

Considerações - Atividade de Automação:

1. **Framework escolhido:** Cypress;
2. **Linguagem escolhida:** Javascript;
3. **Link para o repositório:** <https://github.com/matheusAzevedo/automation-activity-2>.

4. Instruções para execução:

4.1. Pré-requisitos:

- 4.1.1. Ter o Node instalado na máquina;
- 4.1.2. Ter alguma IDE (recomendo o VS Code);
- 4.1.3. Outras informações sobre o Cypress podem ser vistas nesse link [aqui](#) ou na [documentação oficial](#).

4.2. Como configuração inicial do projeto, rode os seguintes comandos:

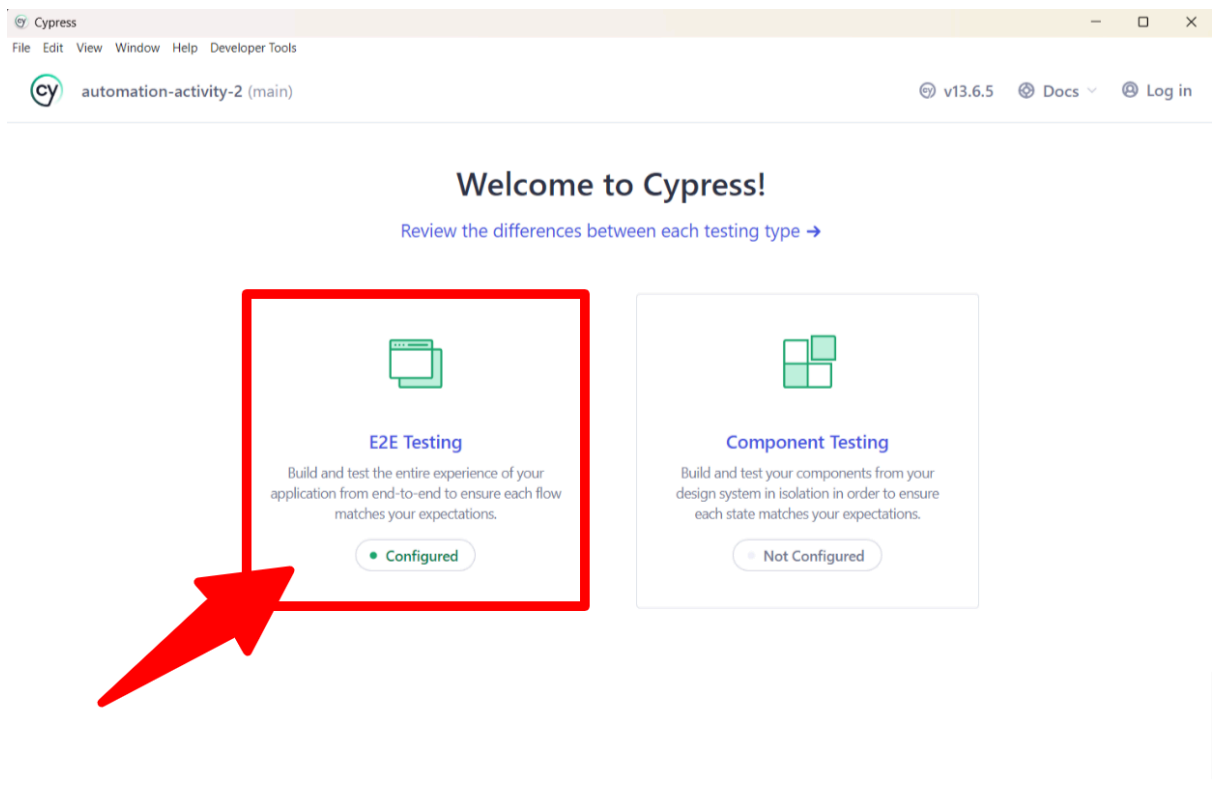
```
$ npm init -y  
$ npm install cypress
```

Eles se encarregarão de montar a estrutura padrão de um projeto cypress, bem como criar o package.json.

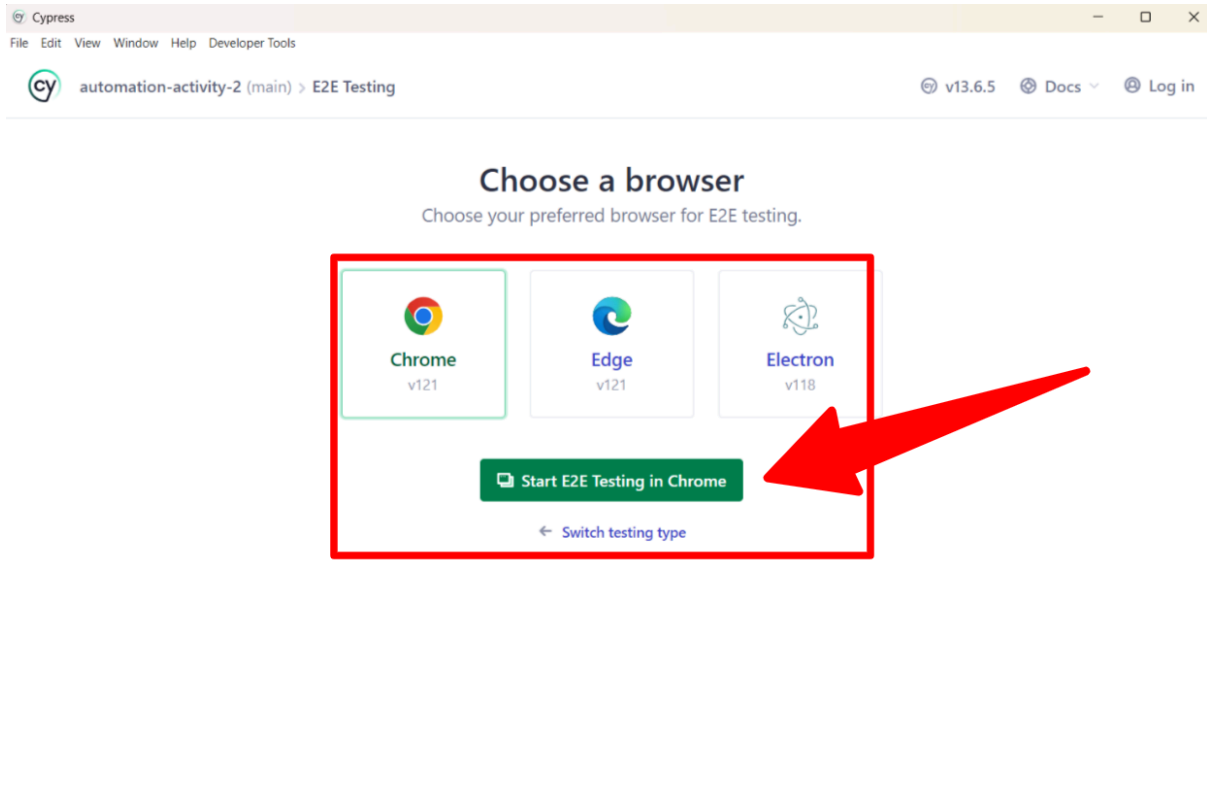
4.3. Após a configuração adequada do projeto, execute o seguinte comando para executar o Cypress:

```
$ npx cypress open
```

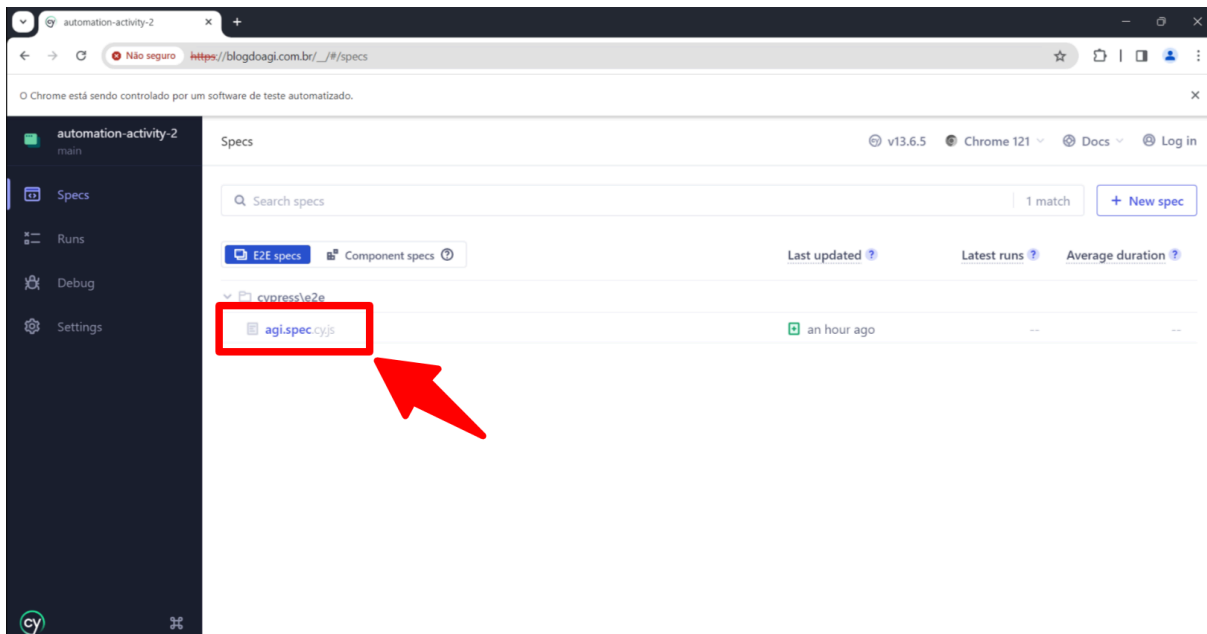
4.4. Ao se deparar com essa tela, selecione a opção indicada em vermelho:



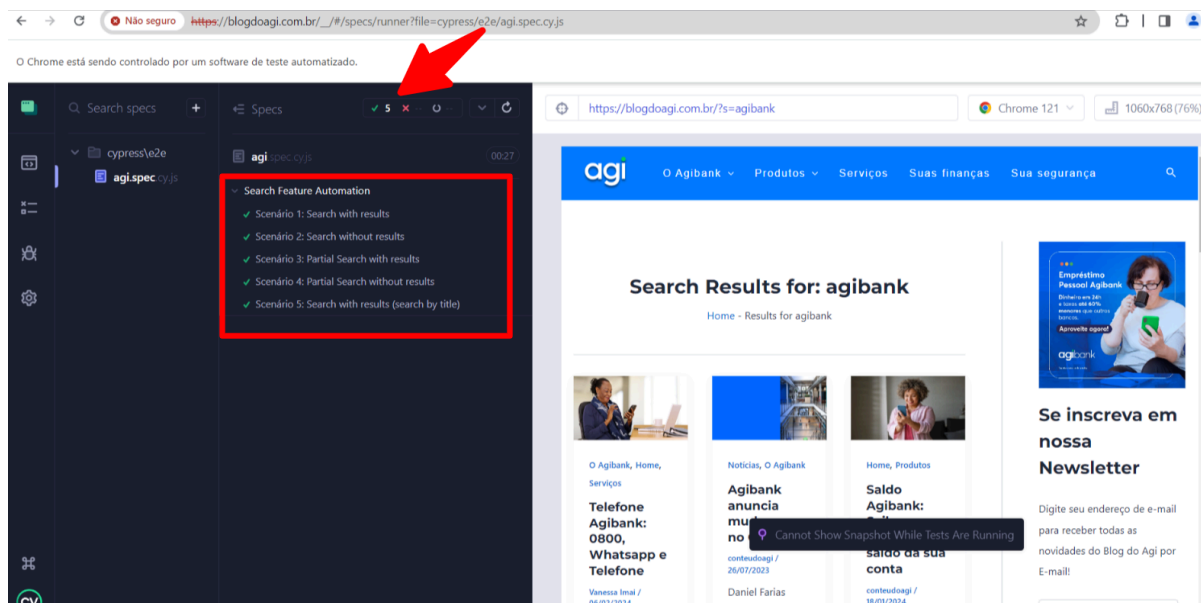
4.5. Selecione o browser (o browser utilizado nos testes foi o chrome):



4.6. Selecione o arquivo que contém as especificações dos testes automatizados que foram pedidos na atividade:



4.7. Pronto! Os testes serão executados um a um, exibindo um report semelhante ao mostrado abaixo:



5. Considerações sobre a implementação dos testes:

- 5.1. Além dos cenários que foram pedidos, mais três cenários (3, 4 e 5) também foram implementados e constam na classe **agi.spec.cy.js**;
- 5.2. Não ficou claro se a busca deveria contemplar nomes que estivessem contidos apenas no **título** do artigo ou se poderiam estar presentes por todo o texto também. As duas possibilidades foram automatizadas, respectivamente cobertas pelos cenários 5 e 1;
- 5.3. Não ficou claro também se a busca parcial deveria funcionar. Fiz alguns testes, procurei por algumas palavras e a busca parcial é realizada com sucesso para algumas palavras e suas derivações (ex: busquei por “**árvore**” e o blog me retornou artigos com a palavra “**árvores**”), especificado no cenário 3; mas quando busco por “**tecno**”, artigos com a palavra “**tecnologia**”, por exemplo, não são retornadas, provavelmente porque a palavra “**tecno**” não existe, especificado no cenário 4.
- 5.4. Cláusulas gherkin foram adicionadas como comentários pelo código, para facilitar o entendimento dos fluxos.