

FABRÍCIO BARBOSA VIEGAS

# Automatize a web



SA  
COMP  
25



---

[www.fabriciobarbosa.com](http://www.fabriciobarbosa.com)

# 01 Recados Breves

# Está gostando da Sacomp?

Mostre para seus colegas como o evento está legal! Marque o Instagram da sacomp nas fotos que tirar do evento

[@sacomp\\_ufpel](#), caso publique algo deste minicurso marque também [@fabriciobarbosaviegas](#).



**Abandone  
o trivial,  
comece a  
programar  
paralelo!**



# 02 Roteiro Da Aula





# O que vamos aprender?

- 1 O que é o Selenium?
- 2 Como funciona uma pagina Web?
- 3 Criando seu primeiro Script
- 4 Automatizando o WhatsApp
- 5 Próximos passos

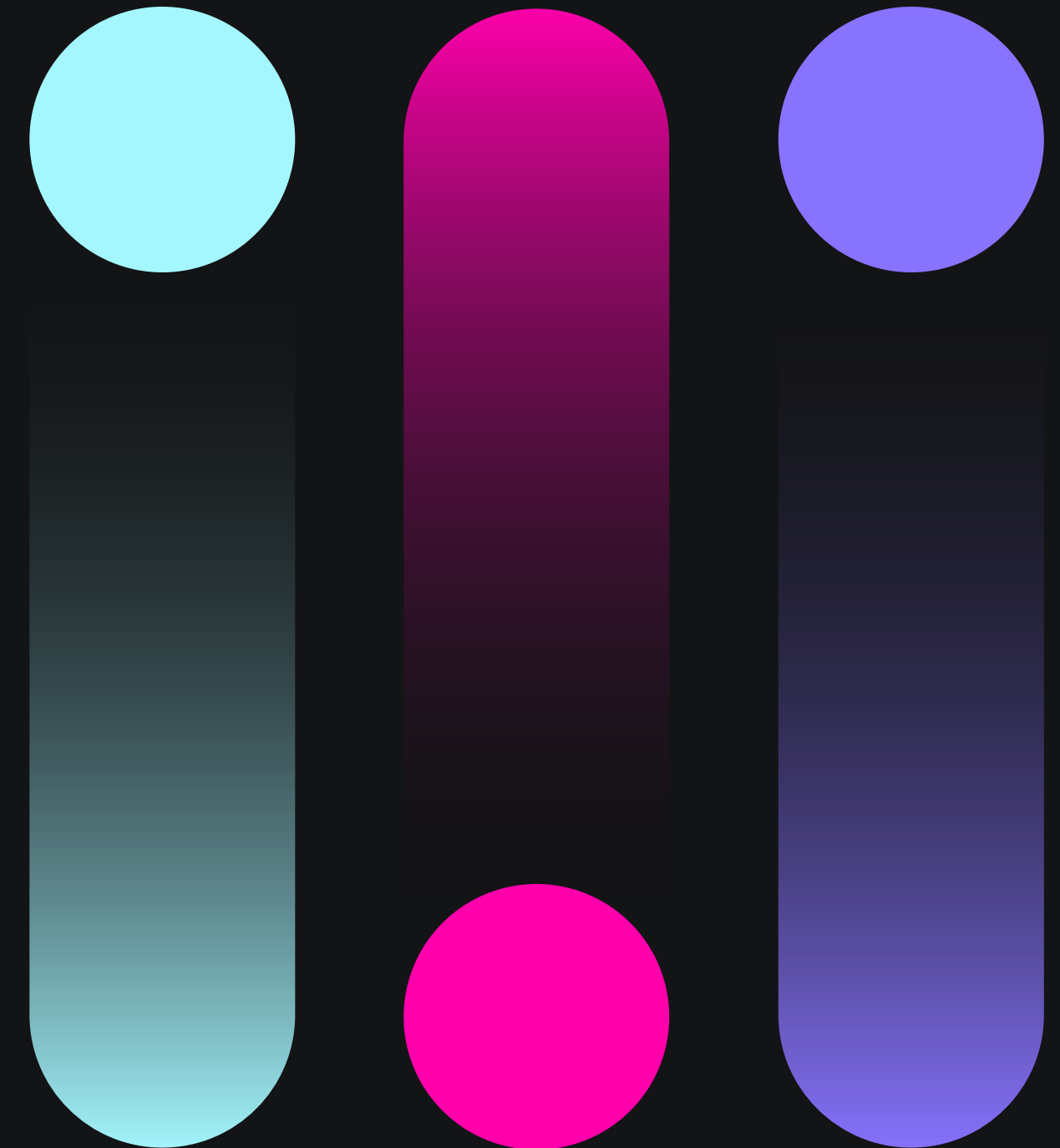


# 02 O que é Selenium?

# O que é Selenium?

- Framework para controle automático de navegadores
- Suporta Chrome, Firefox, Edge, Safari
- Usado para:
  - Testes automatizados
  - Web scraping
  - Robôs de navegação

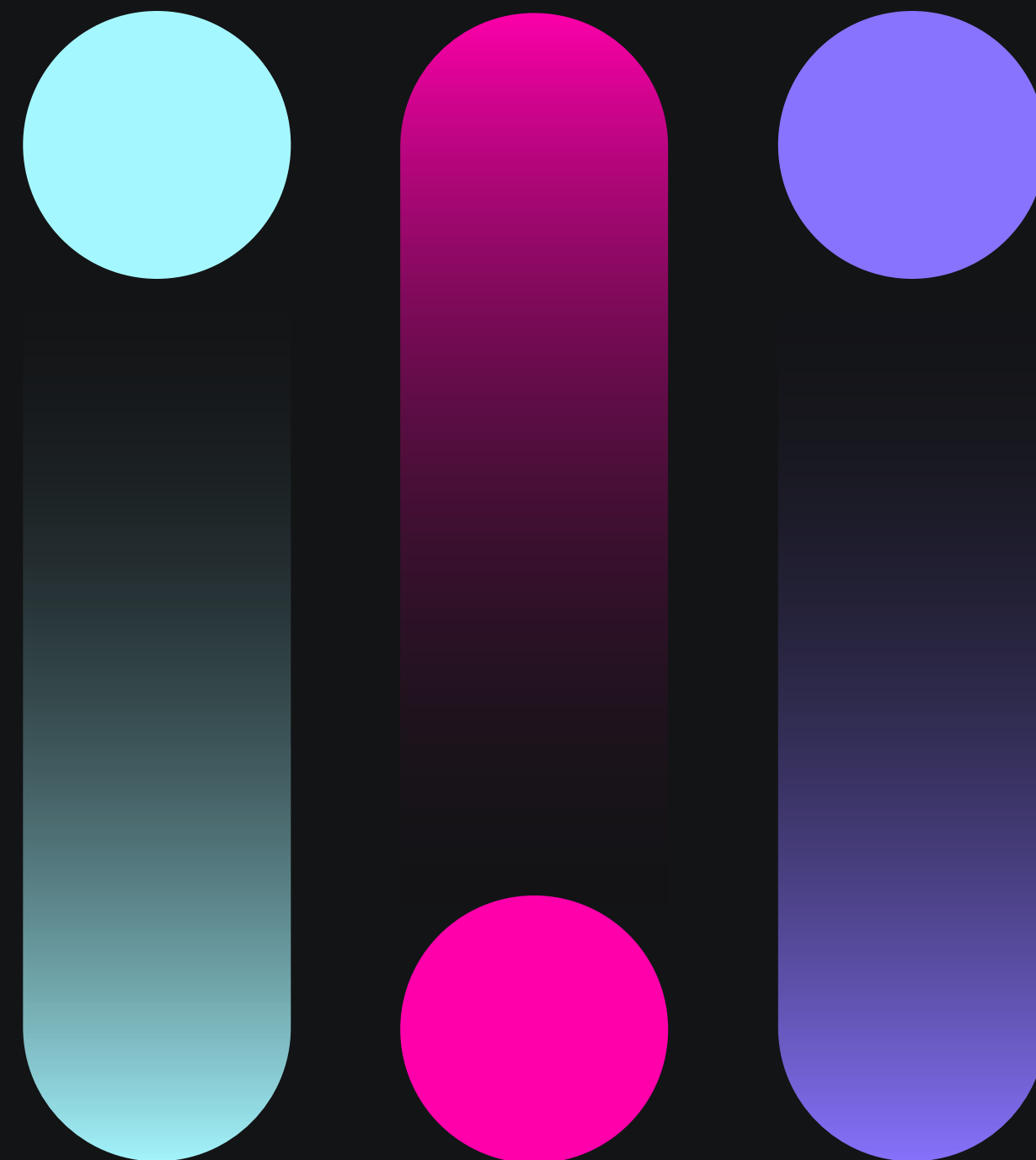
📌 Simula o comportamento humano no navegador





# Porque aprender automação?

Todos os dias executamos centenas de tarefas repetitivas sem pensar que ocupam nosso tempo. Se pudéssemos criar scripts que fizessem essas tarefas ao apertar um botão, poderíamos canalizar essa energia para coisas mais produtivas. O mesmo acontece com as empresas!

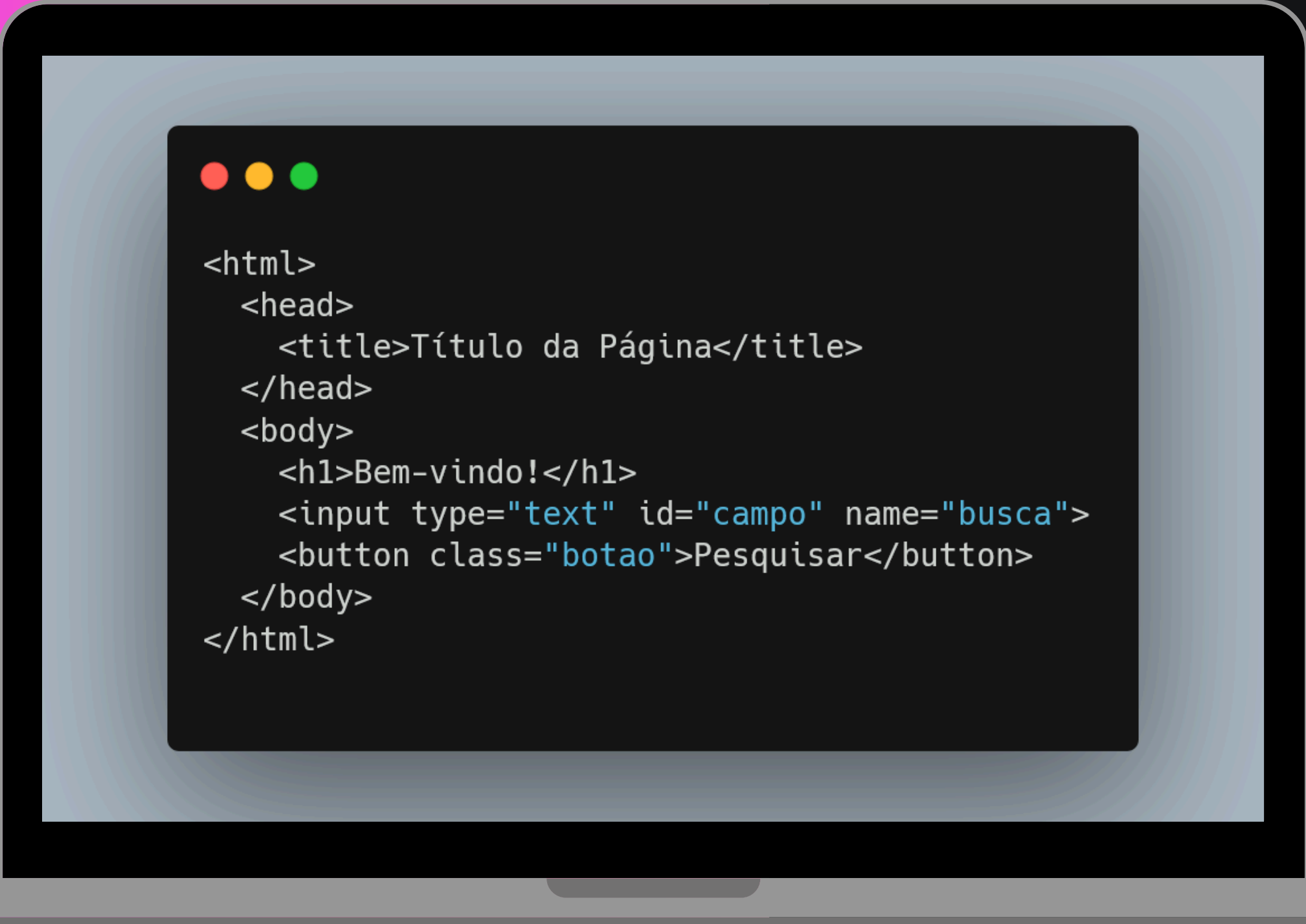


03

# Exemplos De aplicações



# 04 Como a Web Funciona?



```
<html>
  <head>
    <title>Título da Página</title>
  </head>
  <body>
    <h1>Bem-vindo!</h1>
    <input type="text" id="campo" name="busca">
    <button class="botao">Pesquisar</button>
  </body>
</html>
```

# Como é a estrutura de uma página?

Toda página web é composta por HTML (HyperText Markup Language), que organiza o conteúdo da seguinte forma.

O Selenium interage com esses elementos simulando ações humanas: clicar, digitar, rolar a página etc.

# Elementos HTML comuns

Tag HTML	Função	Exemplo
<code>&lt;input&gt;</code>	Campo de texto	<code>&lt;input type="text" name="email"&gt;</code>
<code>&lt;button&gt;</code>	Botão clicável	<code>&lt;button&gt;Enviar&lt;/button&gt;</code>
<code>&lt;a&gt;</code>	Link	<code>&lt;a href="..."&gt;Clique aqui&lt;/a&gt;</code>
<code>&lt;div&gt;</code>	Contêiner genérico	<code>&lt;div class="card"&gt;...&lt;/div&gt;</code>
<code>&lt;form&gt;</code>	Agrupa campos de formulário	<code>&lt;form action="..."&gt;...&lt;/form&gt;</code>

<https://automationintesting.com/selenium/testpage/>





# 05 Criando Seu primeiro script

# Instalando o Selenium

Vamos começar instalando o selenium através do pip.

Logo em seguida vamos instalar o webdriver-manager, ele que irá nos ajudar a gerenciar os nossos navegadores.



```
pip install selenium
```

```
pip install webdriver-manager
```



```
import time
from selenium import webdriver

driver = webdriver.Chrome()

driver.get("https://fabricioarbosa.com")
time.sleep(3)
driver.quit()
```

# Seu primeiro script!

Começamos com algo simples, um script que acessa uma página na internet a exibe por alguns segundos e depois fecha.



Encontrando o titulo da pagina

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

driver = webdriver.Chrome()

driver.get("https://fabriciobarbosa.com")
time.sleep(1)

titulo = driver.find_element(By.TAG_NAME, "h1")

print(titulo)
print(titulo.text)

driver.quit()
```

# Procurando elementos na pagina

O método `find_element()` nos permite encontrar qualquer elemento visível em um página HTML buscando pelo atributo desejado.

```
Interagindo com uma pagina

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

driver = webdriver.Chrome()

driver.get("https://automationintesting.com/selenium/testpage/")

firstName = driver.find_element(By.ID, "firstname")
firstName.send_keys("Fabrício")
time.sleep(1)

surname = driver.find_element(By.ID, "surname")
surname.send_keys("Barbosa")
time.sleep(1)

btn = driver.find_element(By.ID, "submitbutton")
btn.click()
time.sleep(1)

driver.quit()
```

# Interagindo com a pagina

Existem muitas formas de interagir com uma página web, mas podemos generalizar nossas ações para duas formas principais, clicar em botões (método `click()`) e preencher inputs de texto (método `send_keys()`).



The background features two overlapping teal circles on a dark navy blue background. The larger circle is on the left, and a smaller one is on the right, partially overlapping the first.

# 05 Automatizando

- WhatsApp

# O que vamos fazer?

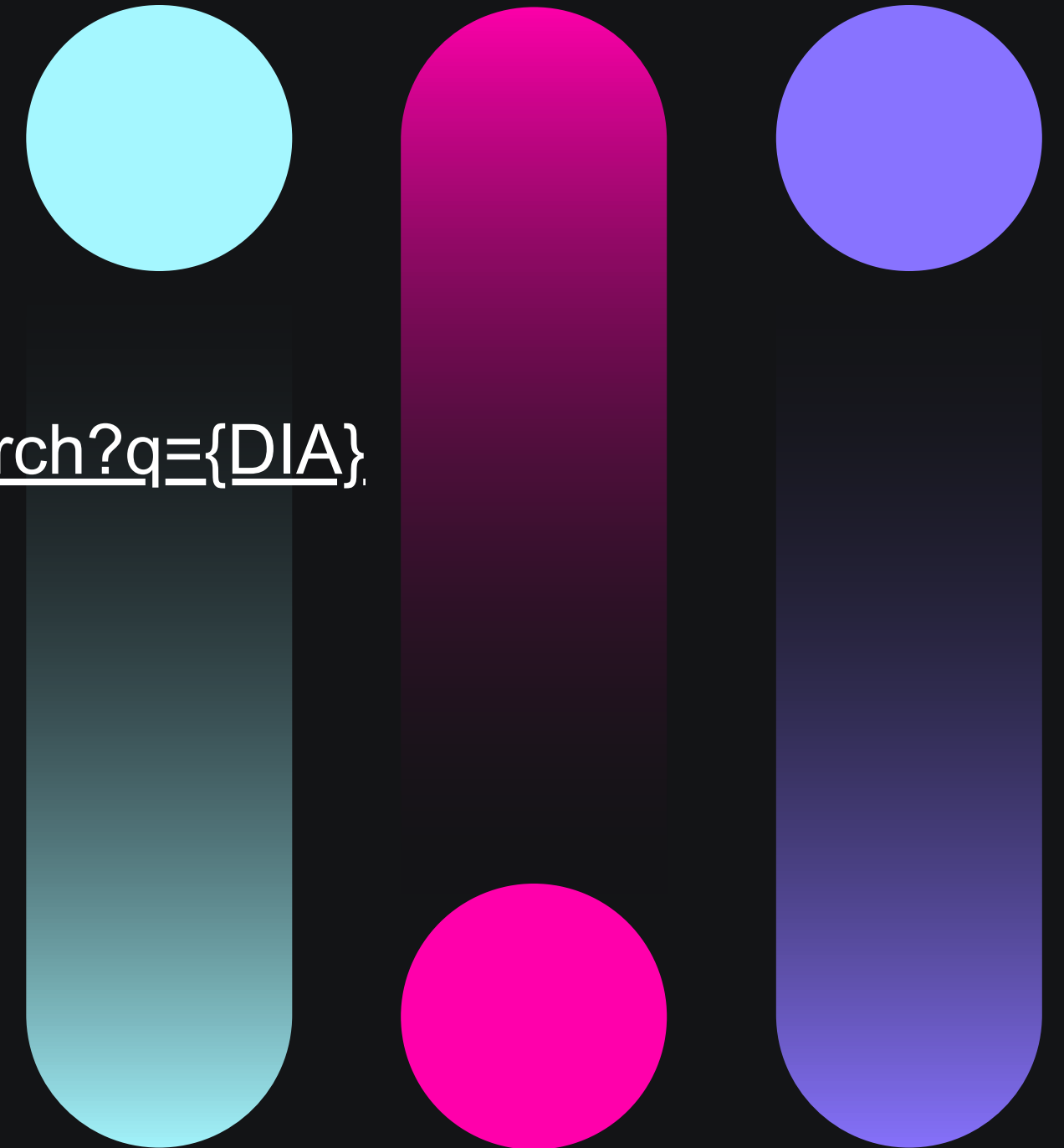


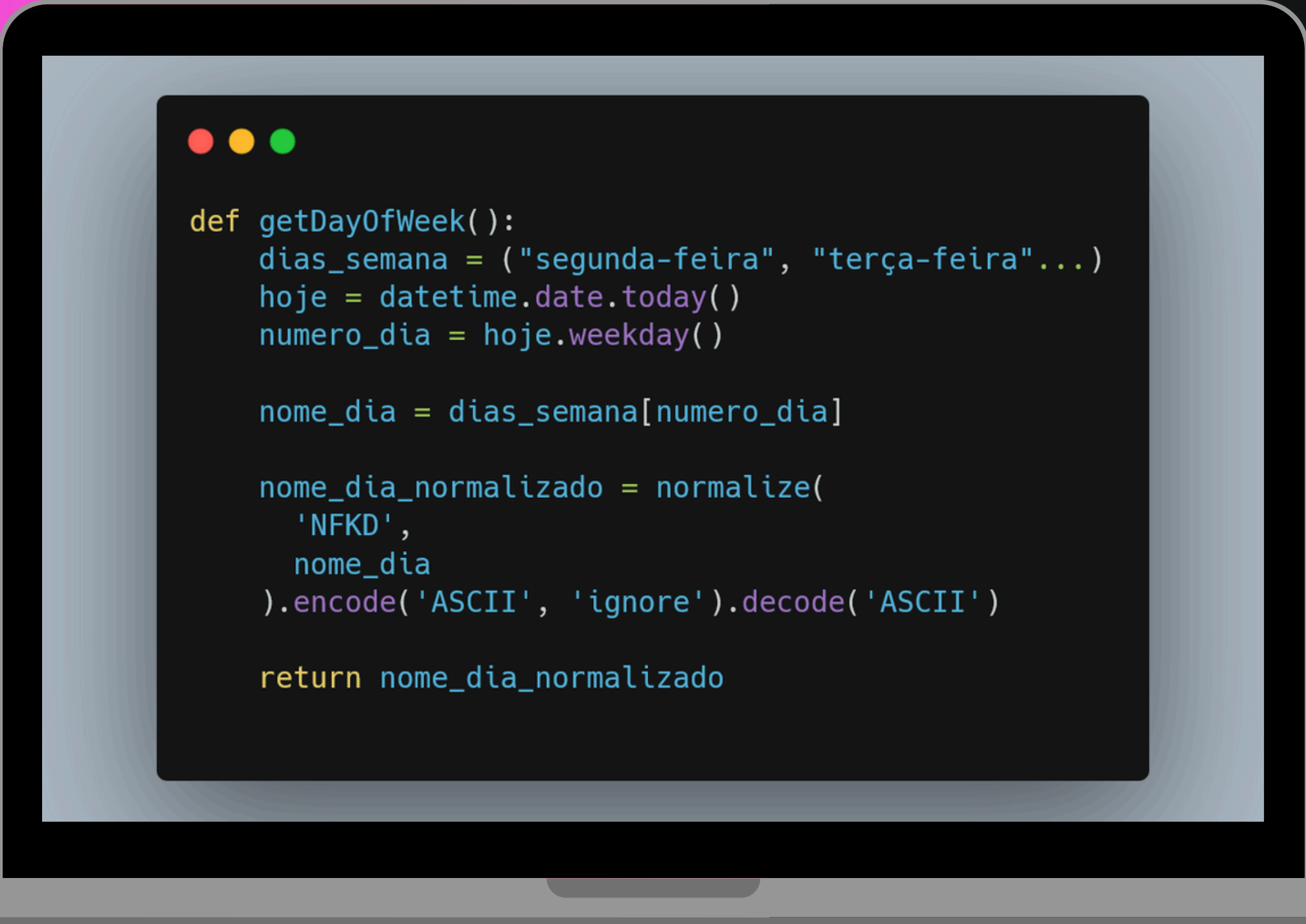
Para nosso projeto prático vamos realizar o sonho de sua vó, suas tias e até mesmo sua mãe.

Vamos criar um script que todos os dias baixe imagens inéditas de bom dia na internet e envie essas imagens pelo WhatsApp!

# Algoritmo

- Descubre o dia da semana
- Abre o site: <https://www.imagensdiarias.com.br/search?q={DIA}>
- Baixa uma imagem aleatória
- Abre o WhatsApp
- Espera o usuário logar
- Escolhe o primeiro contato da lista
- Envia a imagem!
- Encerra





```
def getDayOfWeek():  
    dias_semana = ("segunda-feira", "terça-feira"...)  
    hoje = datetime.date.today()  
    numero_dia = hoje.weekday()  
  
    nome_dia = dias_semana[numero_dia]  
  
    nome_dia_normalizado = normalize(  
        'NFKD',  
        nome_dia  
    ).encode('ASCII', 'ignore').decode('ASCII')  
  
    return nome_dia_normalizado
```

# Descobrimos o dia da semana

Vamos começar implementando a funções que descobre o dia da semana para buscar o bom dia personalizado.

Pegar imagem aleatória

```
def getImage(driver):  
    driver.get(f'https://www.imagensdiarias.com.br/search?q={getDayOfWeek()}')  
    imagens = driver.find_elements("css selector", ".post img")  
    lenImages = len(imagens)  
  
    if lenImages > 0:  
        randomImage = random.choice(imagens)  
        randomImage = randomImage.get_attribute('src')  
        return randomImage  
    else:  
        print("Sem imagens.")  
  
    return None
```

# Obtendo a imagem

Agora vamos acessar o site e escolher uma imagem aleatória.





Pegar imagem aleatória

```
def downloadImage(imageUrl):  
    response = requests.get(imageUrl)  
    if response.status_code == 200:  
        with open('image.jpg', 'wb') as file:  
            file.write(response.content)  
        print("Image downloaded successfully.")  
    else:  
        print("Failed to download image.")
```

# Baixando a imagem

Pegar imagem aleatória

```
def login(driver):  
    driver.get('https://web.whatsapp.com/')  
  
    # Espera até que o elemento canvas apareça na página  
    canvas = WebDriverWait(driver, 10).until(  
        EC.presence_of_element_located((By.CSS_SELECTOR, "canvas"))  
    )  
    #espera até que a pagina seja modificada, se for modificado segue  
    driver.implicitly_wait(10)  
    if canvas.is_displayed():  
        print("Por Favor escaneie o QR Code com seu celular...")  
        WebDriverWait(driver, 30).until(  
            EC.presence_of_element_located(  
                (By.CSS_SELECTOR, ".xk50ysn.x14ug900.x1603h9y.xgif2c7")  
            )  
        )  
        print("Login realizado com sucesso.")
```

# Logando no WhatsApp


Vamos esperar até que o usuário leia o QR Code e a página seja modificada usando métodos avançados.



```
def disablePopUp(driver):  
    time.sleep(3)  
  
    actions = ActionChains(driver)  
    actions.send_keys(Keys.ESCAPE).perform( )
```

# Fechando Pop Up

Vamos fechar o popup que é aberto quando o usuário loga pela primeira vez, usando o teclado.

A laptop screen with a black bezel and a light gray base. The screen shows a code editor with a dark background and light blue, green, and red syntax highlighting. The code is a Python function named 'openContact' that uses Selenium WebDriver to wait for an element and click it. The function takes a 'driver' argument and returns 'contato'. The code is as follows:

```
def openContact(driver):  
    contato = WebDriverWait(driver, 20).until(  
        EC.presence_of_element_located(  
            (By.CSS_SELECTOR, ".x10l6tqk.xh8yej3.x1g42fcv")  
        )  
    )  
    contato.click()
```

# Abrindo contato

Vamos localizar o elemento que representa o primeira contato e clicar.

```
def sendFile(driver):
    attach_button = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located(
            (By.CSS_SELECTOR, "span[data-icon='plus-rounded']")
        )
    )
    attach_button.click()


    image_path = os.path.abspath("image.jpg")

    file_input = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located(
            (By.CSS_SELECTOR, "input[type='file']")
        )
    )
    file_input.send_keys(image_path)
    time.sleep(2)

    send_button = WebDriverWait(driver, 10).until(
        EC.element_to_be_clickable(
            (By.CSS_SELECTOR, "span[data-icon='wds-ic-send-filled']")
        )
    )
    send_button.click()
```

# Enviando imagem

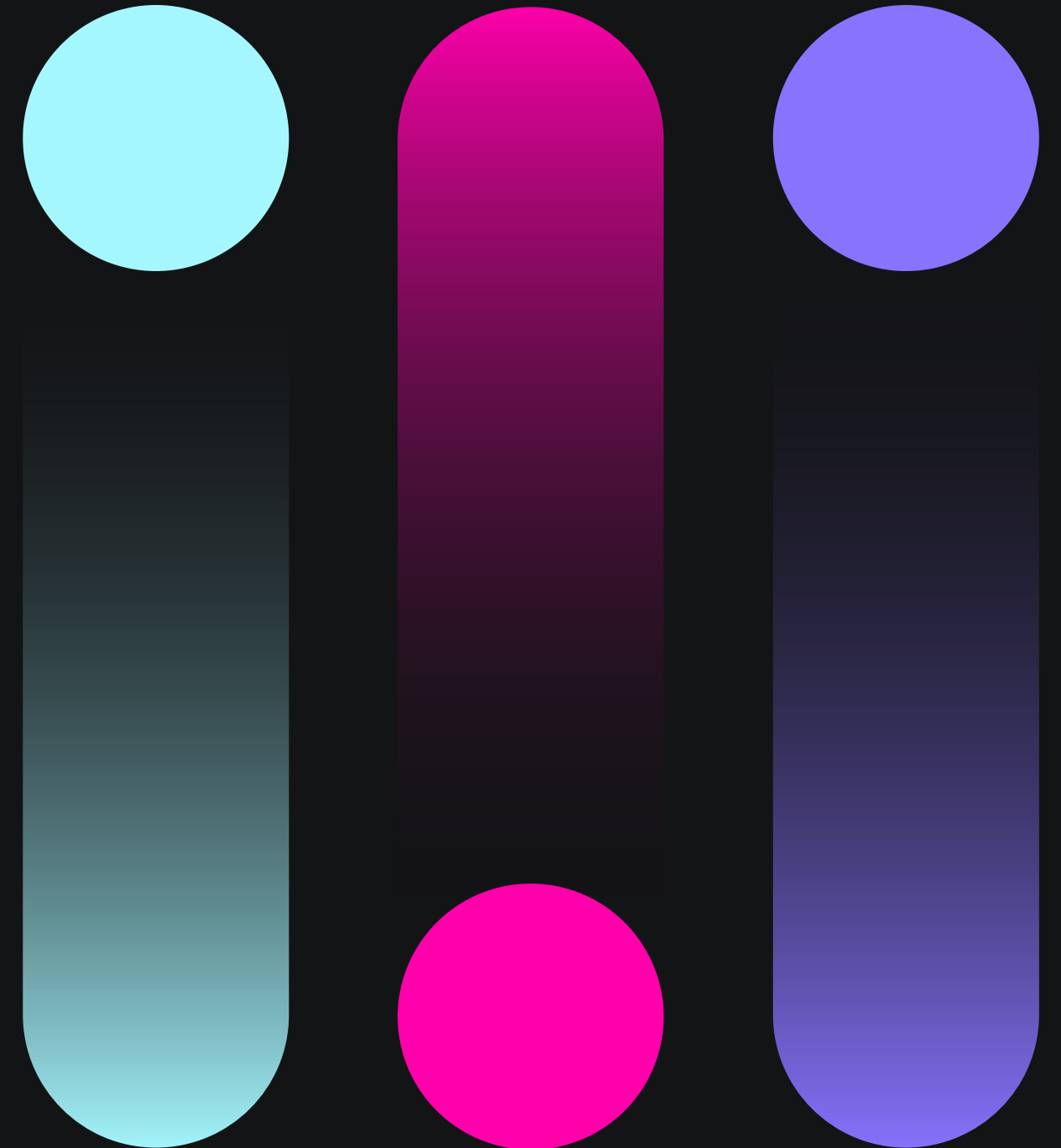




# 06 Limitações E cuidados

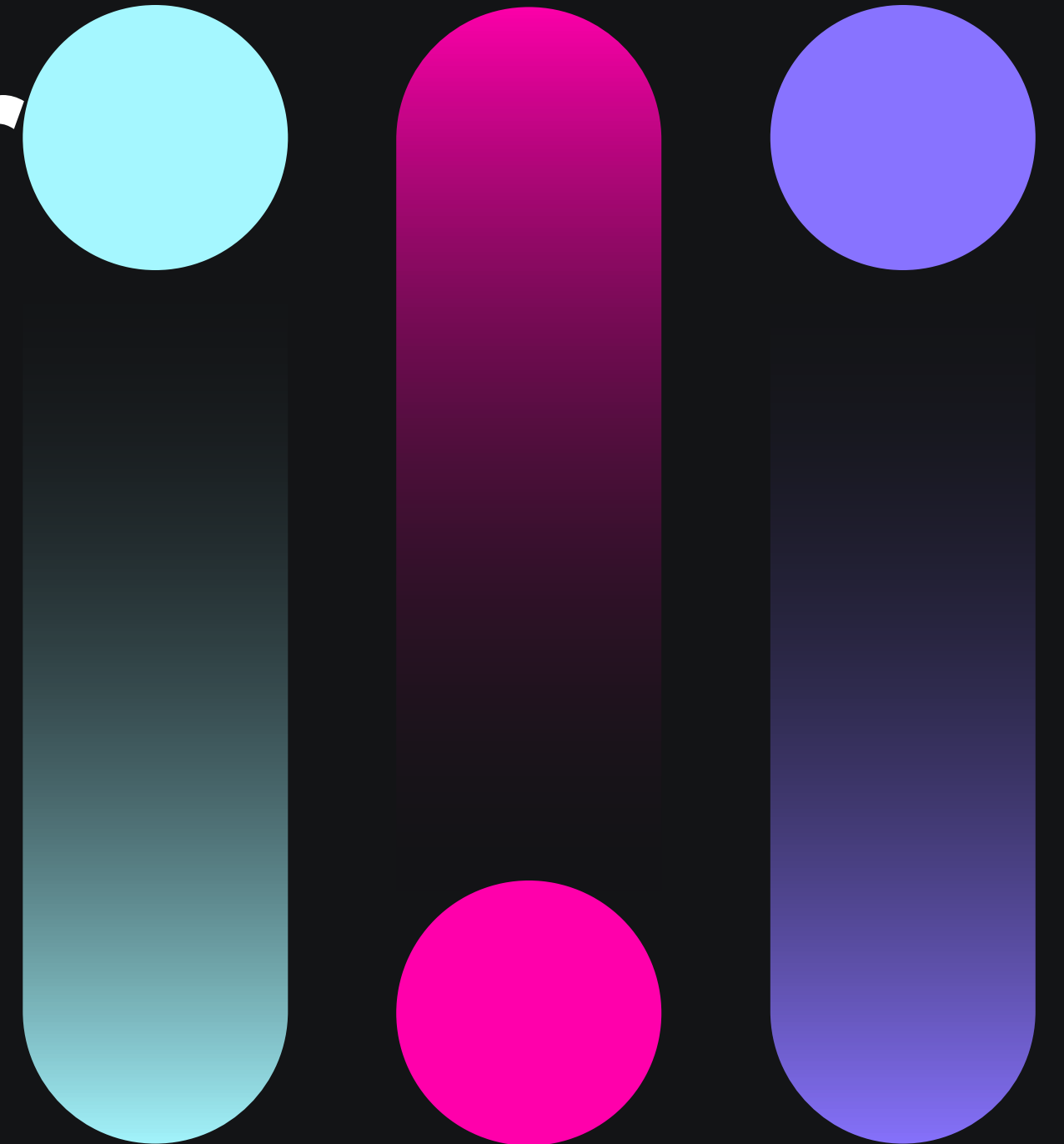
# Limitações

- Sites que detectam bots;
- Conteúdo dinâmico com JavaScript pesado
- Restrições legais (consulte os termos de uso!)
- CAPTCHA, 2FA e anti-automação



# Quando não usar Selenium?

- Quando o serviço já fornece uma API oficial;
- Sites com sistemas anti-bots complexos;
- Sistemas com punições para ferramentas automatizadas.



# Contato



[www.fabriciobarbosa.com](http://www.fabriciobarbosa.com)



[fbviegas@inf.ufpel.edu.br](mailto:fbviegas@inf.ufpel.edu.br)

Obrigado!

