

Automação de testes com Selenium WebDriver Esperas

Antônio Moraes Trindade

<https://about.me/amtrindade>

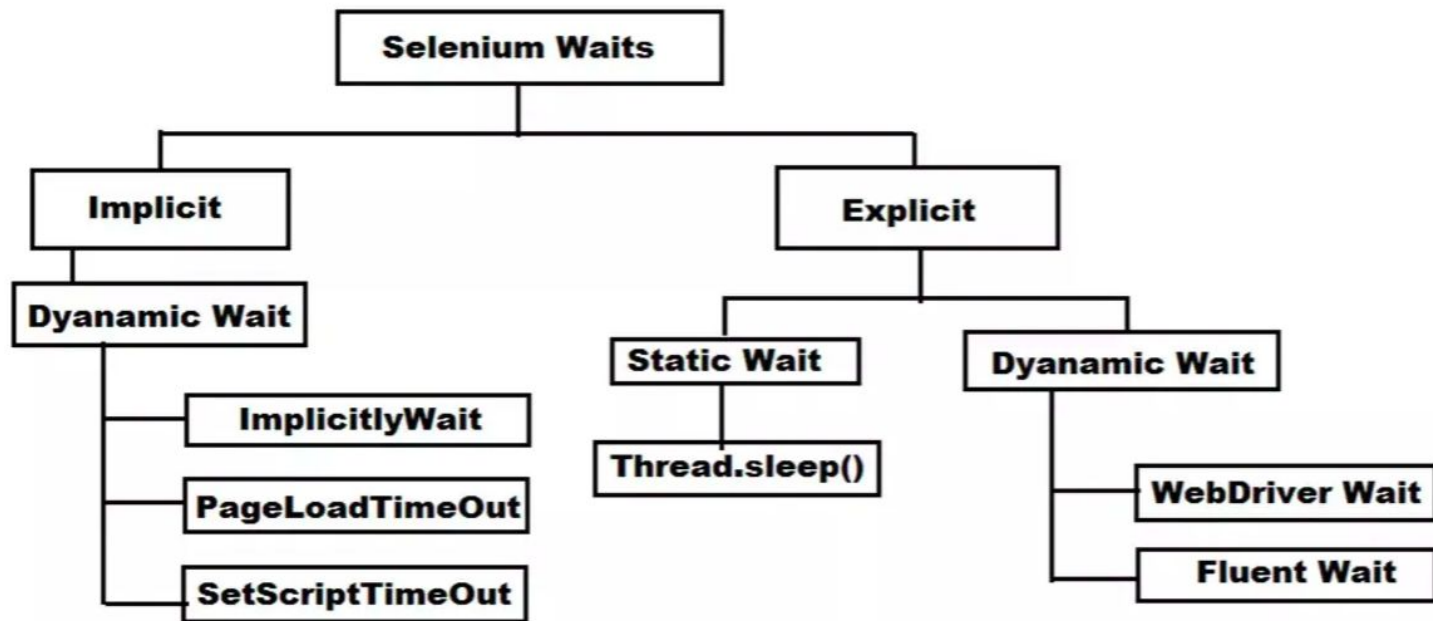
Esperas com WebDriver



Existem 3 formas de implementarmos as esperas:

- Espera fixa
- Espera implícita
- Espera explícita

Esperas com WebDriver



• Fluent Wait

1. Espera fixa

Tempo fixo. Não é recomendado. NÃO UTILIZE!!!

Razões:

1. Nunca é possível prever o tempo exato de espera, tendência é sempre definirmos um tempo maior que o necessário.
2. Aumenta a sobrecarga de tempo execução dependendo da quantidade de testes da suíte.

```
Thread.sleep(3000);
```

2. Espera implícita

UTILIZE ESTE!!!

Faz o driver esperar por um elemento, que não esteja disponível imediatamente. O driver espera até o tempo máximo definido para lançar a exceção NoSuchElementException.

Normalmente utilizado no setup da classe.

A espera está condicionada a um request.

```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

3. Espera explícita

UTILIZE ESTE!!!

O WebDriverWait é utilizado quando precisarmos esperar por um elemento específico. Espera o elemento específico para dar prosseguimento ao teste.

O objeto neste caso é constantemente verificado, o pooling default é de 0,5 segundos.

```
WebDriverWait wait = new WebDriverWait(driver, 10);  
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("botaoOk")));
```

A classe Expected Condition

Nesta classe existem várias condições pré-definidas que podemos utilizar em nossos testes.

Vamos ver algumas delas!!

Visibilidade dos elementos

▣ `visibilityOfElementLocated`

Dica: opte por essa em vez da "presenceOfElement". A "presence" verifica apenas que o elemento está no DOM, enquanto a "visibility" verifica que o elemento está presente e enabled (ou seja, disponível para interação).

Texto presente no elemento

▣ `textToBePresentInElement`

Espera até que uma mensagem de sucesso apareça em um elemento. Para aguardar até que um texto esteja presente em um `TextField`, use a `ExpectedCondition` `"textToBePresentInElementValue"`

Aguarda checkbox selecionado

▣ **elementToBeSelected /**

elementSelectionModeToBe(By locator, true)

Espera até que o CheckBox ou RadioButton esteja selecionado para dar sequência ao teste quando o segundo parâmetro for “true”.

Aguarda checkbox deselecionado

▣ **elementToBeSelected /**

elementSelectionStateToBe(By locator, false)

Espera até que o CheckBox ou RadioButton esteja deselecionado para dar sequência ao teste quando o segundo parâmetro for “false”.

FluentWait

É uma implementação da interface Wait que permite configurar o tempo de timeout, o intervalo de polling e adicionar ignores para exceções específicas.

```
Wait<WebDriver> waitFluent = new FluentWait<WebDriver>(driver)
    .withTimeout(10, TimeUnit.SECONDS)
    .pollingEvery(1, TimeUnit.SECONDS)
    .ignoring(NoSuchElementException.class)
    .ignoring(StaleElementReferenceException.class);

waitFluent.until(ExpectedConditions.visibilityOf(element));
```