Code Review

Product_test

- **Revisor:** Squad 5 - Raique Alfredo
- **Data:** 18/06/2025
- **Autor do código:** Squad 6 - Cavalheiros da Qualidade
- **API:** ServeRest
- **Link do repositório:** GitHub - Rodrigo-Matuz/robot_framework_serverest
- **Branch:** main
- **Test Cases analisados:**
  - Cadastrar produto no sistema
  - Pesquisar produto e verificar resultados

Pontos Positivos

Organização:

- A estrutura dos arquivos está muito bem organizada. Separar os **test cases** em `tests` e as **keywords** em `resources` reflete um entendimento sólido das melhores práticas para modularidade e manutenção de código.

Boas Práticas:

- A utilização de um arquivo central de variáveis (`env_variables.resource`) foi uma escolha excelente. Isso facilita a reutilização e a alteração de dados sensíveis ou mutáveis.
- A abordagem de criar uma `common_keywords.resource` para funções compartilhadas, como abrir o navegador, demonstra um bom uso de abstração e reaproveitamento de código.

Funcionalidade e Fluxo:

- O código cobre cenários importantes para a aplicação ServeRest, como o cadastro e a busca de produtos, utilizando dados dinâmicos das variáveis.
- As keywords são bem nomeadas, claras e seguem um padrão consistente, o que ajuda na leitura e entendimento do propósito de cada uma.

Pontos de Melhoria

Cobertura de Testes:

1. **Validações Explícitas:** Na **keyword "Criar Produto com sucesso como administrador"**, seria interessante validar a resposta ou o estado final da aplicação (por exemplo, verificar a presença do produto na lista após o cadastro).

Padrões e Consistência:

1. **Timeouts:** Os timeouts utilizados nas ações (`Wait For Elements State`) poderiam ser extraídos para uma variável global, garantindo consistência e facilidade de ajuste futuro.
2. **Mensagens de Erro:** Adicionar assertivas ou verificações que retornem mensagens personalizadas quando algo não ocorre como esperado ajudaria a identificar problemas rapidamente.

Pontos de Melhoria Relacionados aos Erros

Timeout nos Locators:

1. **Descrição do Problema:**
   Nos testes que utilizam a keyword `Pesquisar produto`, foi reportado um `TimeoutError` ao tentar localizar o elemento `[data-testid="pesquisar"]`. Isso indica que o elemento não ficou visível no tempo esperado.
2. **Possíveis Causas:**
   - O seletor CSS `[data-testid="pesquisar"]` pode estar incorreto ou não corresponde a um elemento carregado na página atual.
   - O tempo de espera configurado (`timeout=3s`) pode ser insuficiente, especialmente em ambientes onde o carregamento da página pode ser mais lento.
   - O elemento pode não estar sendo renderizado devido a erros na navegação para a URL correta ou a problemas com a aplicação ServeRest.
3. **Recomendações:**
   - **Verifique o seletor CSS:** Confirme se `[data-testid="pesquisar"]` é o identificador correto para o campo de busca na aplicação ServeRest.
   - **Aumente o tempo de espera:** Ajuste o `timeout` para um valor maior, como `timeout=10s`, para garantir que o elemento tenha tempo suficiente para ser carregado.
   - **Adicione validação de navegação:** Inclua uma validação para garantir que a página correta foi carregada antes de executar os testes, utilizando um elemento único da página como referência.
   - **Teste manualmente a aplicação:** Verifique se o campo de busca está presente e funcional na aplicação ServeRest, descartando problemas de infraestrutura ou configuração local.

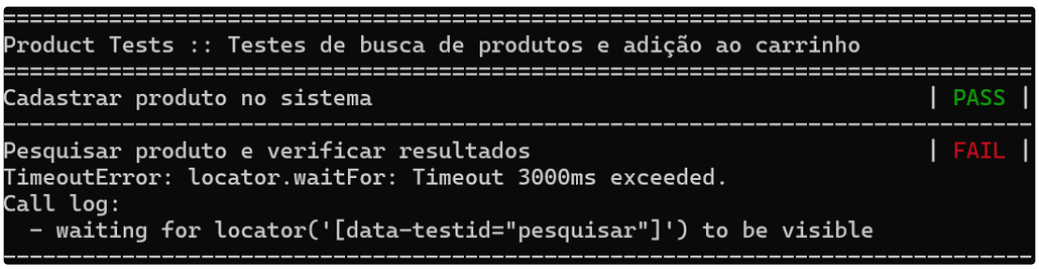
Sugestões de Melhorias Adicionais

1. **Refatoração das Keywords:**
   - **Pesquisar Produto:** Adicionar validações ao final da keyword para garantir que os resultados retornados são condizentes com o termo de busca.

Conclusão:

- O código está muito bem estruturado e segue boas práticas de organização e reutilização. A separação de responsabilidades entre **test cases**, **keywords** e **variáveis** está clara e consistente, refletindo um entendimento sólido de desenvolvimento com Robot Framework.
- Os problemas identificados parecem ser majoritariamente relacionados à visibilidade dos elementos e ao tempo de espera. Com as correções sugeridas, como a validação dos seletores, ajustes nos timeouts e a inclusão de mensagens personalizadas nos logs, é provável que os testes sejam executados com mais consistência e eficiência.
- As sugestões apresentadas têm como objetivo melhorar a clareza, a manutenção, a cobertura do projeto e a rastreabilidade dos testes. Parabéns pelo excelente trabalho e dedicação ao projeto!

Evidência da execução do teste:



Observação: Utilizei ChatGPT para consulta nesse code review.