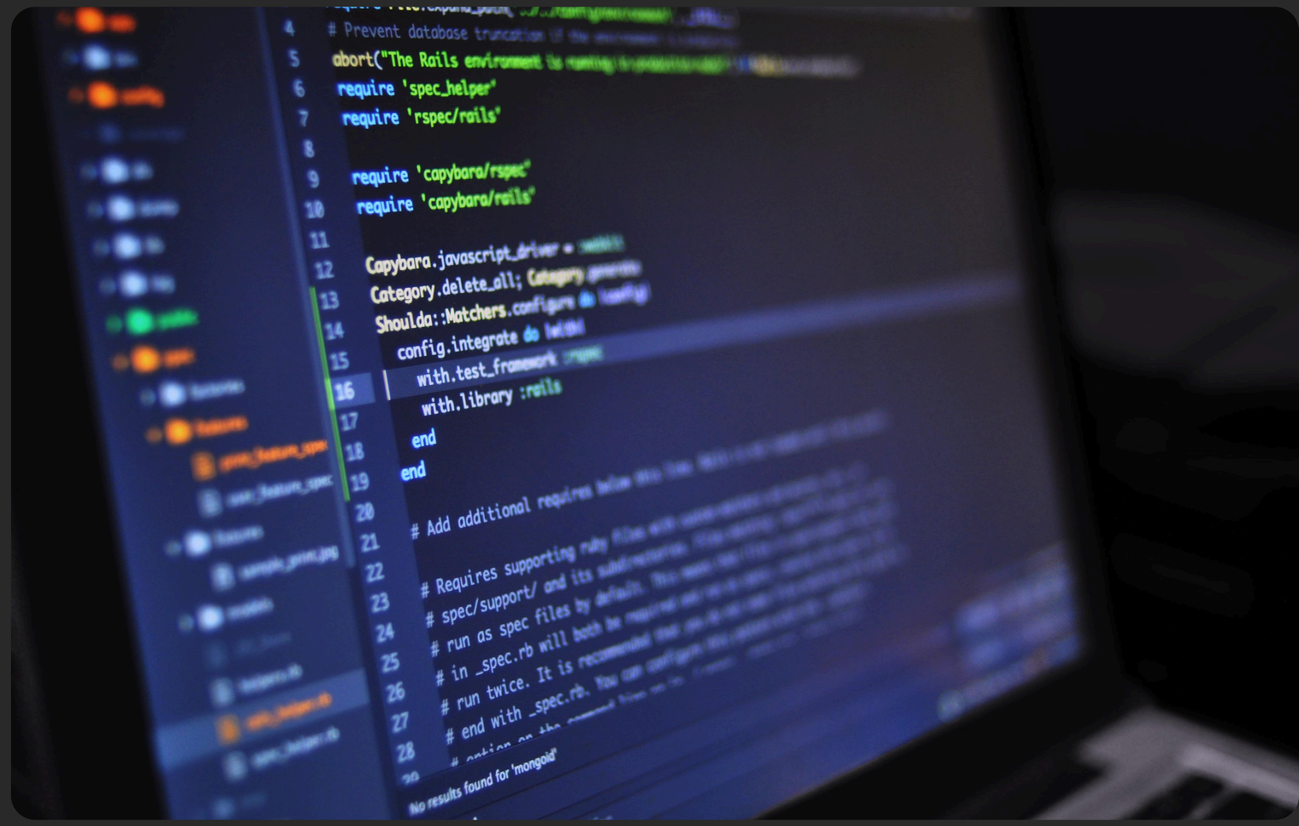
TESTE DE SOFTWARE



Equipe: Artur Paes de Medeiros e Pedro Paulo Fernandes Cardoso

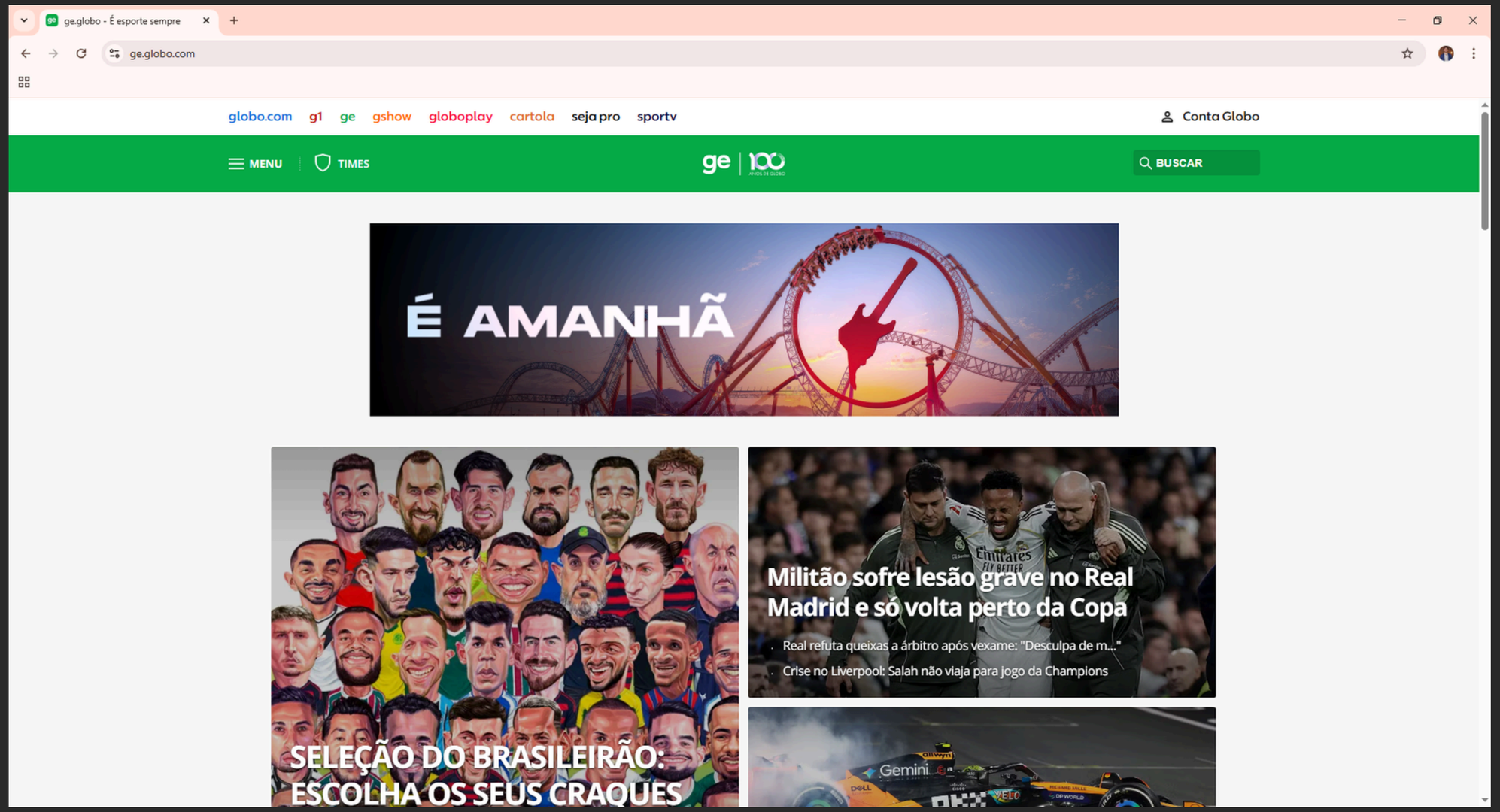
Site Escolhido

- Escolhemos o site do ge(antigo Globo Esporte), que é o portal de esportes da Grupo Globo — ele reúne notícias, reportagens, estatísticas e cobertura esportiva sobre futebol, vôlei, basquete, surfe, tênis, lutas, Olimpíadas e várias outras modalidades.

- No site há notícias em texto, análises, estatísticas e tabelas de campeonatos (nacionais e internacionais), bem como resultados, partidas do dia, mercado de transferências, e conteúdo multimídia (imagens, vídeos).

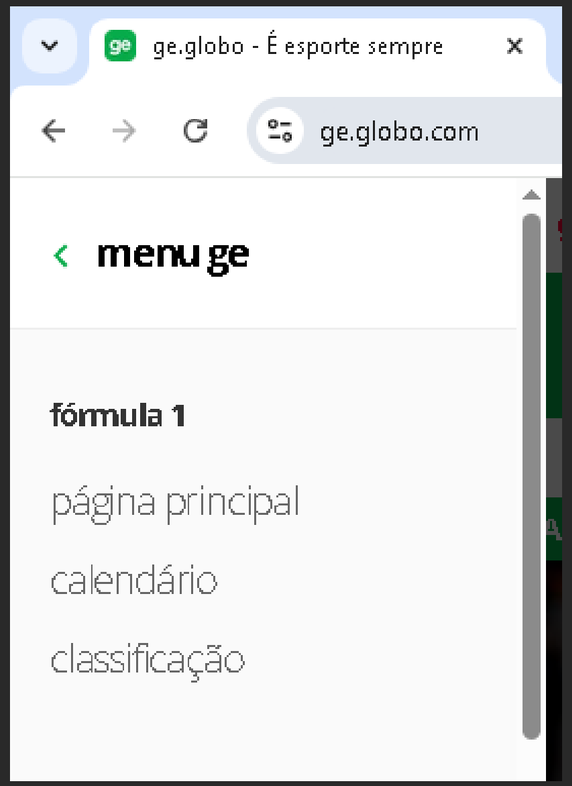
- Link do site: https://ge.globo.com

Print do site:

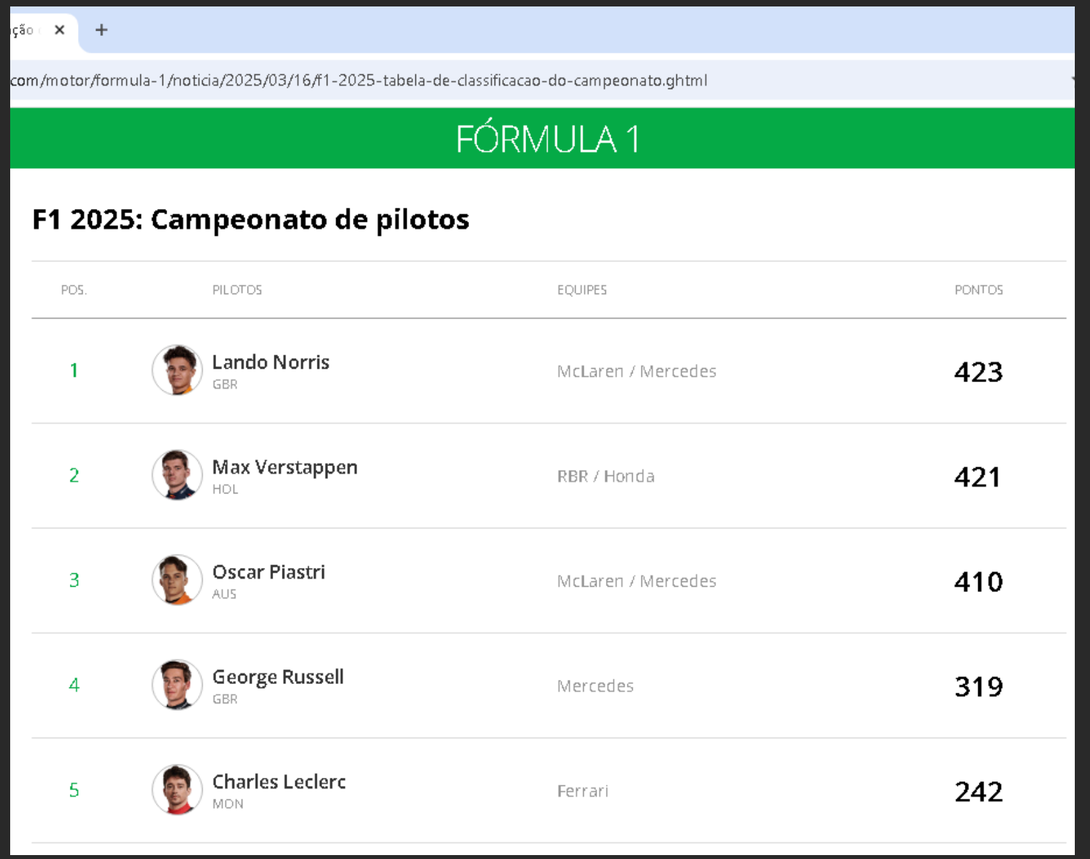


Primeiro Teste

- O que o teste faz: Navega pelo menu lateral até a classificação da Fórmula 1 e valida a classificação.

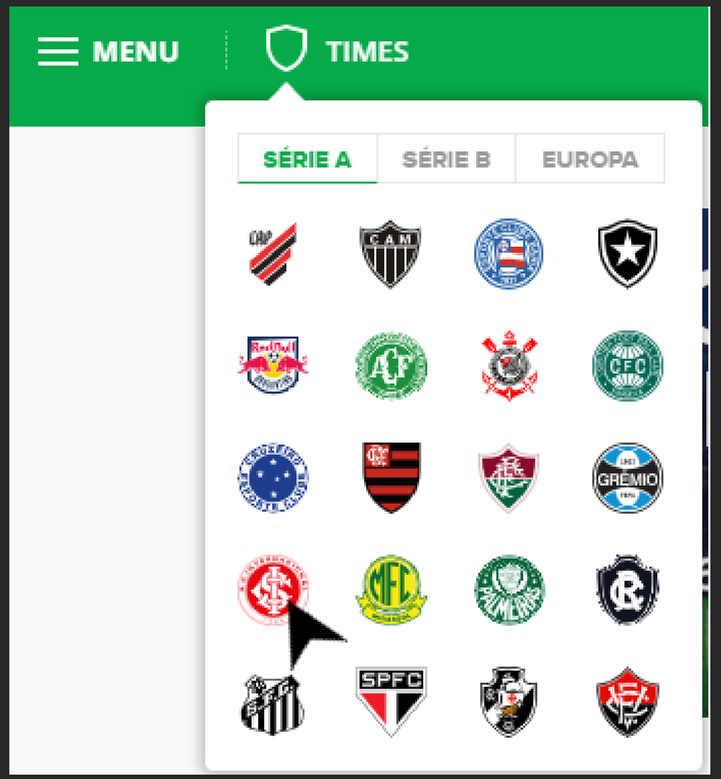


- Resultado Esperado: A página de classificação deve carregar e a tabela de pilotos deve estar visível na tela.

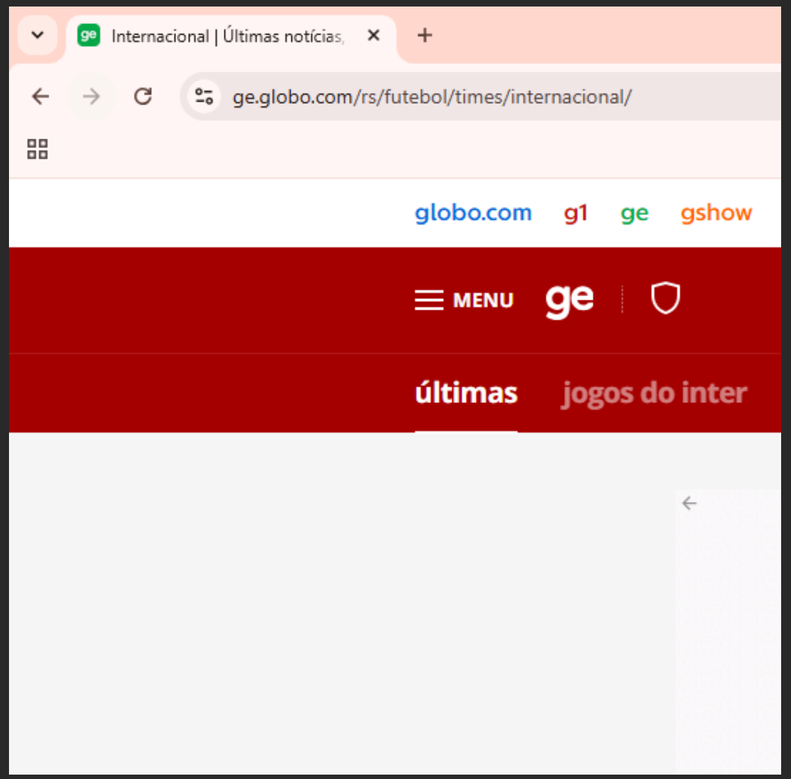


Segundo Teste

- O que o teste faz: Clica no escudo do Internacional, na aba de times no menu superior, e verifica se vai para a página do time.

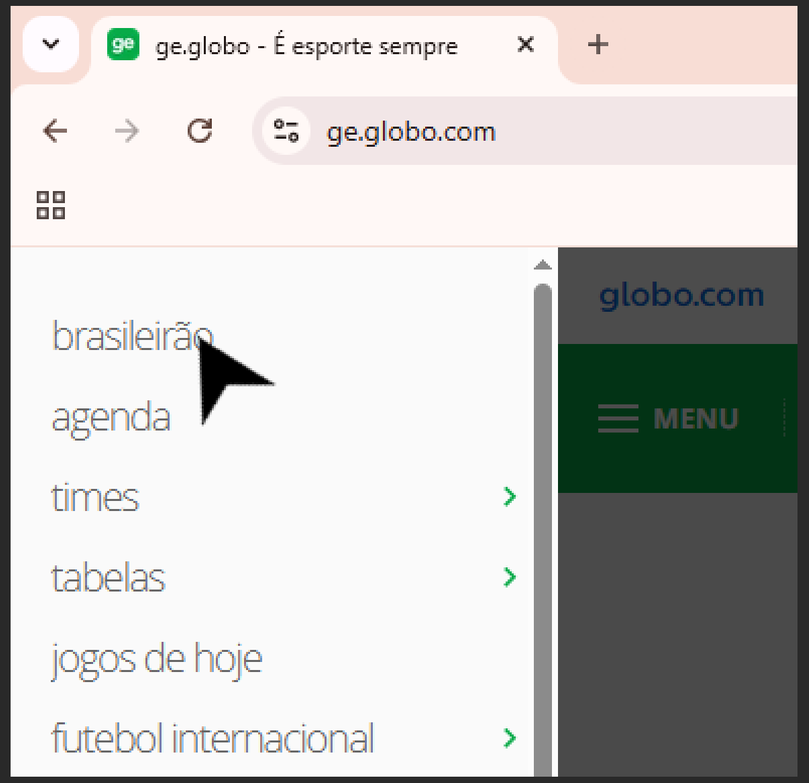


- Resultado Esperado: URL deve conter "times/internacional".

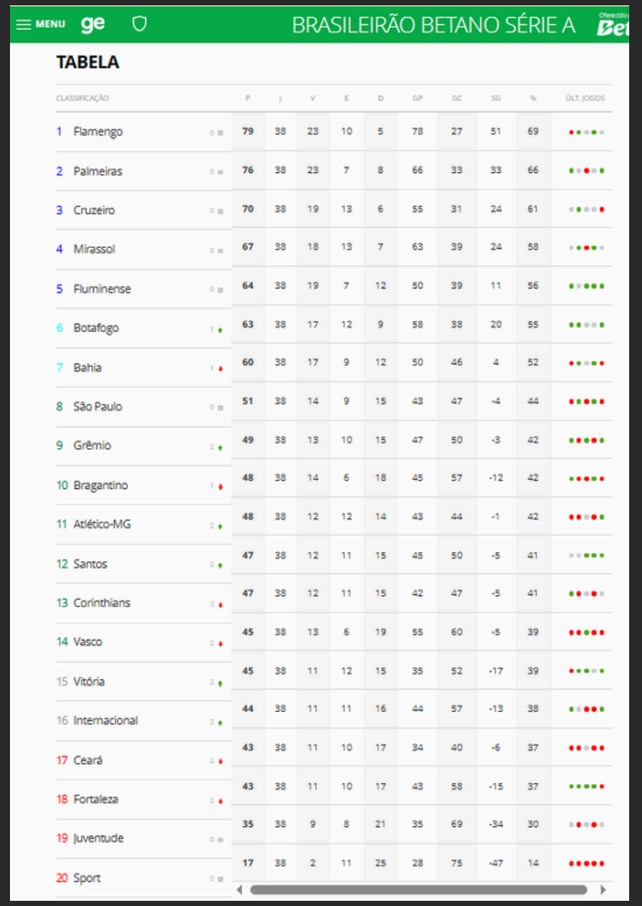


Terceiro Teste

- O que o teste faz: Abre o menu lateral e clica na opção 'brasileirão'. Após, verifica se a tabela carrega a lista completa de times.

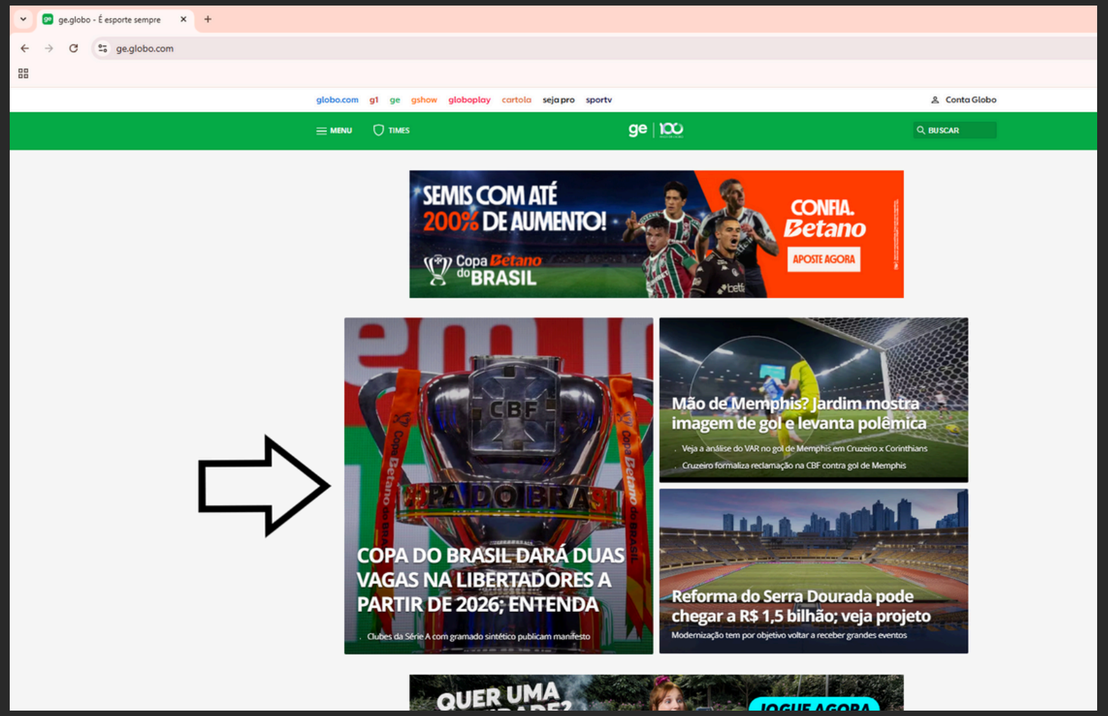


- Resultado Esperado: Devem ser encontrados pelo menos 20 times (Série A).

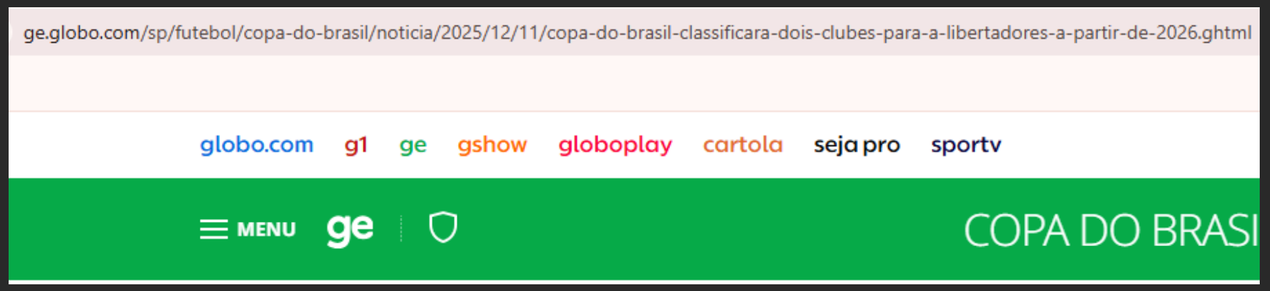


Quarto Teste

- O que o teste faz: Procura a primeira notícia com apenas texto no artigo e a abre.
- Resultado Esperado: A URL aberta deve corresponder ao link da manchete.




```
<a class="bstn-hl-link" href="https://ge.globo.com/sp/futebol/copa-do-brasil/noticia/2025/12/11…brasil-classificara-dois-clubes-para-a-libertadores-a-partir-de-2026.ghtml" data-mrf-layout-anchor data-mrf-link="https://ge.globo.com/sp/futebol/copa-do-brasil/noticia/2025/12/11/copa-do-brasil-classificara-dois-clubes-para-a-libertadores-a-partir-de-2026.ghtml" cmp-ltrk="Home - Destaques" cmp-ltrk-idx="4" mrfobservableid="5aef3528-eea9-4bf7-b0e4-7c8869b75618">
```



Trechos de código

```
class HomePage extends BasePage {
  constructor(driver) {
    super(driver);

    this.botaoMenu = By.css(".gl-header__menu-button, div.menu-button");
    this.menuF1 = By.id("menu-1-formula-1");
    this.linkClassificacao = By.css("#menu-2-classificacao a");
  }

  async navegarParaClassificacaoF1() {
    await this.click(this.botaoMenu);
    await this.driver.sleep(2000);

    const f1Element = await this.find(this.menuF1);
    await f1Element.click();
    await this.driver.sleep(2000);

    const classifElement = await this.find(this.linkClassificacao);
    await this.driver.executeScript("arguments[0].click();", classifElement);

    await this.driver.wait(until.urlContains("classificacao"), 10000);
  }
}
```

```
it("Deve navegar pelo menu até a classificação da F1 e validar a tabela", async
function () {
  await homePage.navegarParaClassificacaoF1();

  const tabela = await f1Page.obterTabela();

  const estaVisivel = await tabela.isDisplayed();
  await driver.sleep(3000);

  assert.isTrue(
    estaVisivel,
    "A tabela de pilotos deveria estar visível após a navegação"
  );
});
```

– Encapsulamento (Classe HomePage):

Na classe, centralizamos seletores complexos (como By.id("menu-1-formula-1")) e a lógica técnica de interação (clicar, esperar elemento). O método navegarParaClassificacaoF1 esconde toda a "sujeira" do Selenium, permitindo que, se o menu do site mudar, a manutenção seja feita apenas neste arquivo.

– Legibilidade e Abstração (Teste it):

No script de teste, o código transforma-se numa narrativa simples. A linha await homePage.navegarParaClassificacaoF1() descreve a ação do utilizador de forma clara, sem poluir o teste com driver.findElement ou seletores CSS longos.

— Configuração pré-teste (Setup)

```
describe("Automação de Testes - GE Globo", function () {
  this.timeout(60000);

  let driver;
  let homePage;
  let f1Page;

  beforeEach(async function () {
    let options = new chrome.Options();
    options.addArguments("--start-maximized");
    options.addArguments("--disable-notifications");

    driver = await new Builder()
      .forBrowser("chrome")
      .setChromeOptions(options)
      .build();

    homePage = new HomePage(driver);
    f1Page = new ClassificacaoF1Page(driver);

    await homePage.visit("https://ge.globo.com");
  });
```

— Teste Legível e Validação(Assertion)

```
  it("Deve abrir a notícia correta ao clicar na manchete principal", async function
  () {
    const expectedUrl = await homePage.getMainHeadlineUrl();
    console.log("URL esperada da manchete: " + expectedUrl);
    expect(expectedUrl).to.not.be.null;
    await homePage.clickMainHeadline();
    await driver.wait(until.urlContains(expectedUrl), 5000);
    const currentUrl = await driver.getCurrentUrl();
    console.log("URL atual aberta: " + currentUrl);
    await driver.sleep(3000);
    expect(currentUrl).to.include(expectedUrl);
  });
});
```

- Mocha (TestRunner): Responsável pela estrutura do teste. Utilizamos o describe para agrupar a suite "GE Globo" e o beforeEach para garantir que, antes de cada cenário, o browser abre limpo e os Page Objects são reiniciados, garantindo a independência dos testes.

- Chai (Assertion Library): Utilizada para realizar as validações. Através da interface expect, o Chai permite escrever verificações semânticas que se assemelham à linguagem natural (ex: expect(url).to.include(...)), tornando o teste autoexplicativo e fácil de manter."

Dificuldades Encontradas

- Validação de Conteúdo (Manchete)

- Problema: O texto da manchete algumas vezes diferia do título interno da matéria quebrando assim a validação por texto que havia sido implementada.

- Solução: Alteramos a estratégia de validação para Conferência de URL. Capturamos o href (link) da manchete antes do clique e validamos se a URL final continha esse mesmo link, criando um teste muito mais confiável.

Testes Rodando

Link do vídeo: https://youtu.be/odVqcHMOKwY