**Definição de Testes:**

Testes utilizados em **Front-end:**

1. **Framework de Teste:**

É um framework de teste de front-end voltado especificamente para testar aplicações web modernas. Oferece uma API intuitiva que permite aos desenvolvedores escrever testes de forma declarativa e eficiente.

1. **JavaScript:**

O próprio Cypress é escrito em JavaScript e isso torna fácil para o uso de desenvolvedores familiarizados com a linguagem, então os testes podem ser escritos na mesma linguagem e executados diretamente no navegador, tornando tudo muito mais ágil.

1. **Testes End-to-End:**

Onde você simula a interação de usuário com a aplicação, desde a abertura do navegador até a realização de ações como clicar em botões, preencher formulários e verificar resultado.

1. **Acesso ao DOM:**

Uma das maiores vantagens do Cypress é seu acesso direto ao DOM (Document Object Model), que permite aos desenvolvedores interagir diretamente com os elementos da página e realizar verificações precisas sem necessidade de esperas prolongadas.

1. **Tempo Real:**

Oferece uma visualização em tempo real dos testes enquanto são executados, o que facilita a depuração e a identificação dos problemas.

1. **Facilidade de Configuração:**

É fácil de configurar e já começar a ser usado em projetos. Ele é distribuído como um pacote NPM e pode ser integrado facilmente a pipelines de integração contínua para automação de testes.

1. **Documentação e Comunidade Ativa:**

Cypress possui uma extensa documentação e uma comunidade ativa de usuários, o que facilita a aprendizagem e o suporte para problemas específicos.

Testes utilizados em **Back-end:**

O Cypress originalmente faz testes focado em usar o front-end e não comumente usados para testes em back-end. Em vez disso, para testar o back-end de uma aplicação, é mais como utilizar ferramentas especificas para isso, como Jest, Mocha, Chai, Postman, entre outras.

Mas é possível usar o Cypress para simular requisições HTTP e verificar se as respostas estão corretas. Isso pode ser útil para testar a integração entre front-end e back-end de uma aplicação.

**Jest:** É um framework de teste de JavaScript mantido pelo Facebook, frequentemente utilizado para testar os códigos nessa linguagem, incluindo back e front-end.

**Chai:** Biblioteca de asserções para JS e Node.js. Ele permite que você escreva testes de forma mais expressiva e legível, fornecendo uma ampla gama de estilos de asserção para verificar se os resultados dos testes são os esperados.

**Mocha:** Também é um framework de teste de JS flexível e extensível. Fornece uma estrutura simples para escrever e organizar testes de forma clara e concisa ao mesmo tempo.

**Postman:** Postman é uma ferramenta popular para testar as API´s. Ele fornece uma interface amigável para enviar solicitações HTTP para sua API e verificar as respostas. Essa ferramenta permite criar e organizar coleções de testes, automatizar fluxos de trabalho de teste e compartilhar facilmente esses testes e documentação da API com sua equipe.

Em resumo, testes Cypress oferecem uma maneira poderosa e eficiente de garantir a qualidade de aplicação web front-end, especialmente em termos de testes E2E e integração contínua. Diferentemente do backend que existem essas ferramentas resposaveis pela realização de tais testes.