621

•

Electron 100

•

E2E

✖ Deve realizar Checkout com sucesso > ...a tela de Pedido Recebido

🔄 Test Replay >

Register.cy.js

✖ 0

✔ 1

🔄 0

•

00:21

•

Windows 10.0.22621

•

Electron 100

•

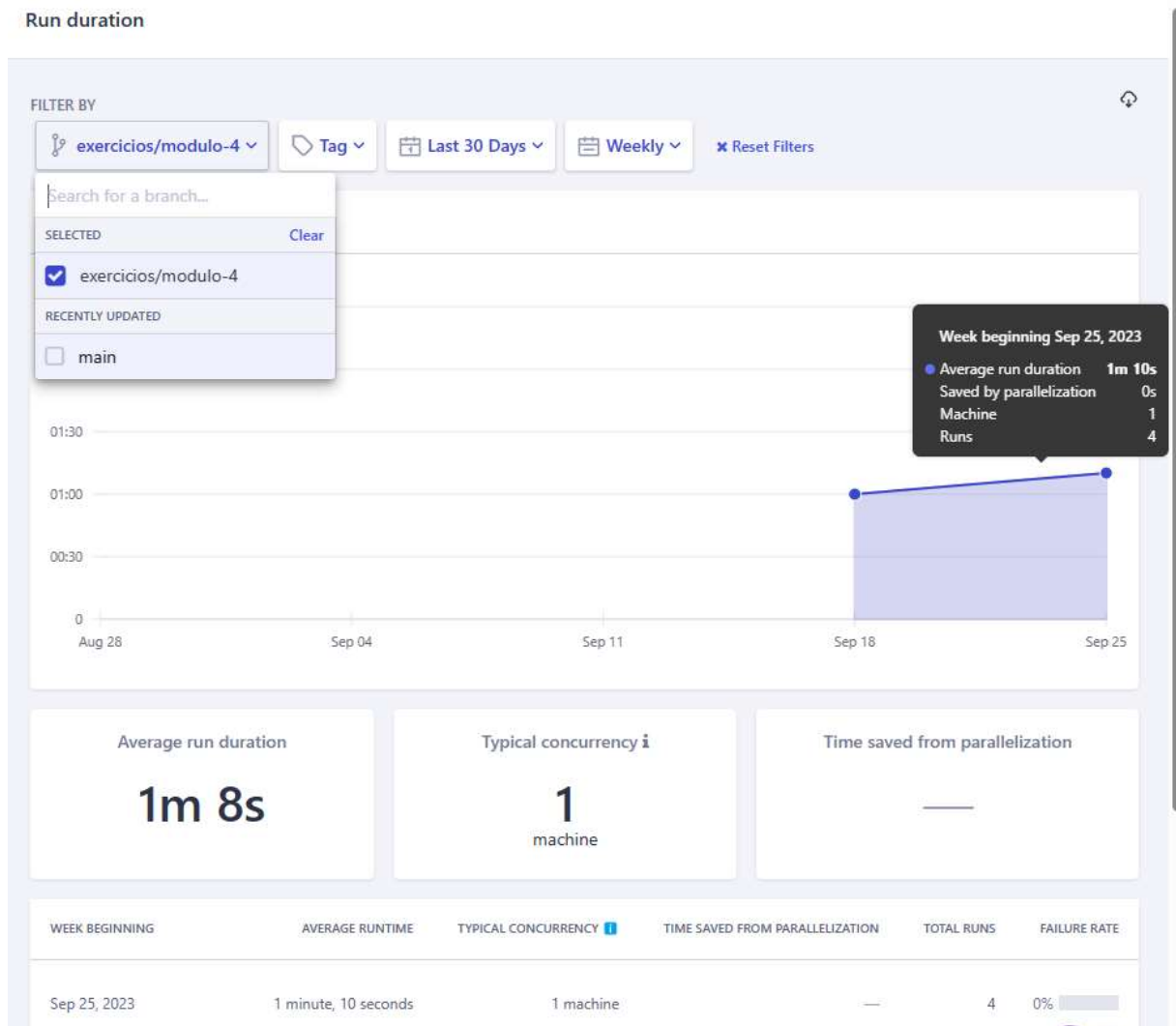
E2E

✔ Register Account > ... > ... > ...welcome message in the logged area FIRST SEEN

🔄 Test Replay >

Data: 30/09/2023

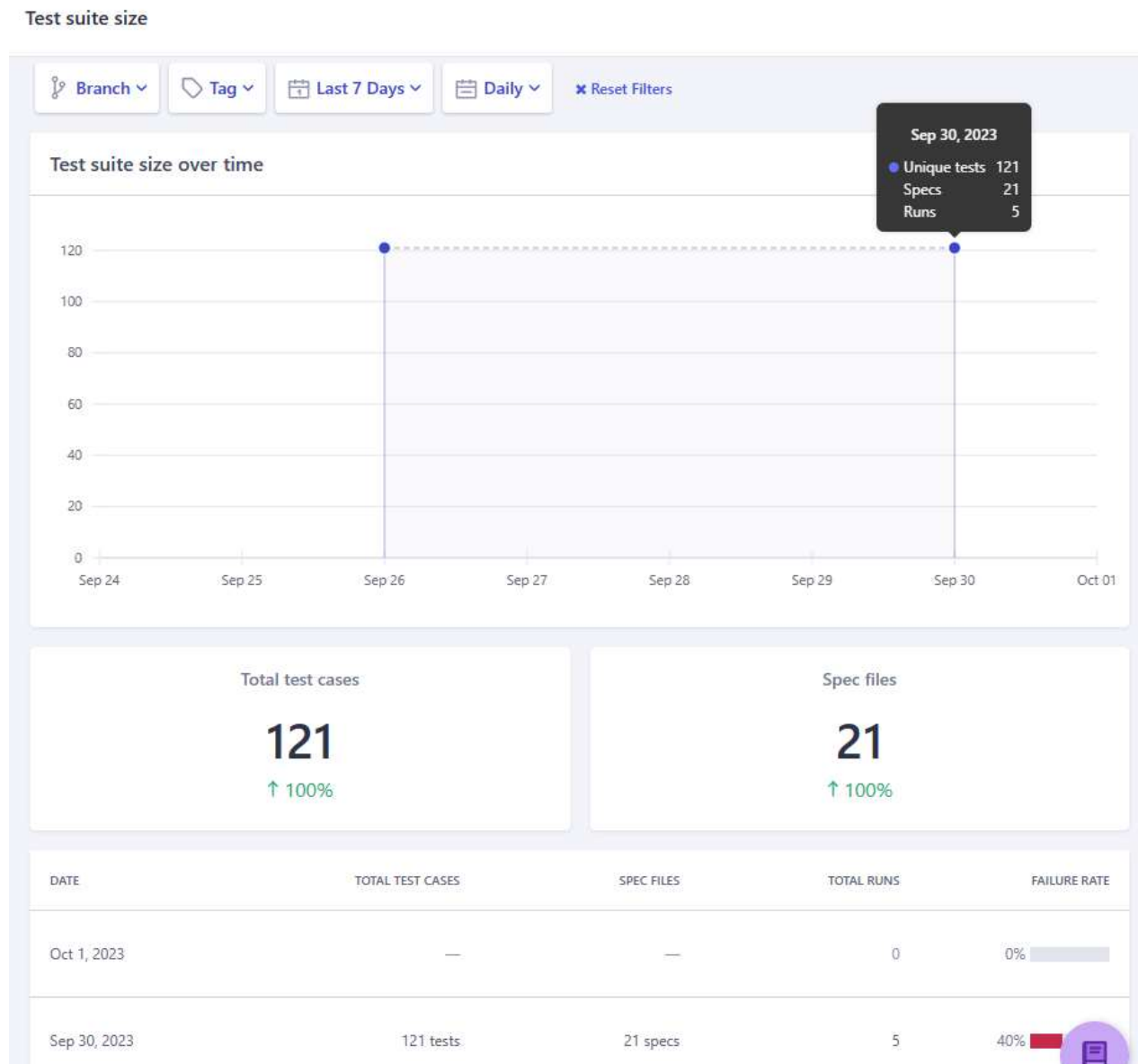
Run duration com filtros aplicados:



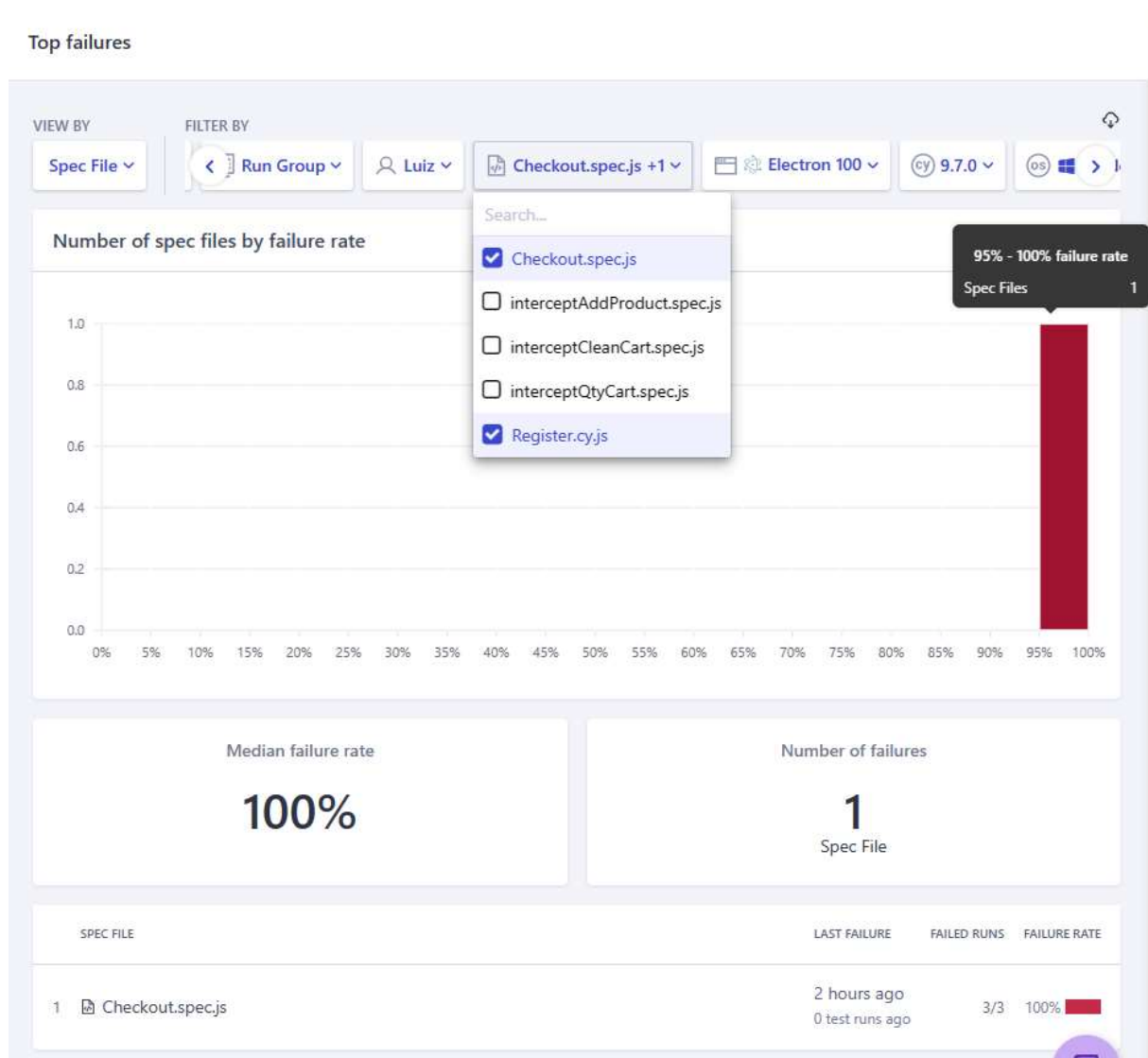
Neste gráfico é possível ver a duração média das execuções semanalmente nos últimos 30 dias, no qual foi filtrado por apenas um branch. Em contas pagas é possível melhorar o desempenho ao utilizar o recurso de paralelismo para execução em mais de uma máquina. Desta forma, é possível saber como está a estabilidade do servidor e dos serviços utilizados no decorrer do tempo.

Data: 30/09/2023

Test suite size com filtros aplicados:

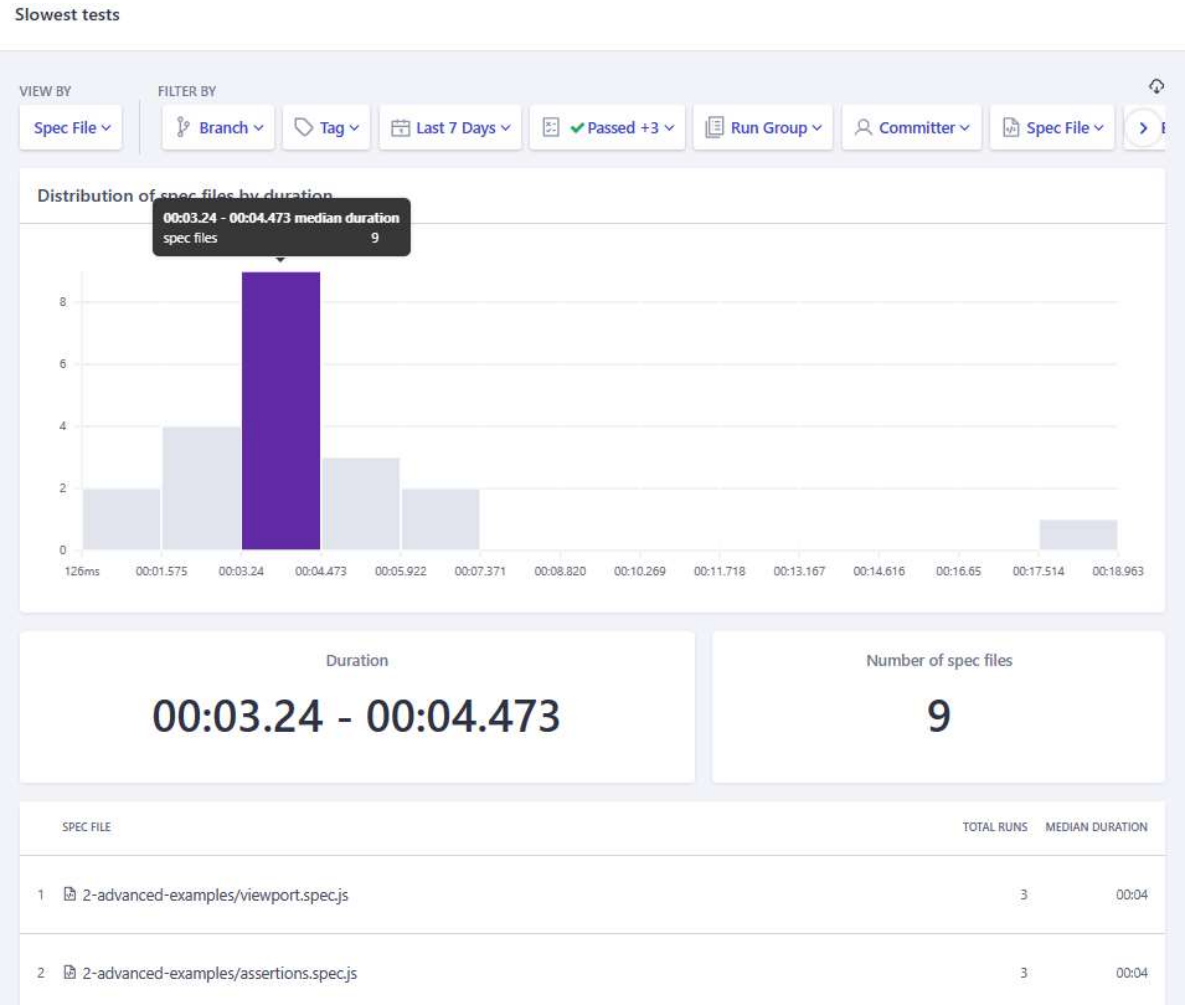


Neste gráfico é possível acompanhar a evolução das quantidades de casos de testes e de arquivos Spec do projeto ao longo do tempo, além de ser possível visualizar a quantidade de execuções.

Top failures com filtros aplicados:

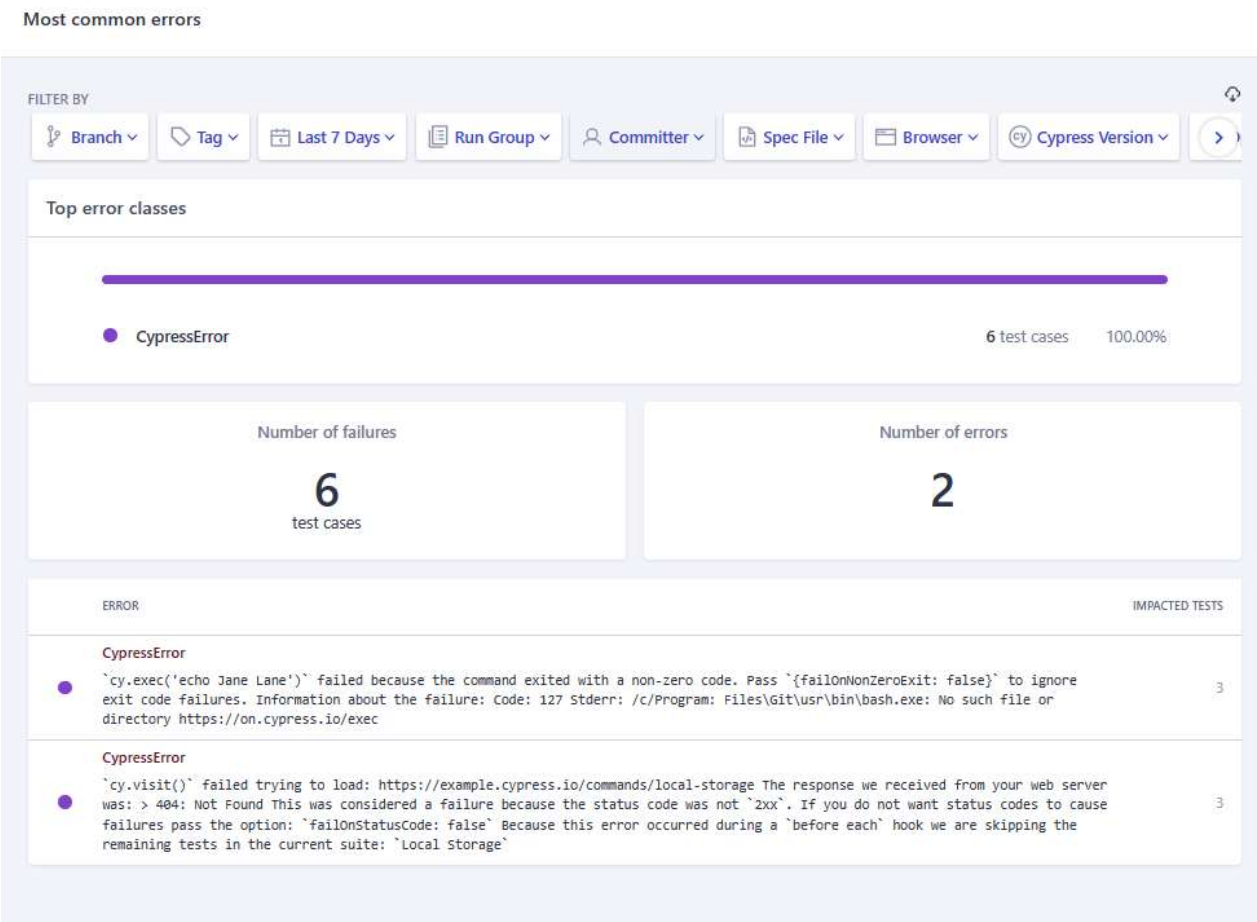
Neste gráfico são apresentadas as falhas que mais se repetem no projeto, sendo que foi filtrado por dois arquivos Spec, retornando que um deles apresentou falha nas três vezes que foi executado.

Slowest tests com filtros aplicados:



Neste gráfico é possível ver quais foram os testes mais lentos do projeto. Neste caso, foi filtrado por arquivo Spec. Ao clicar em uma barra é possível ver mais detalhes da execução na lista inferior.

Most common errors:



Neste gráfico são apresentados os tipos de erros mais comuns que acontecem nas execuções do projeto, indicando quais foram os testes impactados.

Referências:

- [Analytics & Insights | Cypress Documentation](#)
- [Flaky Tests \(cypress.io\)](#)
- Repositório Módulo 21 (branch: main): https://github.com/Lewiz-QA/atividade_31_ebac.git
- Repositório Módulos 22 (branch: main) e 23 (branch: exercicios/modulo-4): https://github.com/Lewiz-QA/atividade_22_ebac.git
- Relatório Módulo 21 (branch main): [sample-component-test-app \(cypress.io\)](#)
- Relatório Módulo 22 (branch main) e 23 (branch exercicios/modulo-4): [atividade_22_ebac \(cypress.io\)](#)