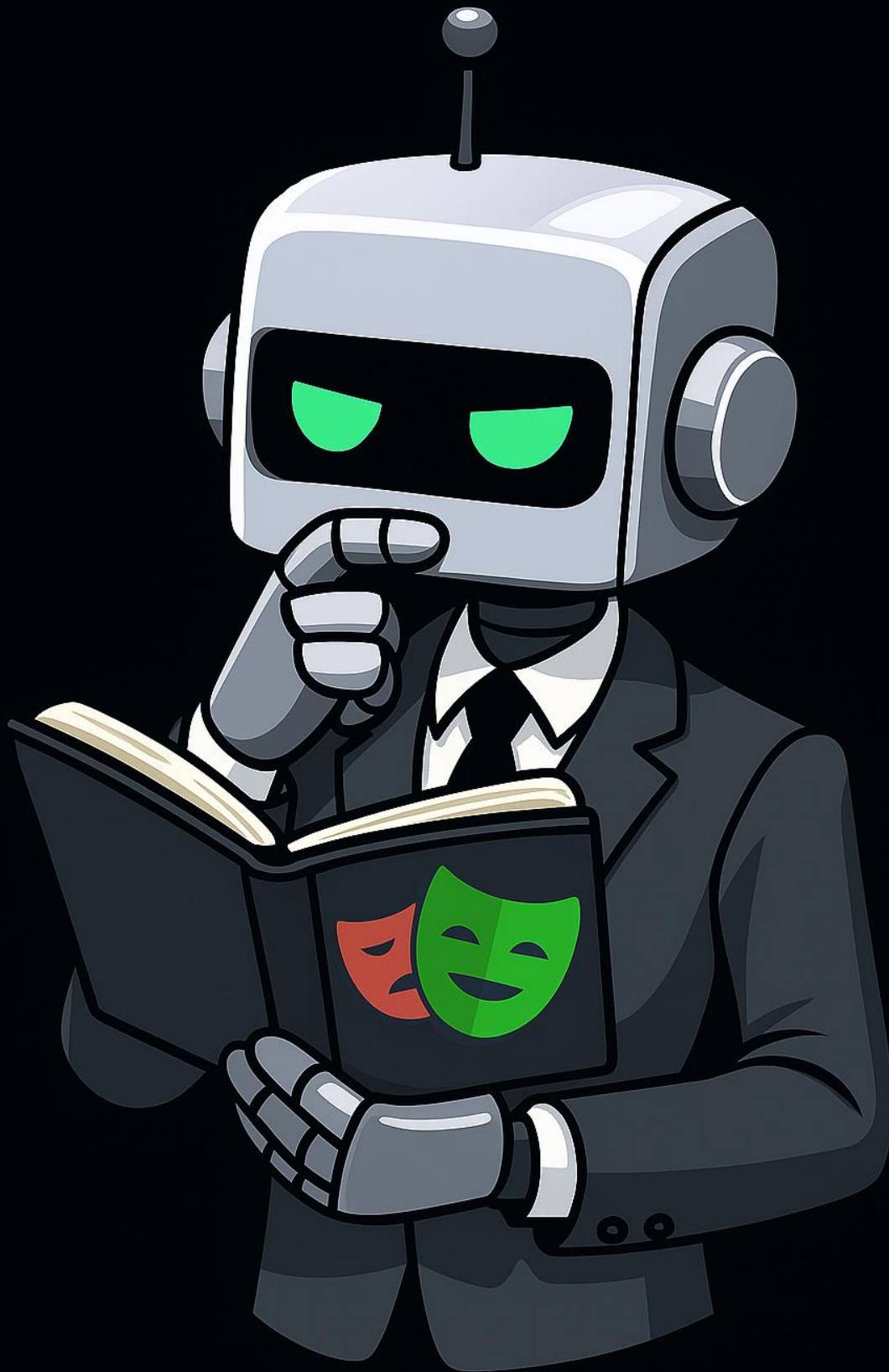


PLAYWRIGHT

Sintetize. Teste. Domine.



RHUAN QUADROS

01

INTRODUÇÃO



O QUE É O PLAYWRIGHT?

- Playwright é uma ferramenta da Microsoft para automatizar testes e interações em sites, simulando ações reais em diferentes navegadores, com alta performance e estabilidade.



- **Suporta:** Chromium, Firefox e WebKit
- **Linguagens:** JavaScript, TypeScript, Python, Java, .NET

02

TESTE SIMPLES



COMO FAZER UM TESTE

- Primeiramente instale o playwright usando:

```
npm init -y
```

```
npm i -D playwright
```

- Crie um arquivo teste na linguagem desejada:

```
const { chromium } = require('playwright');

(async () => {
  const navegador = await chromium.launch();
  const pagina = await navegador.newPage();
  await pagina.goto('https://example.com');
  await pagina.screenshot({ path: 'exemplo.png'
});
  await navegador.close();
})();
```

- Abrimos o navegador, acessamos a URL e tiramos um print da tela

03

INTERAGINDO



INTERAGINDO COM ELEMENTOS

- Automatize interações como preencher campos e clicar em botões:

```
await pagina.fill('#usuario', 'Rhuan');  
await pagina.fill('#senha', '123456');  
await pagina.click('#login');
```

- Playwright espera automaticamente pelos elementos, mas você pode forçar:

```
await pagina.waitForSelector('#mensagemSucesso');
```

- Verifique se algo realmente foi exibido

```
const sucesso = await  
pagina.isVisible('#mensagemSucesso');  
console.log(sucesso ? 'Login OK' : 'Erro no login');
```

04

PREENCHENDO
FORMULÁRIOS
E
FUNÇÕES



PREENCHENDO FORMULÁRIOS

- Um exemplo completo de preenchimento de formulário:

```
await pagina.goto('https://meusite.com/cadastro');
await pagina.fill('#nome', 'Rhuan Quadros');
await pagina.fill('#email', 'rhuan@email.com');
await pagina.check('#termos');
await pagina.click('#enviar');
```

- Tornando o código acima reutilizável:

```
async function fillForm(pagina, nome, email) {
  await pagina.goto('https://meusite.com/cadastro');
  await pagina.fill('#nome', nome);
  await pagina.fill('#email', email);
  await pagina.check('#termos');
  await pagina.click('#enviar');
}
```

05

BUSCANDO
ELEMENTOS
E
DOCUMENTANDO



BUSCANDO E DOCUMENTANDO ELEMENTOS

- Quer pegar vários itens de uma lista?

```
const itens = await pagina.$$('.item-lista');
for (const item of itens) {
  const texto = await item.textContent();
  console.log(texto);
}
```

- Para depurar testes:

```
await pagina.screenshot({ path: 'tela.png' });
// Ou ative vídeos:
const navegador = await chromium.launch({
  args: ['--use-fake-ui-for-media-stream']
});
```

06

RODANDO O
TESTE



RODANDO COM PLAYWRIGHT TEST

- Você pode usar o Playwright Test Runner:

```
npm init playwright@latest
```

- Exemplo de teste com test():

```
const { test, expect } = require('@playwright/test');

test('deve exibir título correto', async ({ page }) => {
  await page.goto('https://example.com');
  await expect(page).toHaveTitle('Example Domain');
});
```

- 🧠 Com o runner, você organiza testes com describe, test e expect.

AGRADECIMENTOS



OBRIGADO POR LER ATÉ AQUI

`console.log`

`  Agradecimento

Muito obrigado por acompanhar este material!

Este conteúdo foi gerado com o auxílio de Inteligência Artificial, com curadoria e diagramação realizada por um ser humano. O objetivo é exclusivamente educacional, para compartilhar conhecimento sobre automação com Playwright de forma acessível e prática.

Sinta-se à vontade para reutilizar, adaptar e aprimorar este conteúdo em seus estudos e projetos.

— Rhuan Quadros   `

]