

## 03 Instalando o Cypress



### Transcrição

Iniciamos nossa jornada na escrita de testes E2E, e ficamos responsáveis por testar a página da **Adopet**.

Na **tela de boas-vindas** da Adopet, temos um botão para visualizar pets disponíveis para adoção, além de botões para cadastro e login.

Clicando em "Cadastrar", nos redirecionamos para a **página de cadastro**, onde temos um formulário com os campos de nome, e-mail, senha e confirmação de senha e um botão para efetuar o cadastro.

Podemos testar esses campos preenchendo-os um a um. No entanto, realizar esses testes manualmente seria muito trabalhoso!

Por isso, nossa equipe decidiu trabalhar com uma ferramenta chamada **Cypress** para **automação dos testes**. Com ela, vamos simular uma pessoa usuária interagindo com as funcionalidades da página.

## Explorando a documentação do Cypress

Vamos conhecer a ferramenta a partir de sua documentação, acessando-a no site `cypress.io`, onde temos uma visão geral de como o Cypress funciona e como podemos instalá-lo.

Clicamos no botão "***Documentation***" no centro da tela para acessar a documentação em si. Nessa página, vamos clicar em "***Getting Started***" no menu lateral esquerdo, para conferir os passos iniciais e de instalação dessa ferramenta. Então, clicamos na opção "***Installing Cypress***" (Instalando Cypress).

Conforme o que diz a documentação, precisaremos usar o comando `npm install cypress --save-dev` para instalar a ferramenta. No entanto, para trabalhar com esse comando, precisamos primeiro **inicializar um projeto com o NPM**.

Então, vamos deixar a documentação de lado por um momento, assim como a página do Adopet, para configurar o ambiente local na nossa máquina.

## Inicializando um projeto com NPM

Criamos uma pasta chamada "**curso\_cypress**" no nosso computador. Abrimos essa pasta e o terminal dentro dela, clicando em seu interior com o botão direito do mouse e selecionando a opção "*Git Bash Here*".

Com o Git Bash aberto, verificamos se realmente estamos na pasta "curso-cypress" (a partir dos dizeres iniciais do terminal, indicando o diretório atual) e inicializamos um **repositório git** executando o comando:

```
git init
```

[COPIAR CÓDIGO](#)

Em seguida, vamos executar um comando para **inicializar nosso projeto com o NPM**:

```
npm init
```

[COPIAR CÓDIGO](#)

Esse comando traz algumas informações, como o nome do pacote, versão, descrição, licença, etc. Para confirmá-las, vamos pressionando a tecla **"Enter"**.

Fazemos isso até momento em que o terminal perguntar **"Is *this OK?*"** para confirmar a inicialização do projeto. Pressionamos "Enter" mais uma vez e pronto! Projeto criado.

## Instalando o Cypress no projeto

Vamos limpar o terminal com o comando `clear` para instalar o Cypress. Para isso, utilizaremos o comando que a documentação nos indicou, mas queremos uma versão específica do Cypress, a **13.6.4**. Vamos fazer essa especificação no comando:

```
npm install cypress@13.6.4 --save-dev
```

[COPIAR CÓDIGO](#)

Executando e aguardamos a instalação do Cypress finalizar.

## Abrindo o Cypress

Com o Cypress instalado na nossa máquina, vamos abrir a ferramenta para começar a trabalhar. Para isso, executamos no terminal:

```
npx cypress open
```

[COPIAR CÓDIGO](#)

Isso inicializa a ferramenta no navegador. O Cypress é interessante porque trabalha com os navegadores web!

Na tela de boas-vindas dessa interface gráfica do Cypress, temos duas opções:

- ***E2E Testing*** (criar teste *end-to-end*)
- ***Component testing*** (criar teste de componente).

Selecionaremos a primeira opção, *E2E Testing*, o que nos redireciona para uma nova tela que contém os arquivos de configuração do Cypress.

O primeiro, chamado `cypress.config.js`, é um arquivo de **configurações globais** do Cypress, muito importante para nós. Esse arquivo é escrito em JavaScript, o que nos indica que o Cypress é baseado nessa linguagem.

Em seguida, temos um arquivo que está dentro de uma pasta chamada "support" cujo nome é `e2e.js`. É neste arquivo em que vamos **escrever nossos testes**, de fato.

Em seguida, temos dois outros arquivos que vamos entender melhor mais adiante neste curso, chamados `support\commands.js` e `fixtures\example.json`.

Clicamos no botão "***Continue***" (continuar) para seguir para a próxima tela. Essa nova tela pede para **selecionarmos um navegador** para rodar os testes, conforme os navegadores presentes na nossa máquina.

A instrutora selecionou o Chrome e, depois, clicou no botão "***Start E2E Testing in Chrome***". Ao fazer isso, abre-se uma

nova janela de navegador para iniciar os testes.

O Cypress opera como um navegador para **simular o comportamento de usuários**. Nesse ambiente de testes, temos um menu à esquerda com opções como *Specs*, *Runs*, *Debug* e *Settings*. Vamos explorá-las mais detalhadamente ao longo do curso.

Na área central da tela temos duas opções:

- ***Scaffold example specs***: área com exemplos e modelos de testes, que você pode explorar depois, para guiar você na escrita de testes no Cypress.
- ***Create new spec***: criar novo teste.

Vamos clicar na segunda opção para criar nosso primeiro teste. Faremos isso no próximo vídeo. **Esperamos você lá!**