

## CodeReview - Mário

Link do repositório do código avaliado: [PB-Compass/test\\_api at main · Andressa-Mota/PB-Compass](#)

### Pontos Positivos

1. **Excelente Estrutura:** A separação entre `Test Cases` e `Keywords` está perfeita. Isso torna os testes muito mais fáceis de ler e a lógica de automação, mais fácil de manter.
2. **Nomes de Cenários Claros:** Os nomes dos cenários (ex: `Cadastrar novo usuário com SUCESSO`) são descritivos e deixam claro o objetivo de cada teste.
3. **Uso de Tags:** A utilização de tags ( `[Tags]` ) é uma ótima prática para organizar e filtrar a execução dos testes (ex: rodar apenas os testes de `POST` ).

### Pontos de Melhoria

#### 1. Repetição na Criação de Sessão

- **Observação:** A keyword `Criar sessao na api` é chamada dentro de quase todas as outras keywords. Isso é repetitivo e ineficiente, pois uma nova sessão é criada para quase toda requisição.
- **Sugestão:** Crie a sessão **uma única vez** no início da suíte de testes usando `Suite Setup`. Isso torna a execução mais rápida e o código mais limpo.

#### 2. Independência e Acoplamento dos Testes

- **Observação:** O uso de `Set Test Variable` para passar valores entre keywords (como `${id_user}` e `${resp_api}`) cria um "estado global". Isso faz com que a ordem de execução dos passos importe e os testes se tornem dependentes uns dos outros.
- **Sugestão:** Em vez de `Set Test Variable`, faça com que as keywords retornem os valores que elas produzem usando `[Return]`. O caso de teste se torna responsável por passar esse valor para a próxima keyword. Isso torna o fluxo de dados explícito e os testes mais independentes.

### 3. Dados Fixos no Código (Hardcoding)

- **Observação:** Dados como e-mails, senhas e nomes ( `eve.holt@reqres.in` , `Andressa Mota` ) estão fixos dentro das keywords. Se você precisar testar com um usuário diferente, terá que alterar a lógica da keyword.
- **Sugestão:** Centralize esses dados na seção `*** Variables ***` no arquivo `.resource` ou passe-os como argumentos para as keywords, tornando-as mais genéricas e reutilizáveis.

### 4. Clareza das Keywords

- **Observação:** A keyword `Fazer login` chama outra keyword ( `Login com usuario` ) e depois faz uma asserção. Isso cria uma camada extra que pode ser simplificada. A keyword de validação ( `Conferência de cadastro` ) depende de uma variável que foi criada em outra keyword.
- **Sugestão:** A asserção (verificação) pode ser feita diretamente no caso de teste ou em uma keyword de validação que recebe o resultado como argumento. Keywords devem ter uma responsabilidade única e clara.