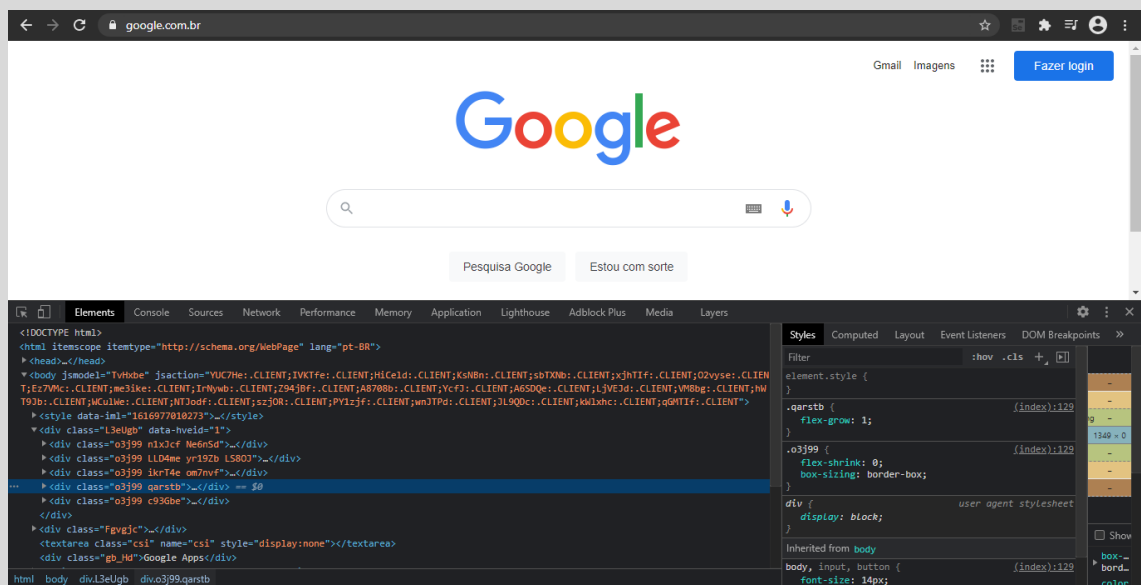


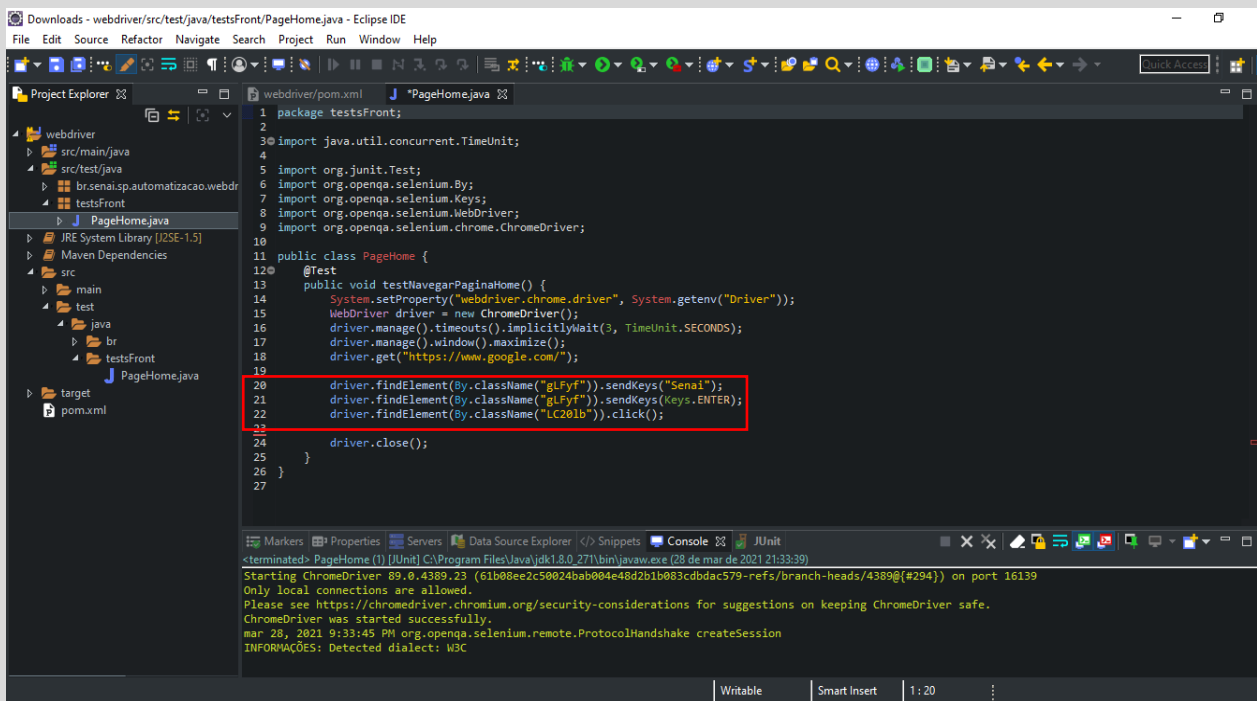
Identificando elementos do site

1. Na janela do navegador, clique com o botão direito do mouse e selecione “**Inspecionar**” para abrir as janelas de ferramentas do desenvolvedor, como mostra a imagem. Você pode utilizar, também, as teclas de atalho F12 ou Ctrl + Shift + i.



2. Para simular o clique do usuário, é necessário especificar, no código, qual elemento será testado, ou seja, qual elemento será clicado ou selecionado. Para determinar o elemento, utilizamos id, class, title, name, texto, xpath, entre outros.

Em nosso exemplo, determinaremos que o programa acesse o site do Google, clique no input, pesquise por “Senai”, aperte “enter” e clique no primeiro link que aparecer como resultado. Parece difícil, mas tudo é feito por meio dos comandos existentes nas linhas 20, 21 e 22, conforme destacado na imagem:



```
1 package testsFront;
2
3 import java.util.concurrent.TimeUnit;
4
5 import org.junit.Test;
6 import org.openqa.selenium.By;
7 import org.openqa.selenium.Keys;
8 import org.openqa.selenium.WebDriver;
9 import org.openqa.selenium.chrome.ChromeDriver;
10
11 public class PageHome {
12     @Test
13     public void testNavegarPaginaHome() {
14         System.setProperty("webdriver.chrome.driver", System.getenv("Driver"));
15         WebDriver driver = new ChromeDriver();
16         driver.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);
17         driver.manage().window().maximize();
18         driver.get("https://www.google.com/");
19
20         driver.findElement(By.className('gLFyf')).sendKeys("Senai");
21         driver.findElement(By.className('gLFyf')).sendKeys(Keys.ENTER);
22         driver.findElement(By.className('LC201b')).click();
23
24         driver.close();
25     }
26 }
27
```

Starting ChromeDriver 89.0.4389.23 (61b08ee2c50024bab004e48d2b1b083cddac579-refs/branch-heads/4389@{#294}) on port 16139
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
mar 28, 2021 9:33:45 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFORMAÇÕES: Detected dialect: W3C

Na linha 15, a variável do driver para acessar o navegador é o browser Chrome. Na sequência, temos o método **findElement()**, que procura pelo elemento determinado dentro dos parênteses (uma class, no caso), e o método **sendKeys()**, que insere “Senai” no input. Na linha seguinte, repare que o parâmetro do **findElement** não muda, mas o do **sendKeys**, sim. Logo, a class procurada será a mesma, porém, em vez de inserir uma string no input, o código simula um “enter” (clique). Por fim, o código simula o clique no primeiro link e fecha a página.

Dica!

Lembre-se de que a class pode se repetir dentro do código, mas que o id é único!



```

1 package testsFront;
2
3 import java.util.concurrent.TimeUnit;
4 import org.junit.After;
5 import org.junit.Before;
6 import org.junit.Test;
7 import org.openqa.selenium.By;
8 import org.openqa.selenium.Keys;
9 import org.openqa.selenium.WebDriver;
10 import org.openqa.selenium.chrome.ChromeDriver;
11
12 public class PageHome {
13     private WebDriver driver;
14
15     @Before
16     public void setUp() {
17         System.setProperty("webdriver.chrome.driver", System.getenv("Driver"));
18         driver = new ChromeDriver();
19         driver.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);
20         driver.manage().window().maximize();
21         driver.get("https://www.google.com/");
22     }
23
24     @Test
25     public void acessarSite() {
26         driver.findElement(By.className("gLFyf")).sendKeys("Senai");
27         driver.findElement(By.className("gLFyf")).sendKeys(Keys.ENTER);
28         driver.findElement(By.className("LC20lb")).click();
29     }
30
31     @After
32     public void fechar() {
33         driver.close();
34     }
35 }
36

```

Note que os elementos foram alterados. Sobre os @ adicionados:

- **@Before:** método público; executa comandos antes do teste.
- **@Test:** Método público; executa o teste; pode ser inserido mais de um no código.
- **@After:** método público; executa comandos depois do teste.

O WebDriver foi deixado fora do escopo @Before, pois nos blocos de @Before, @Test e @After é aplicado o conceito de programação orientada a objetos. Em outras palavras, o escopo que estiver dentro das chaves { } rodará apenas ali, ou seja, o WebDriver não seria reconhecido pelos demais escopos se estivesse dentro dessas chaves.