

09 - Esperando elementos serem carregados

Roteiro

- Carregamento da página
 - Normal
 - Async
 - Defer
- Tipos de esperas
 - Implícita
 - Explícita
- Wait
- By
- Locators
- Esperas personalizadas



“Chegamos na metade do
curso”

@dunossauro



picpay.me/dunossauro | apoia.se/livedepython

```
dunossauro at babbage in ~/git/apoiase on master*  
$ python apoiadores.py  
ADRIANO FERRARI      ALBERTO TAVARES      Alex Lima             Alexandre Possebom    Alexandre Santos  
Alexandre Sá         Alexandre Tsuno       Alexandre Villares    Alynne Ferreira       Alysson Oliveira  
Amaziles Carvalho    And Past              Andre Machado         Andre Rodrigues        Athayr Athayr  
Bernardo Fontes      Bruno Barros          Bruno Oliveira        Bruno Gaffuri          Bruno Guizi  
Bruno Luna           Bruno Lira            Bruno Rocha           CARLOS SANTOS         Caio Nascimento  
Carlos Cardoso       Carlos Augusto        Carol Souza           Cleber Santos          Cleiton Mittmann  
Cleiton Lima         DYLAN MAKER          Davi Ramos            David Reis             David Silva  
Davinir Junior       Dayham Soares        Diego Guimarães       Diego Ubirajara        Dilenon Delfino  
Douglas Bastos      EDUARDO PAZZI        Edimar Fardim         Eduardo Guadanhim      Eduardo Nunes  
Eliabe Silva         Eliakim Morais       Elias Soares          Elias Soares           Eliel Lima  
Emerson Lara         Emerson Lara          Eugenio Mazzini       Eugenio Mazzini        Fabiano Silos  
Fabiano Teichmann    Fabiano Gomes        Fabrício Coelho       Fausto Caldeira        Fernando Furtado  
Filipe Cruz          FlavKaze FlavKaze     Franklin Silva        Fábio Serrão           Gabriel Simonetto  
Gabriela Santiago    Geandreson Costa     Gladson Menezes       Gleison Oliveira      Guilherme Ramos  
Helter Silva         Hemilio Lauro        Humberto Rocha        Hélio Neto            Hélio Neto  
Isaac Ferreira       JONATHAN DOMINGOS    Jean Vetorello        Johnny Tardin          Jonatas Oliveira  
Jonatas Simões       Jonatas Oliveira     Jorge Plautz          José Prado             Jovan Costa  
João Lúgão           João Coelho          Juan Gutierrez        Jucélio Silva          Júlia Kastrup  
Kauan Alves          LEONARDO CRUZ        Leon Teixeira         Leonardo Rezende       Letícia Silva  
Lucas Mendes         Lucas Nascimento      Lucas Neris           Lucas Valino           Lucas Polo  
Magno Malkut         Maiquel Leonel       Marcello Benigno      Marcus Salgues         Maria Boladona  
Maria Clara          Mateus Braga         Matheus Francisco     Melisa Campagnaro     Nilo Pereira  
Nídio Dolfini        Octavio Sydow        Pablo Henrique        Patrick Corrêa         Patrick Gomes  
Paulo Tadei          Pedro Alves          Rafael Galleani       Rafael Dias            Regis Santos  
Renan Gomes          Renan Moura          Renato Santos         Rennan Almeida         Renne Rocha  
Rhenan Bartels       Ricardo Schalch      Richard Nixon         Rodrigo Ferreira       Rodrigo Vaccari  
Régis Tomkiel        Sérgio Passos        Thais Viana           Thiago Araujo          Tiago Cordeiro  
Tyrone Damasceno     Valdir Junior        Valdir Silveira       Vergil Valverde        Vicente Marcal  
Victor Geraldo       Vinícius Ferreira    Vitor Dutra           Vítor Zanoni           Wander Silva  
Wellington Carlos    Wellington Camargo    Welton Souza          William Oliveira       William Oliveira  
Willian Lopes        Willian Lopes        Willian Rosa          Yros Aguiar            Falta você
```

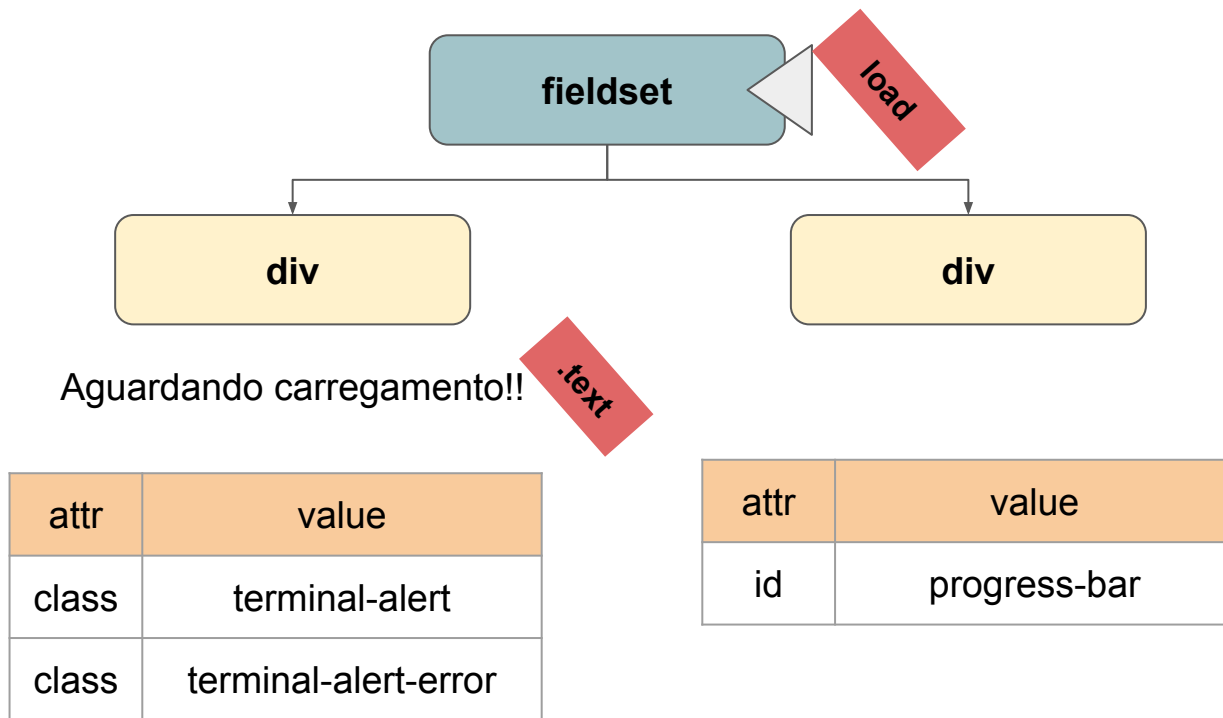


Carregamento da página

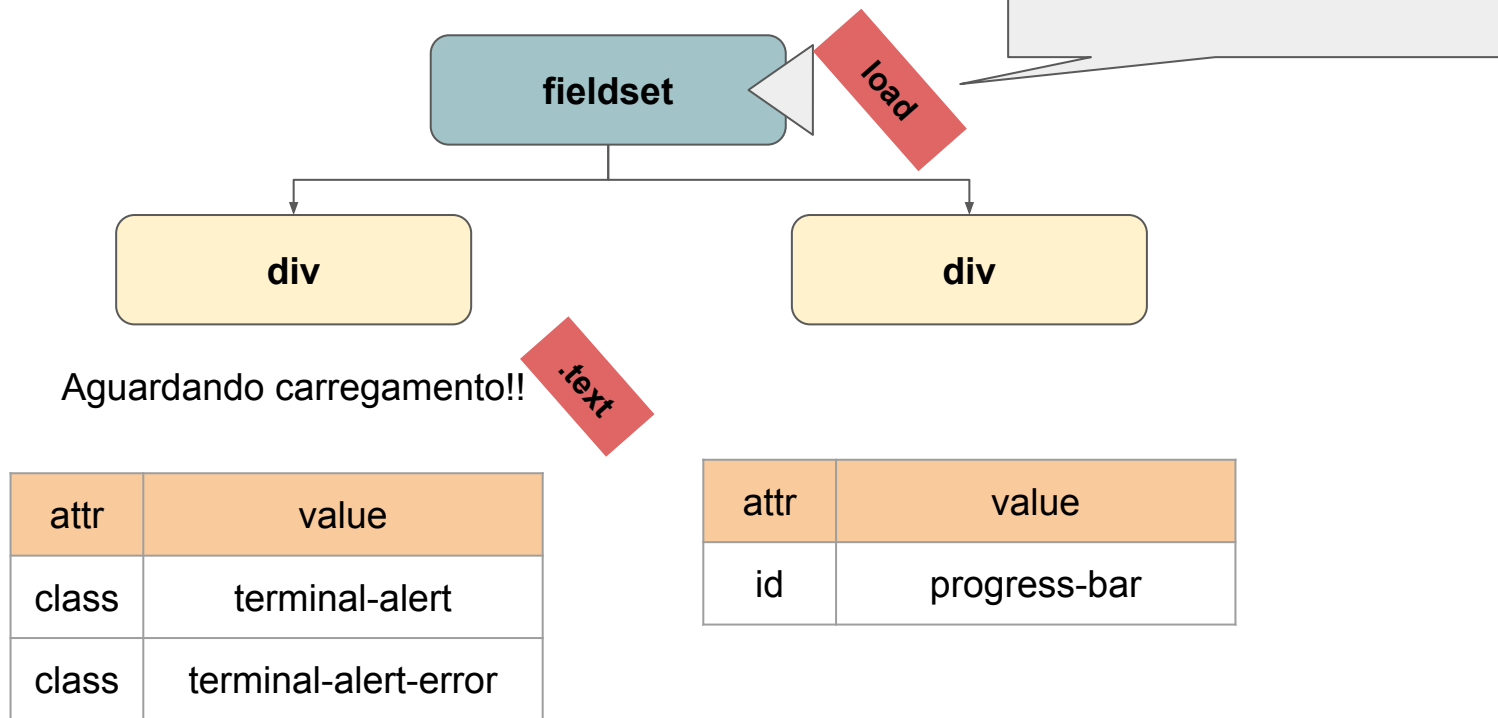
https://selenium.dunossauro.live/aula_09_a.html



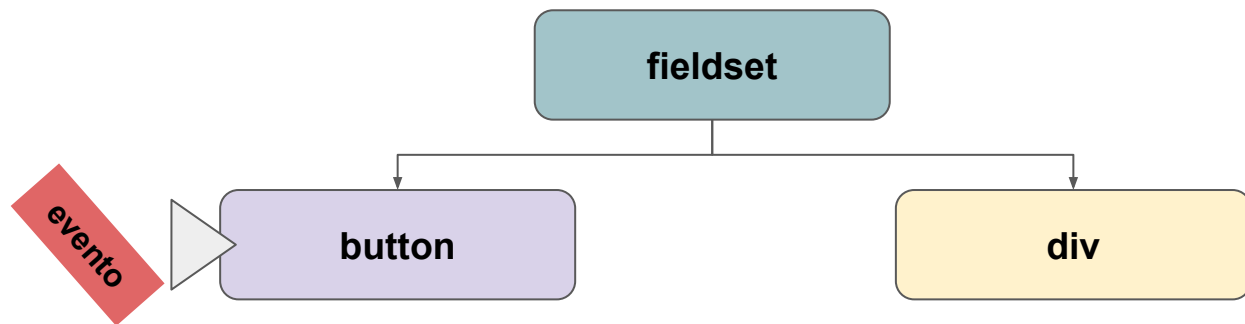
Página A [Antes do carregamento]



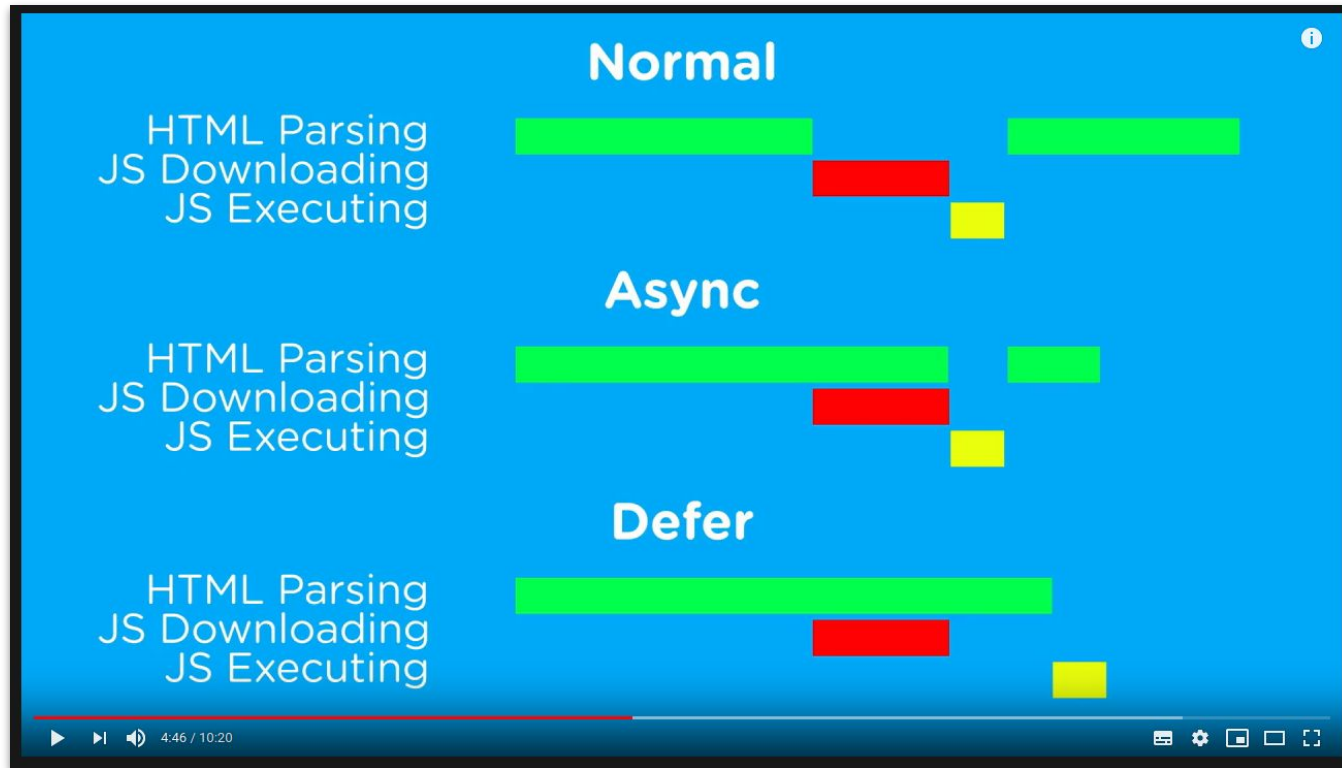
Página A [Antes do carregamento]



Página A [Depois do carregamento]



| attr | value |
|------|--------------|
| id | progress-bar |



<https://www.youtube.com/watch?v=BMuFBYw91UQ>



Tipos de espera

https://selenium.dunossauro.live/aula_09_a.html



Tipos de espera

O Selenium conta com dois tipos de espera:

- Explícitas
- Implícitas
 - Espera todos os elementos, eventos, navegação, com um tempo padrão



Tipos de espera

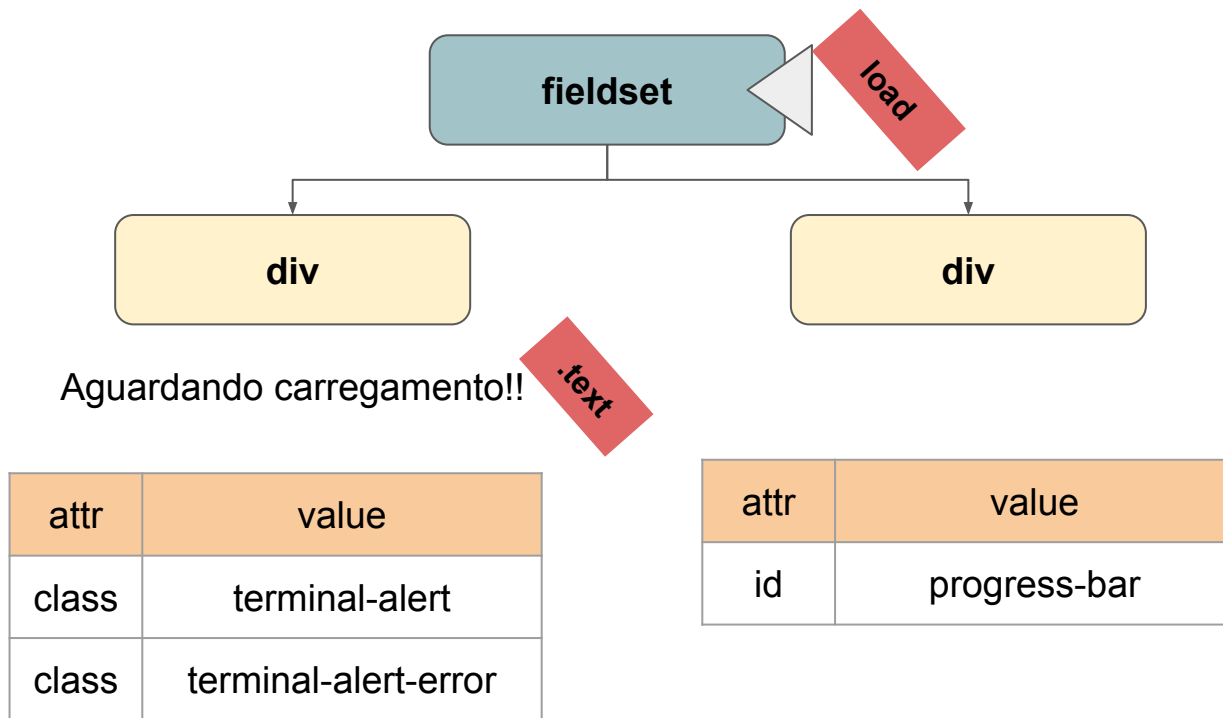
O Selenium conta com dois tipos de espera:

- Explícitas
- **Implícitas**
 - Espera todos os elementos, eventos, navegação, com um tempo padrão

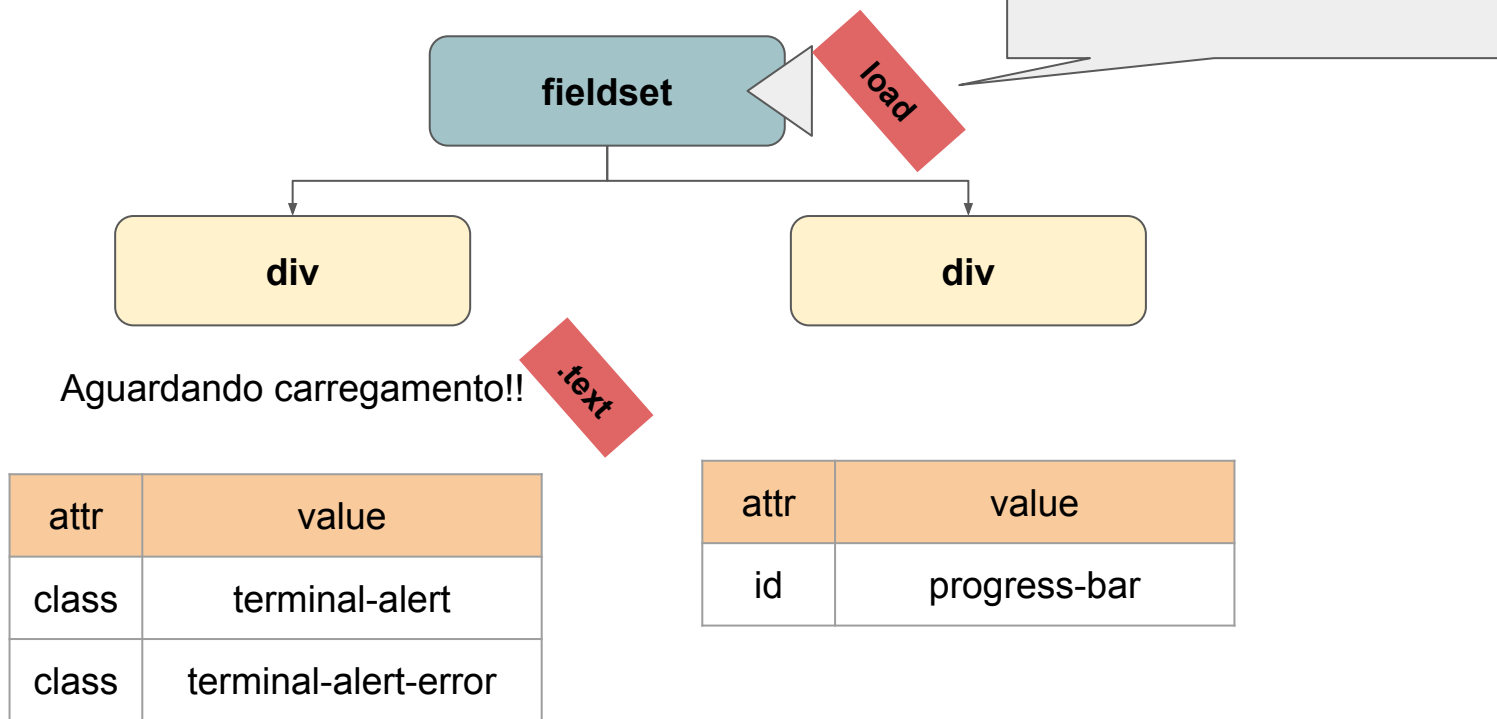
```
from selenium.webdriver import Firefox  
  
f = Firefox()  
f.implicitly_wait(30)
```



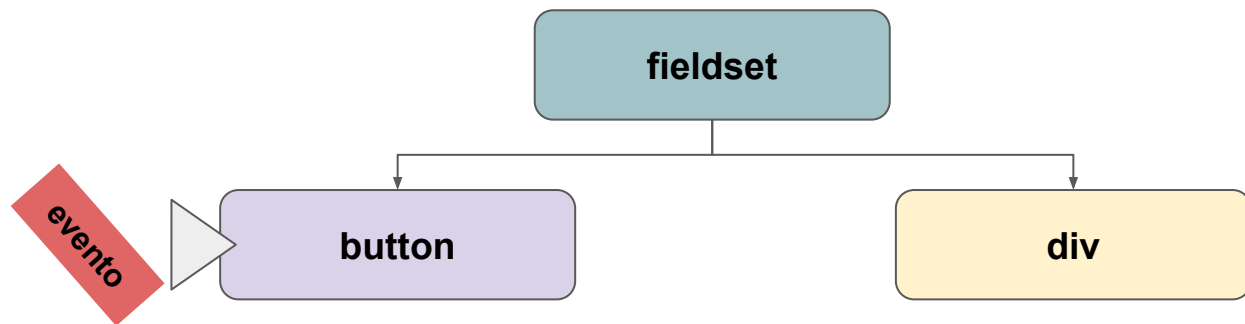
Página A [Antes do carregamento]



Página A [Antes do carregamento]

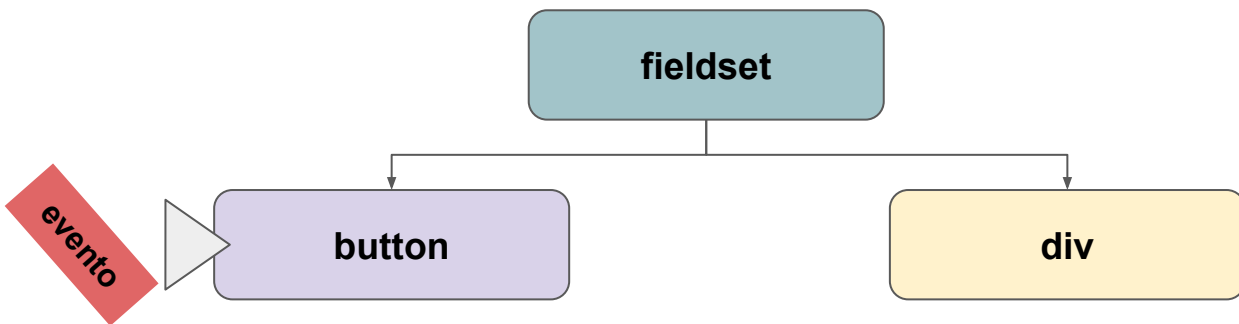


Página A [Depois do carregamento]



| attr | value |
|------|--------------|
| id | progress-bar |

Página A [Evento do botão]



onclick:

1. Remove div.terminal-alert, existir
2. Cria barra de progresso
3. Corre barra de carregamento
4. remove barra
5. Adiciona nova div.terminal-alert

| attr | value |
|------|--------------|
| id | progress-bar |

Bora pro código, jovens

https://selenium.dunossauro.live/aula_09_a.html



Wait implícito

Prós

- Funciona em cenários **Flaky**
- Não se preocupe, nós vamos esperar

Contras

- Segura a aplicação por mais tempo
- Tudo é esperado tendo o mesmo tempo como base
- Não funciona para elementos específicos
- Se algo der errado, vai demorar o tempo do wait para saber



Tipos de espera

O Selenium conta com dois tipos de espera:

- Implícitas
 - Espera todos os elementos, eventos, navegação, com um tempo padrão
- **Explícitas**
 - Selenium disponibiliza um range de waits prontos
 - Customizável
 - Em tipos de espera
 - É possível criar suas próprias esperas
 - Reutilizável



WebDriverWait

WebDriverWait

```
from selenium.webdriver.support.ui import WebDriverWait

wdw = WebDriverWait(
    driver, # Webdriver
    timeout, # Tempo de espera até o erro
    poll_frequency=0.5, # Tempo entre uma tentativa e outra
    ignored_exceptions=None # Lista de coisas que vamos ignorar
)
```

WebDriverWait

```
from selenium.webdriver.support.ui import WebDriverWait
```

```
wdw = WebDriverWait(  
    driver, # Webdriver  
    timeout, # Tempo de espera até o erro  
    poll_frequency=0.5, # Tempo entre uma tentativa e outra  
    ignored_exceptions=None # Lista de coisas que vamos ignorar  
)
```


Declaração

```
from selenium.webdriver import Firefox
from selenium.webdriver.support.ui import WebDriverWait

f = Firefox()

wdw = WebDriverWait(driver, 10) # segundos
```

WebDriverWait [until]

Executa até que o `Callable` retorne `True`, ou até estourar o `timeout` de wdw

```
wdw.until(  
    Callable, # Operação que vai ser executada  
    message # Mensagem caso o erro ocorra  
)
```

WebDriverWait [not_until]

Executa até que o `Callable` retorne `False`, ou até estourar o `timeout` de wdw

```
wdw.not_until(  
    Callable, # Operação que vai ser executada  
    message # Mensagem caso o erro ocorra  
)
```

Tudo junto agora [de forma usual]

```
from selenium.webdriver import Firefox
from selenium.webdriver.support.ui import WebDriverWait

browser = Firefox()
wdw = WebDriverWait(browser, timeout=30) # segundos
wdw.until(<FUNÇÃO>)
```



Tudo junto agora [de forma usual]

```
from selenium.webdriver import Firefox
from selenium.webdriver.support.ui import WebDriverWait

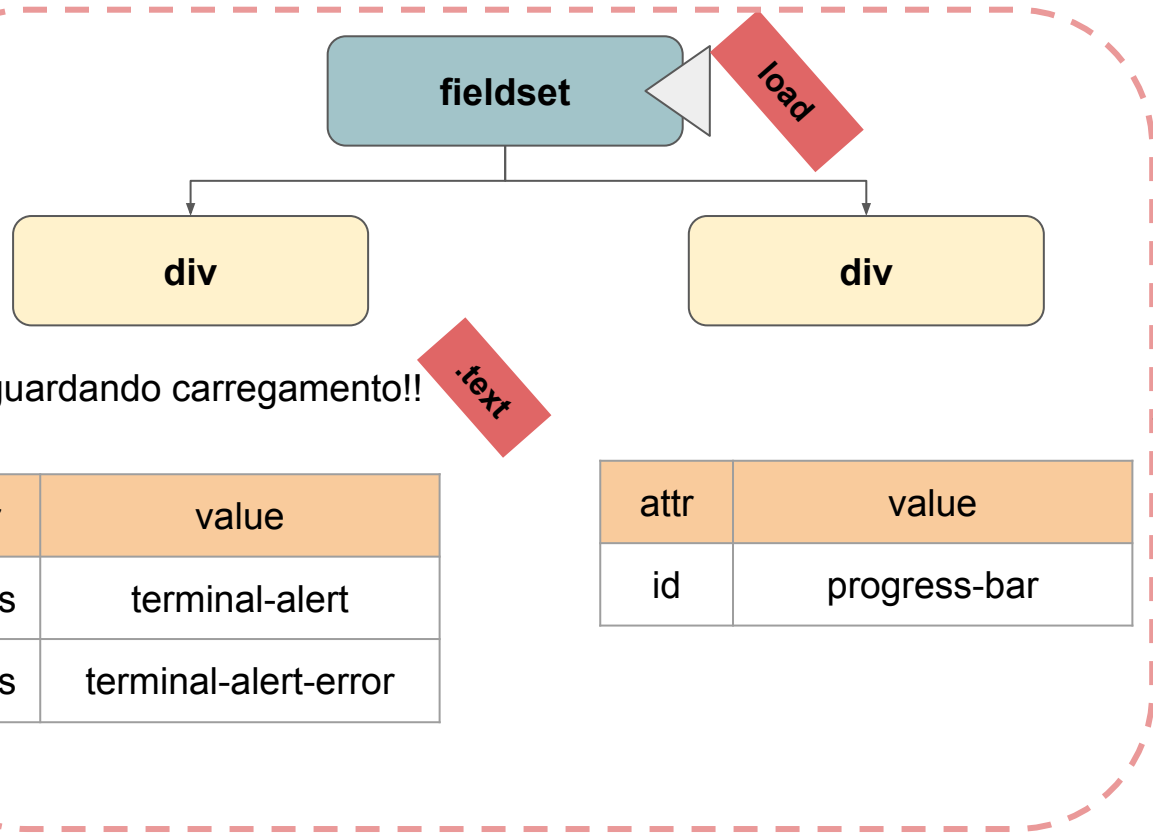
browser = Firefox()
wdw = WebDriverWait(browser, timeout=30) # segundos
wdw.until(<FUNÇÃO>)
```

Usando o WDW

https://selenium.dunossauro.live/aula_09_a.html

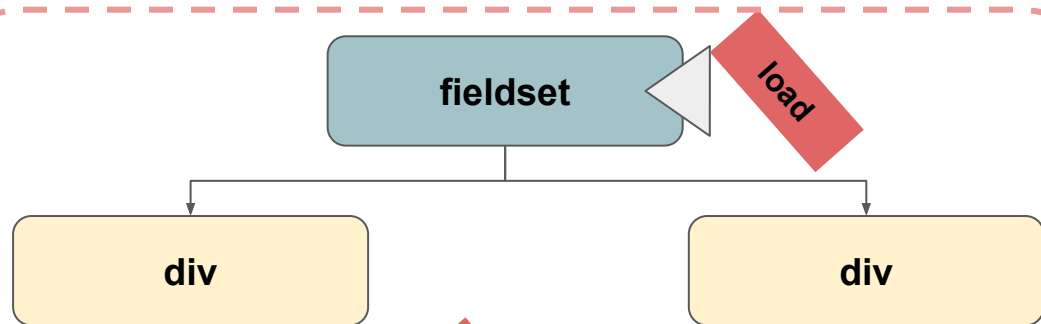


Página A [Antes do carregamento]



| TODOS | |
|-------|-----------------|
| 1 | Abrir página |
| 2 | Esperar botão |
| 3 | Clicar no botão |

Página A [Antes do carregamento]



Aguardando carregamento!!

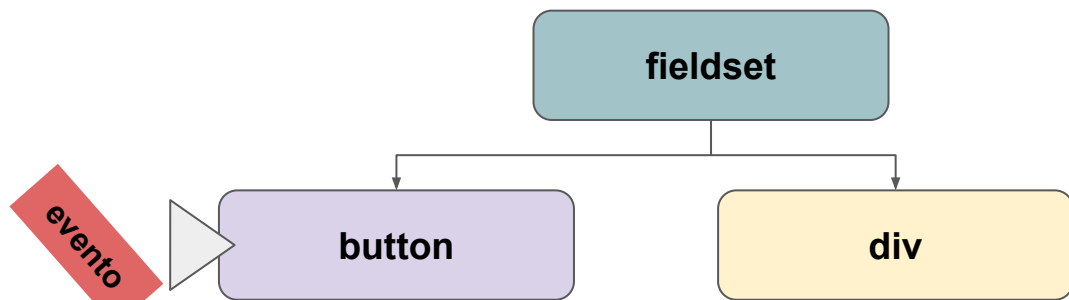
.text

| attr | value |
|-------|----------------------|
| class | terminal-alert |
| class | terminal-alert-error |

| attr | value |
|------|--------------|
| id | progress-bar |

| TODOS | |
|-------|-----------------|
| 1 | Abrir página |
| 2 | Esperar botão |
| 3 | Clicar no botão |

Página A [Depois do carregamento]



| TODOS | |
|-------|-----------------|
| 1 | Abrir página |
| 2 | Esperar botão |
| 3 | Clicar no botão |

| attr | value |
|------|--------------|
| id | progress-bar |

Esperas personalizadas



A mais simples, pra gente entender

```
def esperar_botao(webdriver):  
    """Verifica se o elemento `button` está na tela."""  
    elements = webdriver.find_elements_by_css_selector('button')  
    return bool(elements)  
  
wdw.until(esperar_botao)  
browser.find_element_by_tag_name('button').click()
```



A mais simples, pra gente entender

```
def esperar_botao(webdriver):  
    """Verifica se o elemento `button` está na tela."""  
    elements = webdriver.find_elements_by_css_selector('button')  
    return bool(elements)  
  
wdw.until(esperar_botao)  
browser.find_element_by_tag_name('button').click()
```

- until e not_until sempre passam o webdriver como argumento
 - **E SOMENTE ELE**
- A função sempre deve retornar True ou False

Agora vai

https://selenium.dunossauro.live/aula_09_a.html



functools partial



Parametrizando esperas

Fizemos esperas “especializadas”. Isso é complicado, pois todo código precisa ser feito “mais de uma vez”.

A solução é adicionar um novo parâmetro

```
def esperar_elemento(seletor, webdriver):  
    elements = webdriver.find_elements_by_css_selector(seletor)  
    return bool(elements)
```



Parametrizando esperas

Fizemos esperas “especializadas”. Isso é complicado, pois todo código precisa ser feito “mais de uma vez”.

A solução é adicionar um novo parâmetro

```
def esperar_elemento(seletor, webdriver):  
    elements = webdriver.find_elements_by_css_selector(seletor)  
    return bool(elements)
```

Tudo junto, agora

```
from functools import partial
from selenium.webdriver import Firefox
from selenium.webdriver.support.ui import WebDriverWait

def esperar_elemento(element, web_driver):
    if web_driver.find_elements_by_css_selector(element):
        return True
    return False

wdw = WebDriverWait(f, 30)
wdw.until(partial(esperar_elemento, 'selector'))
```



Vamos pro código

https://selenium.dunossauro.live/aula_09_a.html



By

https://www.selenium.dev/selenium/docs/api/py/_modules/selenium/webdriver/common/by.html



By

O By nos auxilia a deixar o código mais “assertivo”. Ele é comumente usado em conjunto o `wb.find_element`. Assim nos ajuda a parametrizar mais as nossas funções.

```
def esperar_elemento(element, web_driver):  
    if web_driver.find_elements_by_css_selector(element):  
        return True  
    return False
```

By

O By nos auxilia a deixar o código mais “assertivo”. Ele é comumente usado em conjunto o `wb.find_element`. Assim nos ajuda a parametrizar mais as nossas funções.

```
from selenium.webdriver.common.by import By

def esperar_elemento(by, element, web_driver):
    """Concatena o by com o elemento a ser buscado."""
    return bool(web_driver.find_elements(by, element))

wdw = WebDriverWait(f, 30)
wdw.until(partial(esperar_elemento, By.CSS_SELECTOR, 'selector'))
```



E volta cão arrependido

https://selenium.dunossauro.live/aula_09_a.html



Esperas com classes

Para fãs de OO

Aqui evitamos o uso do partial usando `__call__`.

Não é a minha abordagem preferida, mas é a tradicional em relação a lib.

```
class EsperarElemento:
    def __init__(self, by, selector):
        self.locator = (by, selector)

    def __call__(self, driver):
        if web_driver.find_elements(*locator):
            return True
        return False
```

```
web_driver_wait.until(EsperarElemento(By.ID, 'meu_id'))
```



Locators



Locators

Locators são maneiras de unir o **By** a string do **elemento**.

Porém esse conceito é trazido do selenium java e faz pouco ou nenhum sentido usando em python.

Resumidamente um **Locator** é uma tupla com o **By** e a string do **elemento**

```
from selenium.webdriver.common.by import By  
  
locator = (By.CSS_SELECTOR, 'div.minha_classe')
```



Locators

Locators são maneiras de unir o **By** a string do **elemento**.



Porém esse conceito é trazido do selenium java e faz pouco ou nenhum sentido usando em python.

Resumidamente um **Locator** é uma tupla com o **By** e a string do **elemento**

```
from selenium.webdriver.common.by import By  
  
locator = (By.CSS_SELECTOR, 'div.minha_classe')
```



Locators

```
from functools import partial
from selenium.webdriver.common.by import By

def esperar_elemento(locator, web_driver):
    if web_driver.find_elements(*locator):
        return True
    return False

locator = (By.CSS_SELECTOR, 'div.minha_classe')
web_driver_wait = WebDriverWait(f, 30)

web_driver_wait.until(
    partial(esperar_elemento, (By.ID, 'id do elemento'))
)
```



Locators

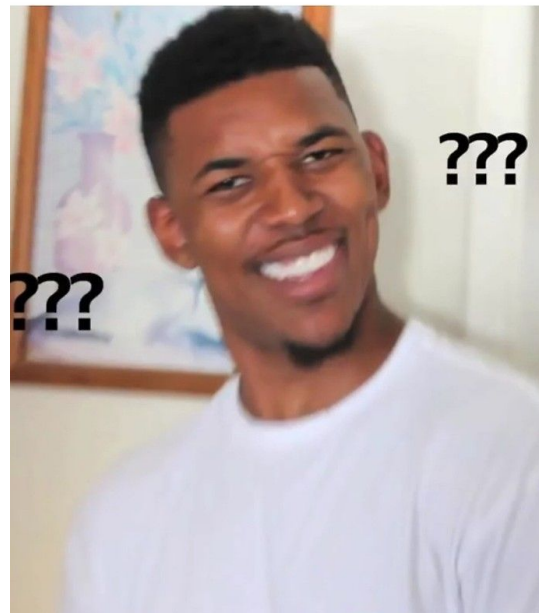
LOCATORS?

```
from functools import partial
from selenium.webdriver.common.by import By
```

```
def esperar_elemento(locator, web_driver):
    if web_driver.find_elements(*locator):
        return True
    return False
```

```
locator = (By.CSS_SELECTOR, 'div.minha_classe')
web_driver_wait = WebDriverWait(f, 30)
```

```
web_driver_wait.until(
    partial(esperar_elemento, (By.ID, 'id do elemento'))
)
```



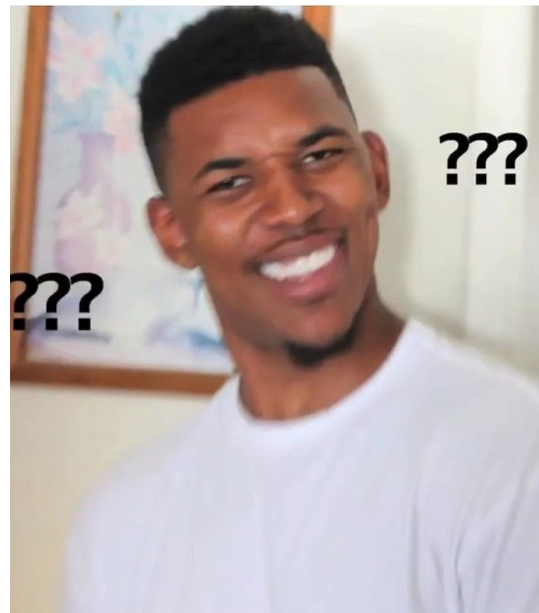
Locators

LOCATORS?

```
class EsperarElemento:
    def __init__(self, locator):
        self.locator = locator

    def __call__(self, driver):
        if web_driver.find_elements(*locator):
            return True
        return False

web_driver_wait.until(EsperarElemento((By.ID, 'meu_id')))
```



Sério, última vez dessa página...

https://selenium.dunossauro.live/aula_09_a.html

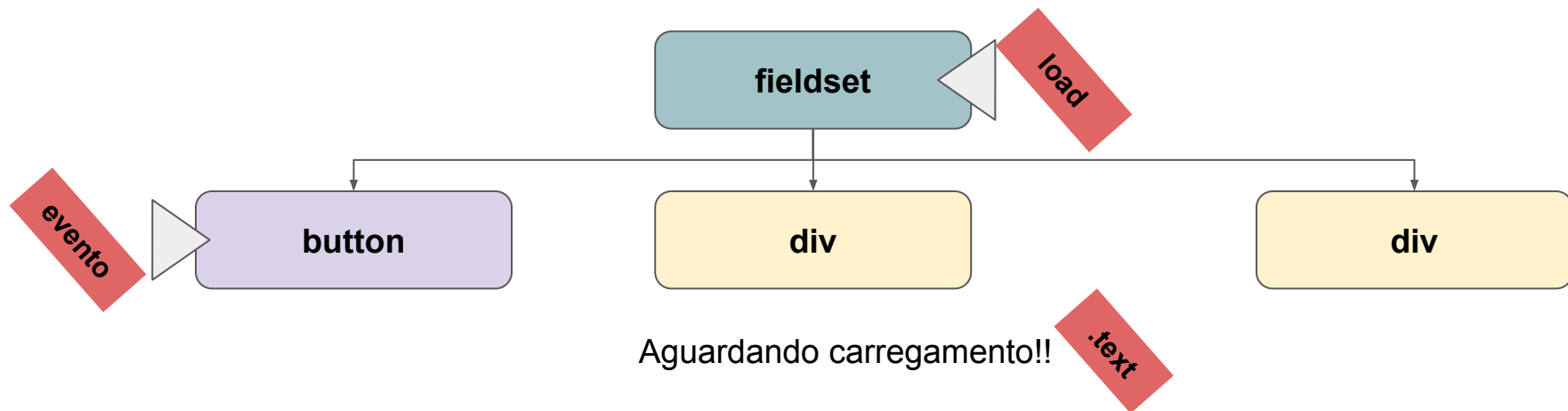


Agora vamos tentar usar em outra página

https://selenium.dunossauro.live/aula_09.html

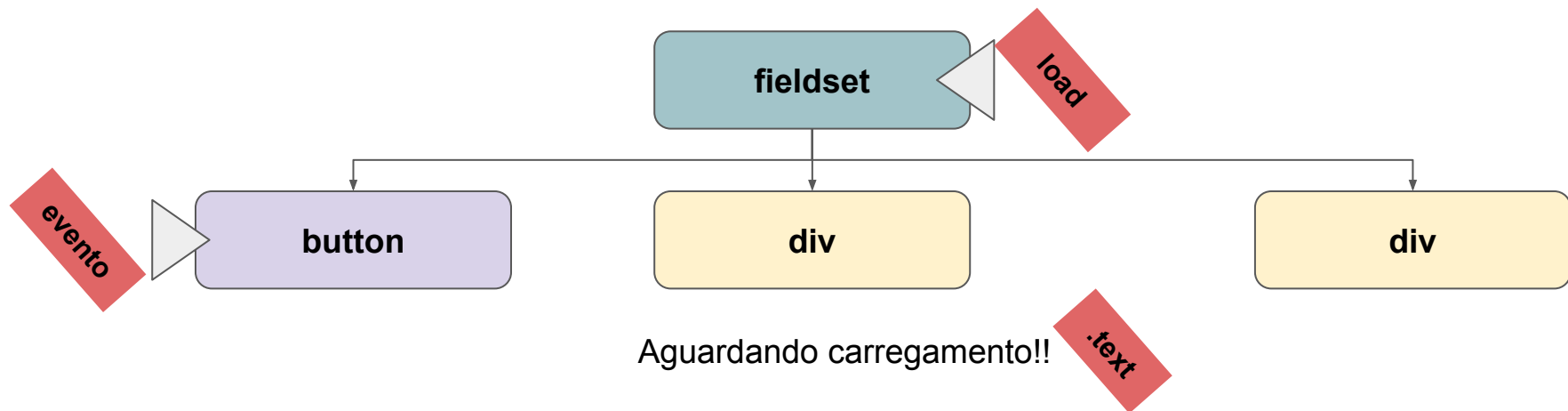


Página B [Antes do carregamento]



| attr | value |
|-------|-----------|
| class | btn-ghost |
| class | btn-error |
| class | unclick |

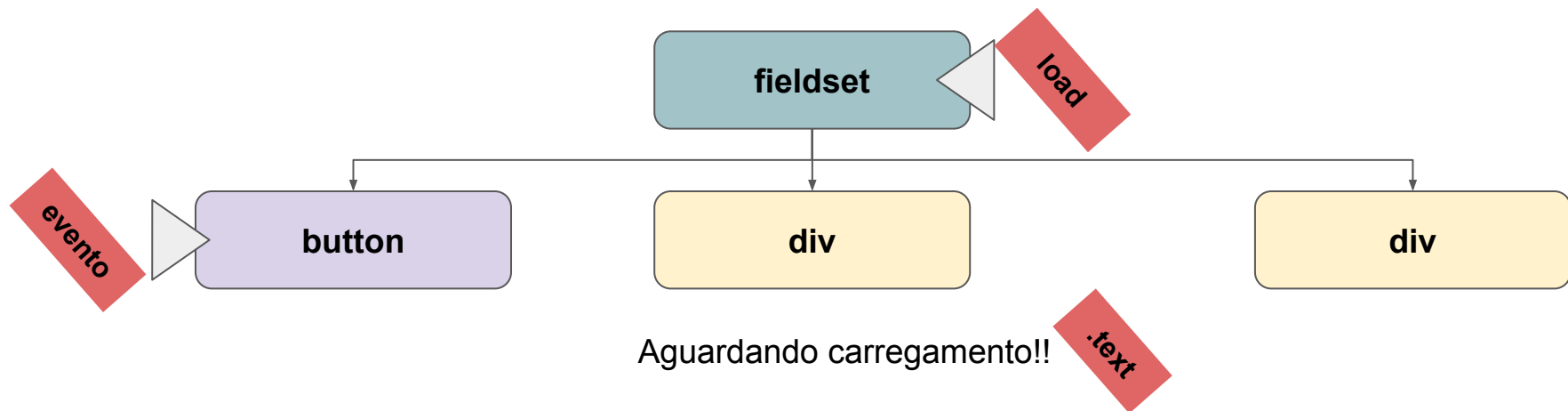
Página B [Antes do carregamento]



| attr | value |
|-------|-----------|
| class | btn-ghost |
| class | btn-error |
| class | unclick |

```
.unclick {  
  pointer-events: none  
}
```

Página B [Depois do carregamento]



| attr | value |
|-------|-------------|
| class | btn-ghost |
| class | btn-primary |

Waits

Aguardando carregamento!!

Barrinha top

attr

value

class

btn-ghost

class

btn-error

class

unclick

Waits

Barrinha top

attr

value

class

btn-ghost

class

btn-primary

Esperando elemento ativo

Embora seja possível ver o elemento, ele não tem as características que precisamos para clicar

Precisamos esperar o elemento estar ativo **is_enabled**

```
def esperar_ser_clicavel(element, web_driver):  
    print(f'esperando {element}...')  
    elements = web_driver.find_elements_by_css_selector(element)  
    if elements:  
        return elements[0].is_enabled()  
    return False
```



Voltando ao desafio inicial

https://selenium.dunossauro.live/aula_09.html



Exercícios

09: https://selenium.dunossauro.live/exercicio_09.html

10: https://selenium.dunossauro.live/exercicio_10.html

