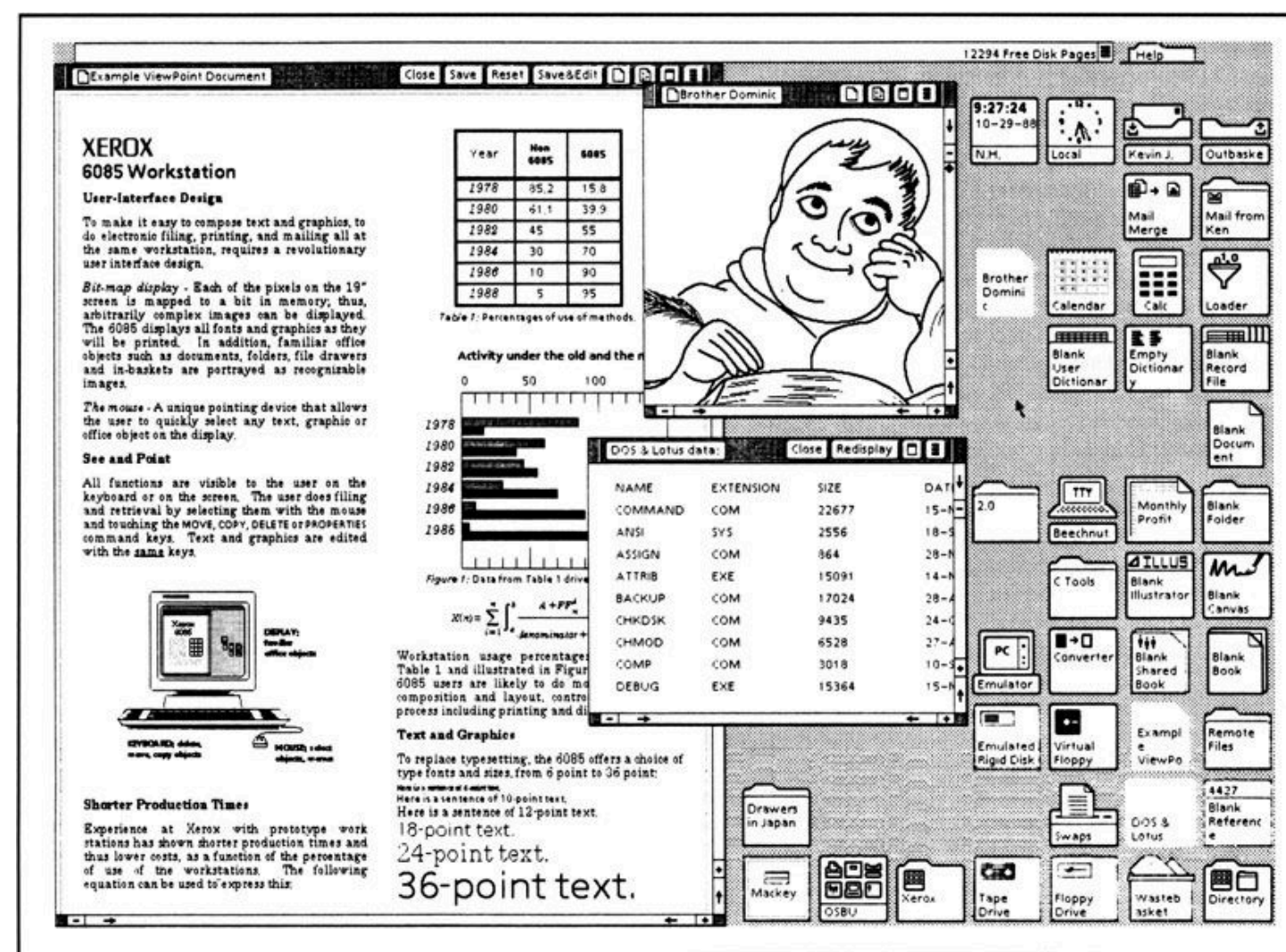


# Automação de GUI com Python

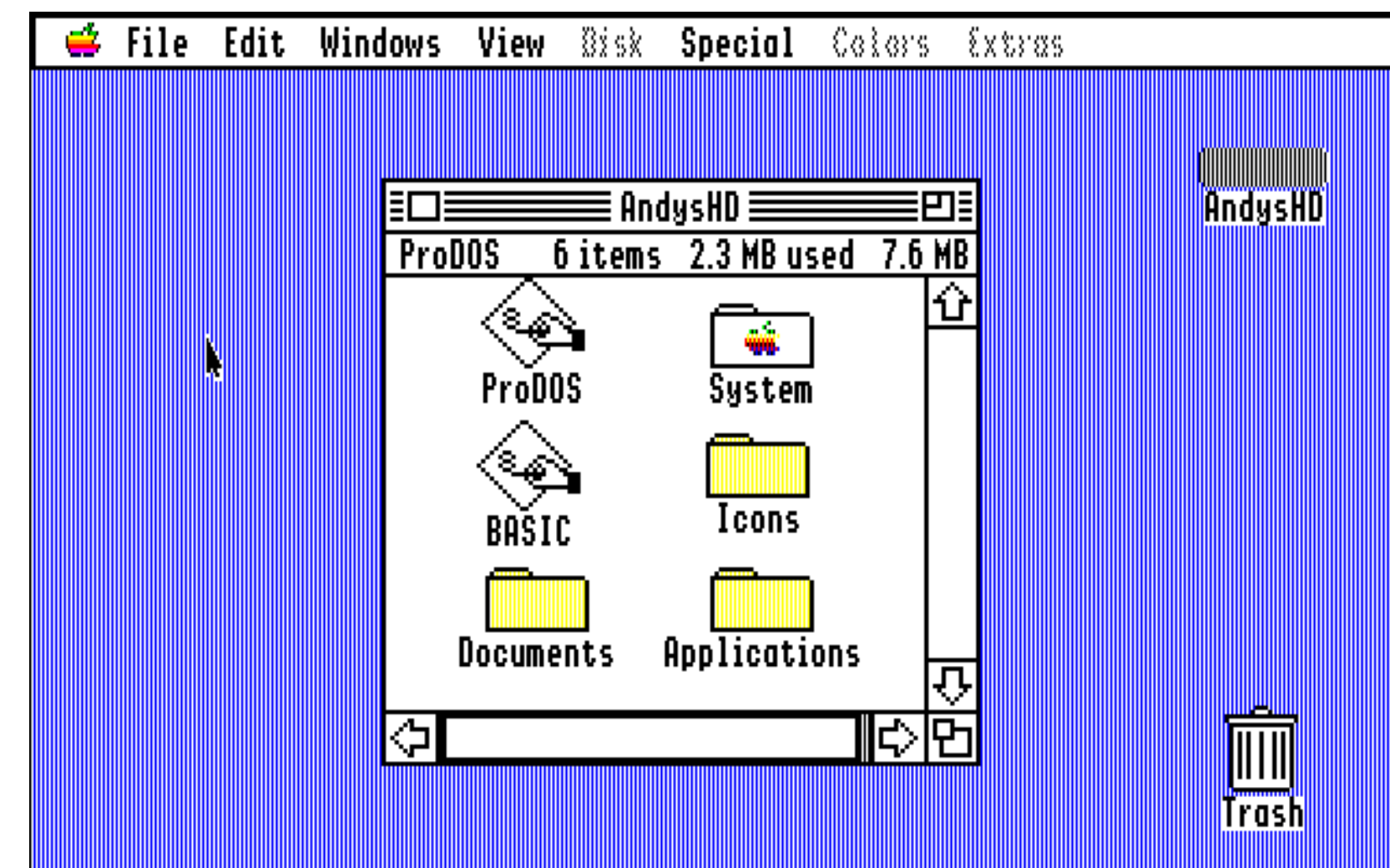
Gabriel Casemiro

# Graphical User Interface

Interface Gráfica do Usuário, em português, é em um modelo de interface do utilizador que permite a interação com os dispositivos digitais através de elementos gráficos.



GUI Xerox 1973



GUI Apple 1986

# Software antigo

## Produtos

- 1 - Incluir cadastro
- 2 - Atualizar cadastro
- 3 - Remover cadastro

## Relatorios

- 4 - Todos os produtos
- 5 - Produto especifico

## Outros

- 0 - Voltar ao menu principal

Digite uma opc do menu:

# Uma GUI Simples

## New Employee Registration

Male

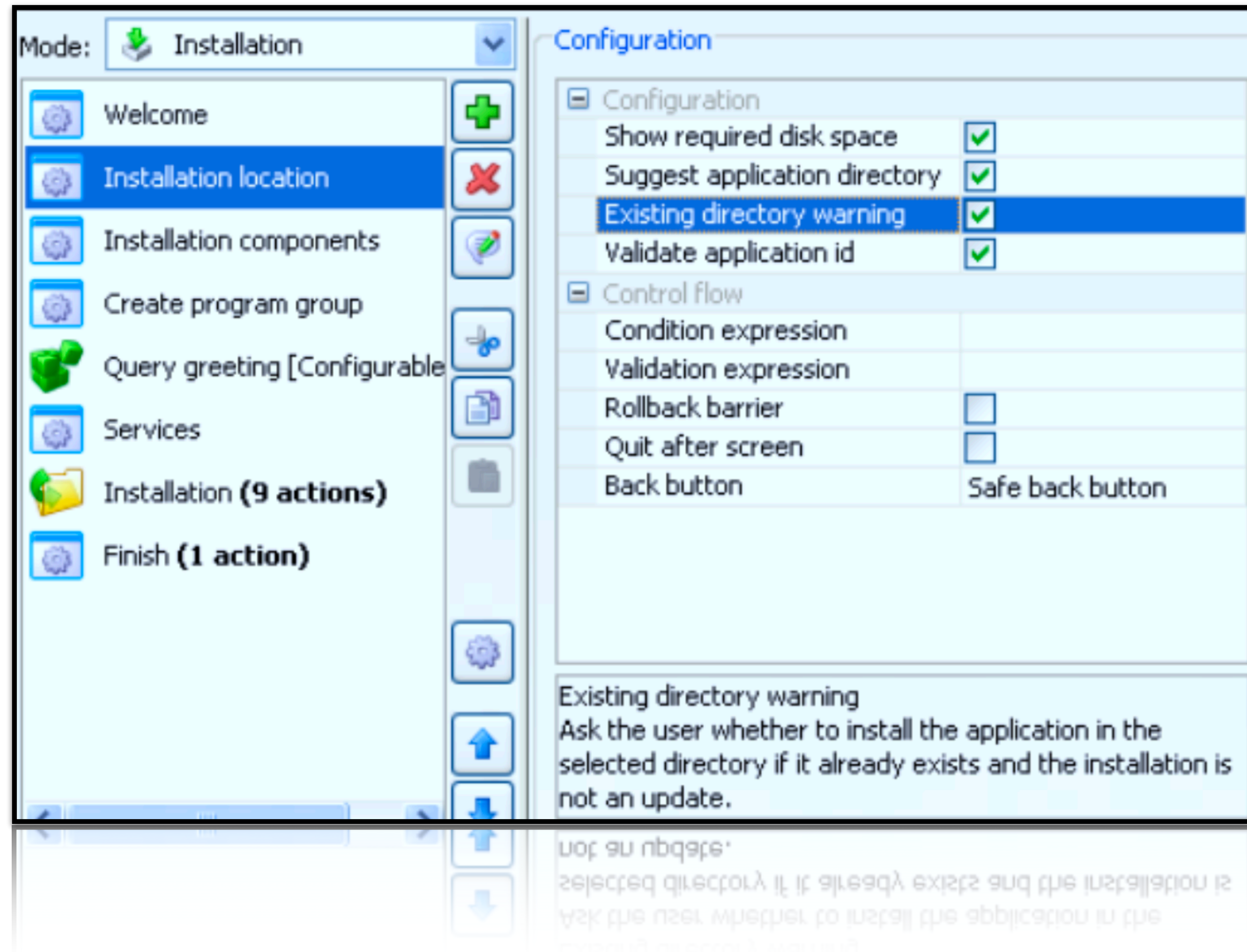
▼

Register

Register



# Uma GUI Simples



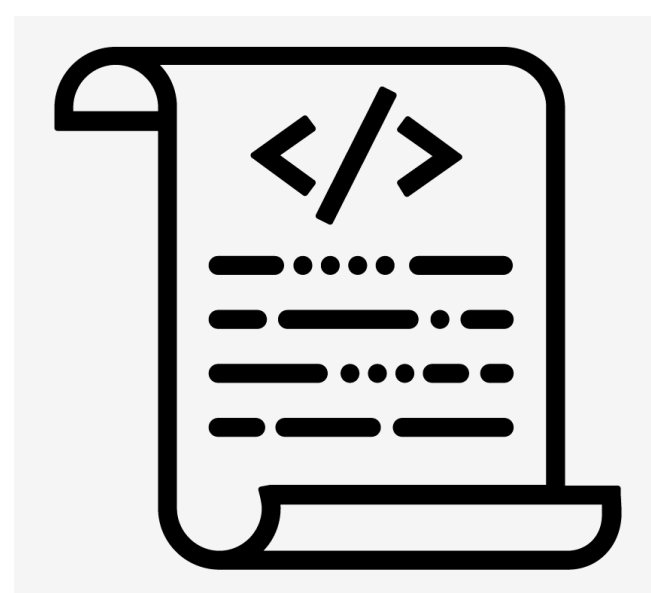


**PyAutoGui**

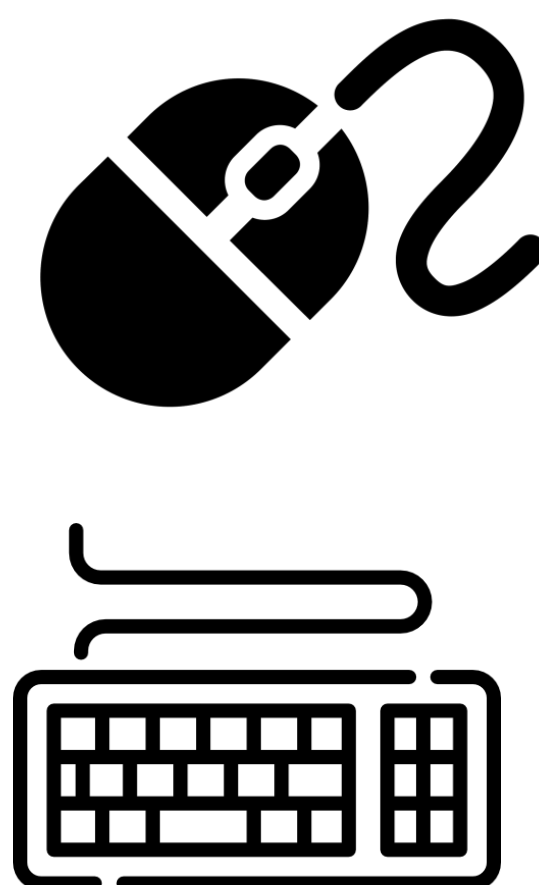
# O que é automação GUI?

A automação de GUI é o processo de simulação de ações de mouse e teclado em janelas e controles.

A maioria das ferramentas de automação é baseada nas coordenadas dos controles ou no texto que eles contêm, mas isso nem sempre é confiável



**Script de Automação**



**Simula ações do  
mouse e teclado**



**Controla programas  
De computador**

# Aplicações da Automação GUI

- ✓ Testes automatizados
- ✓ Entrada de dados automatizada
- ✓ Permite que os usuários automatizem facilmente tarefas tediosas e repetitivas ou tarefas complexas
- ✓ Integração de aplicativos e migração de conteúdo
- ✓ Lidar com aplicativos legados que não oferecem outras APIs

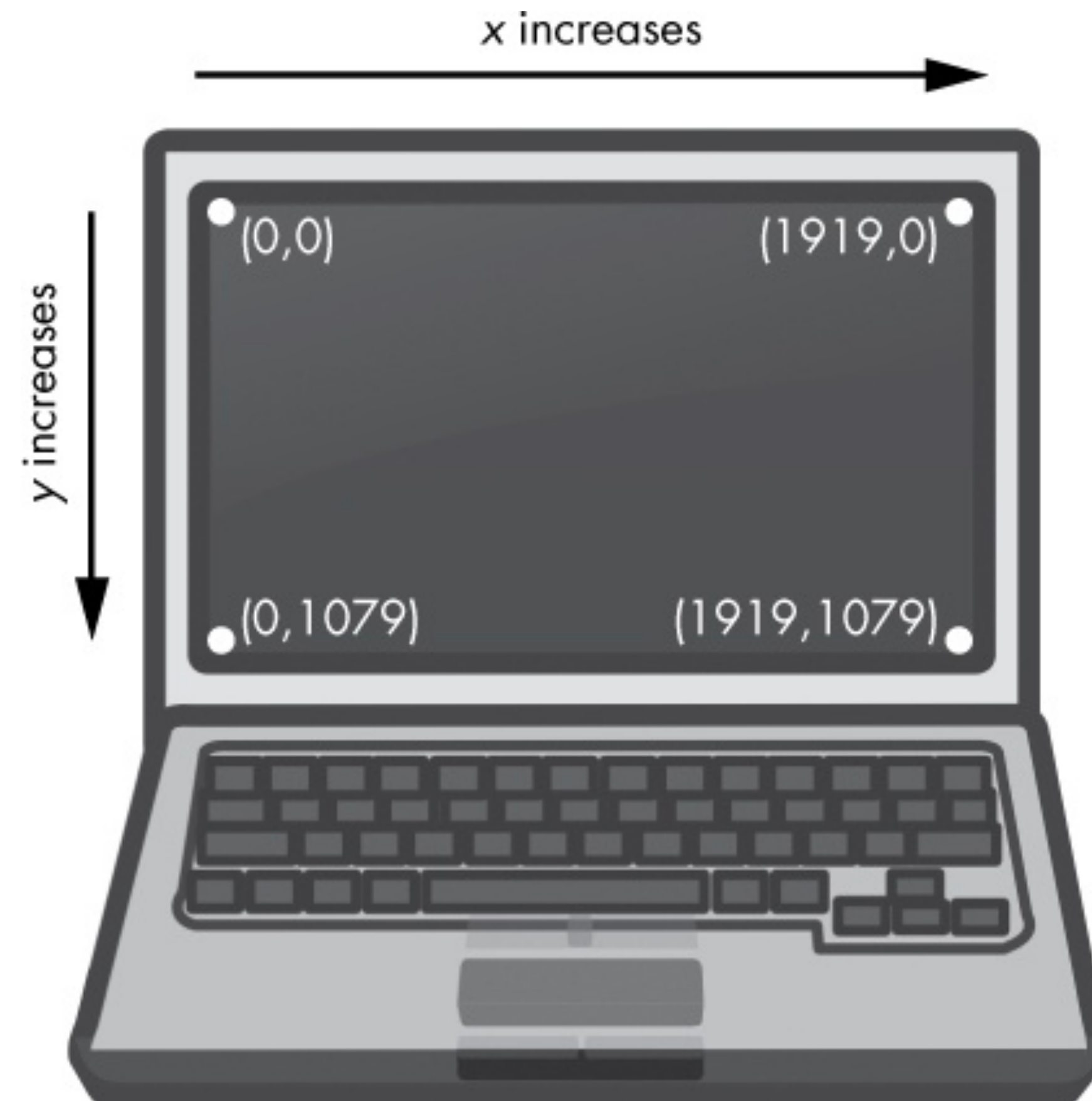


# Ambiente de estudo

- 1) Criar uma Windows VM na Cloud Pública (Para usuários que não são Windows)
- 2) Instalar o Python3
- 3) Instalar Pyautogui

# Calculando coordenadas

1919x1079



# Funções do Mouse

## **pyautogui.size()**

Mostra o tamanho (resolução) da tela

## **pyautogui.position()**

Mostra as coordenadas x e y do Mouse

## **pyautogui.displayMousePosition()**

Mostra em tempo real as coordenadas x e y do Mouse

## **pyautogui.moveTo(x,y,duration)**

Move o mouse para a coordenadas específicas

# Funções do Mouse

```
pyautogui.click(x=moveToX,  
                 y=moveToY,  
                 clicks=num_of_clicks,  
                 interval=secs_between_clicks,  
                 button='left')
```

Realiza um click com o botão do mouse

**Outros métodos similares: rightClick(), middleClick(),  
doubleClick() e tripleClick()**



# Funções do Mouse

## **pyautogui.dragTo(x,y)**

Método utilizado para mover e arrastar o ponteiro do mouse para uma coordenada específica

Parecido com **moveTo** porém ele arrasta até o local.

### **Exemplo:**

`dragTo(100,500)` # *Clica e arrasta o mouse para o ponto (100,500)*

`dragTo(None, 1000)` # *Clica e arrasta o mouse apenas para o 1000<sup>th</sup> pixel do eixo y*

`dragTo(100,200, duration=3)` # *Clica e arrasta o mouse para (100,200) em 3 segundos*

# Funções do Mouse

**`pyautogui.scroll(amount_to_scroll, x=moveToX, y=moveToY)`**

Essa função permite simular a rotação da bolinha do mouse.

**Exemplo:**

```
pyautogui.scroll(10) # scroll up 10 clicks
```

```
pyautogui.scroll(-10) # scroll down 10 clicks
```

```
pyautogui.scroll(10, x=100, y=100) # move mouse to 100, 200, then scroll up 10 clicks
```

# Funções do Mouse

## **`pyautogui.dragRel(x,y)`**

Clica e arrasta o mouse para uma posição relativa a posição do mouse atual

## **`pyautogui.moveRel(x,y)`**

Mova o cursor do mouse para uma posição relativa a posição do mouse atual

# Configurações

## **pyautogui.FAILSAFE = False**

Desabilita a exceção `FailSafeException` se o cursor do mouse atingir a coordenada (0,0)

## **pyautogui.PAUSE = 2.5**

Adiciona uma delay de 2.5 segundos (configurável) a cada ação do mouse.

Isso nos dá algum tempo para levar o mouse até a coordenada (0,0) e finalizar o programa.



# Screenshots

**`pyautogui.screenshot('my_screenshot.png')`**

Grava uma foto da tela e salva no arquivo my\_screenshot.png

**`pyautogui.screenshot(region=(0,0, 300, 400))`**

Se você não quer a tela toda, você pode passar uma região para a screenshot ser tirada.

São quatro inteiros que delimitam a esquerda, topo, largura e altura.

# Screenshots

A funcionalidade de captura de tela requer o módulo **Pillow**

- A Python Imaging Library adiciona recursos de processamento de imagem ao seu interpretador Python.
- Esta biblioteca oferece amplo suporte a formatos de arquivo, uma representação interna eficiente e poderosos recursos de processamento de imagem.

Comando:

**pip install pillow**

# Localizando imagens

Como funciona o processo de identificação da elementos na imagem:

- 1) Tire uma foto da tela toda
- 2) Procure a pequena imagem que passamos para a função e localize-a na captura de tela
- 3) Retorna a posição imagem

# Não está obtendo correspondência exata?

## **Confidence (Parâmetro Opcional)**

Temos que instalar o módulo OpenCV para usar este parâmetro.

Especifica a precisão com que a função deve localizar a imagem na tela.

Isso é útil caso a função não consiga localizar uma imagem devido a diferenças insignificantes de pixel:

## **Exemplo:**

```
pyautogui.locateOnScreen('test.png', confidence=0.8)
```



# Melhorando a performance

## **Grayscale (parâmetro opcional)**

Isso desatura a cor das imagens e capturas de tela, acelerando a localização, mas potencialmente causando correspondências falso-positivas.

Isso acelera o processo de localização da imagem, mas pode causar correspondências falso-positivas.

## **Exemplo:**

```
pyautogui.locateOnScreen('test.png', grayscale=True)
```

# Funções de localização

<b>locateOnScreen(image)</b>	Retorna a coordenada (esquerda, superior, largura, altura) da primeira instância encontrada da imagem na tela.
<b>locateCenterOnScreen (image)</b>	Retorna (x, y) as coordenadas do centro da primeira instância encontrada da imagem na tela
<b>locateAllOnScreen(image)</b>	Retorna um gerador que produz tuplas (esquerda, superior, largura, altura) para onde a imagem é encontrada na tela.
<b>locate(needleImage, haystackImage)</b>	Retorna a coordenada (esquerda, superior, largura, altura) da primeira instância encontrada de needleImage em haystackImage.
<b>locateAll(needleImage, haystackImage)</b>	Returns a generator that yields (left, top, width, height) tuples for where needleImage is found in haystackImage.

Uma exceção *ImageNotFoundException* é gerada se a imagem especificada não estiver presente

# Funções de teclado

## **typewrite()**

Aceita um parâmetro de string e digita-os em caracteres

## **Example:**

```
pyautogui.typewrite('I just love this course')
```

```
pyautogui.typewrite('I just love this course', interval=1)
```

Intervalo é um parâmetro opcional para adicionar atraso entre a digitação de 2 caracteres

# Funções de teclado

## **press()**

Esta função aceita uma string e pressiona a tecla correspondente mapeada para `pyautogui.KEYBOARD_KEYS`

## **Example:**

```
pyautogui.press('enter') # press the Enter key  
pyautogui.press('f5') # press the F1 key  
pyautogui.press('del') # press the left arrow key
```



# Funções de teclado

**Windows Shortcut:** Ctrl + Alt + Tab

Usando keyDown e keyUp

# Pressionando os botões

```
pyautogui.keyDown('ctrl')
```

```
pyautogui.keyDown('alt')
```

```
pyautogui.keyDown('tab')
```

# Soltando os botões na ordem inversa

```
pyautogui.keyUp('tab')
```

```
pyautogui.keyUp('alt')
```

```
pyautogui.keyUp('ctrl')
```

OU

**Windows Shortcut:** Ctrl + Alt + Tab

Usando HotKey

```
pyautogui.hotkey("ctrl", "alt", "tab")
```

# Processo de integração de funcionários

## **Requerimento:**

Automatize o processo de integração do funcionário para o seu organização. O RH envia a você um arquivo CSV contendo o informações de novos associados. Você tem que atualmente insira essas informações neste formulário manualmente.

Automatize esta GUI.

## **Suposições:**

Não há como acessar o banco de dados diretamente.

A página não está permitindo que o selenium a controle.

# Web Scraping

- ✓ 90% dos dados mundiais foram criados nos últimos dois anos
- ✓ Mais de 2,5 quintilhões de dados são criados todos os dias
- ✓ O número de usuários de internet em 2018 é de 4,021 bilhões, um aumento de 7% em relação ao ano anterior
- ✓ Quase 1 milhão de pessoas começaram a usar as mídias sociais pela primeira vez todos os dias durante o ano de 2018 – o que equivale a mais de 11 novos usuários a cada segundo.

# Web Scrapping

Existem várias maneiras de extrair informações da web:

- ✓ Copiar/Colar Manual

- ✓ Uso de API para extrair as informações e armazená-las no formato desejado

- ✓ Web Scrapping

O uso de APIs é provavelmente a melhor maneira de extrair dados de um site. Mas nem todos os sites fornecem APIs.

# Web Scraping

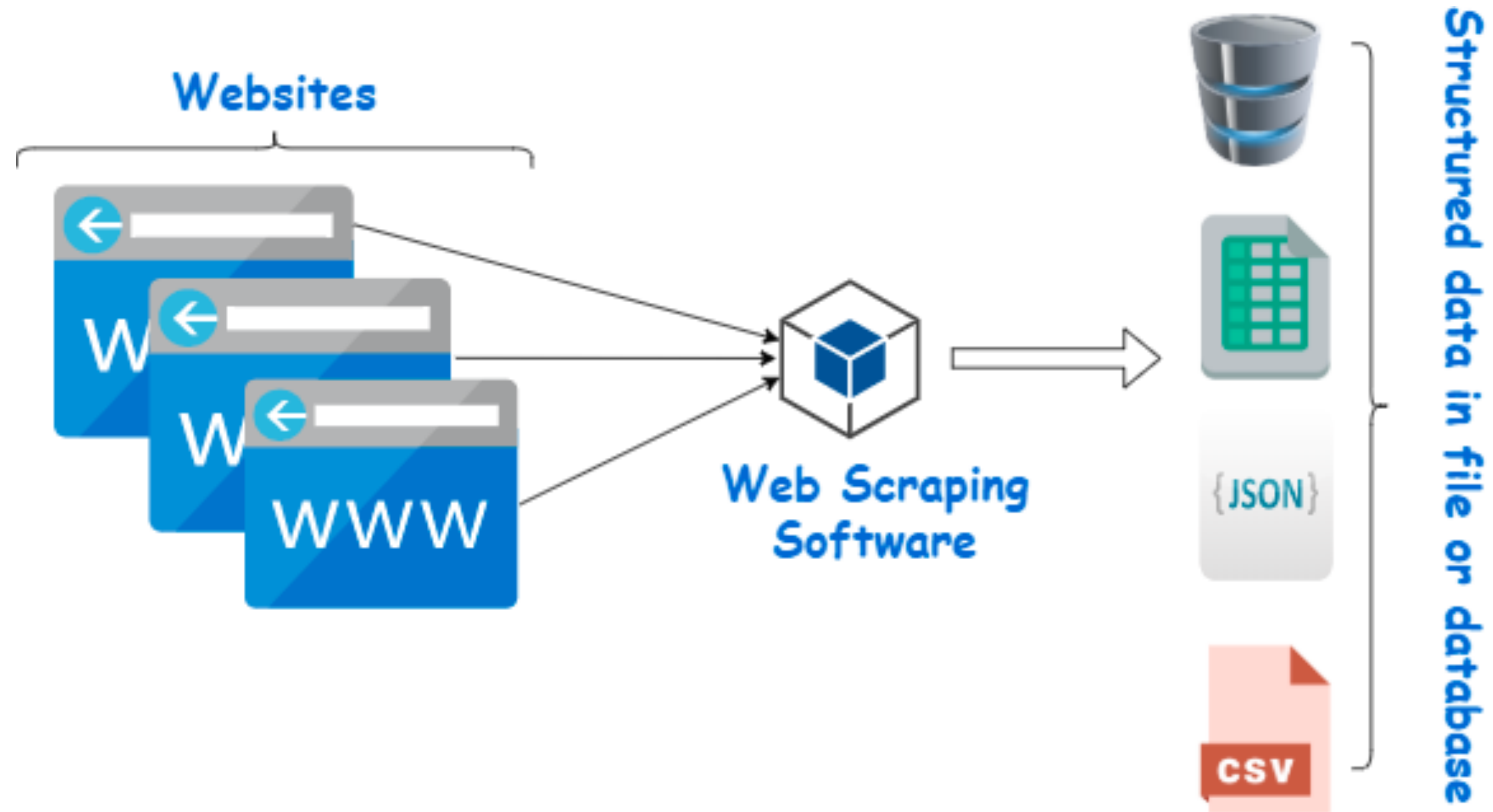
Acesse o HTML da página da web e extraia informações/dados úteis dela. Essa técnica é chamada de web scraping ou web harvesting ou web extraction.

Web scraping é um termo usado para descrever o uso de um programa ou algoritmo para extrair e processar grandes quantidades de dados da web.

Web scraping extrai dados automaticamente e os apresenta em um formato que você pode facilmente entender.



# Web Scrapping



# Vantagens do Web Scrapping

- Web Scrapping diminui o tempo necessário para extrair uma grande quantidade de dados de uma fonte
- Web Scrapping aumenta a precisão da extração de dados
- Web Scrapping é mais rápido do que copiar e colar manualmente os dados

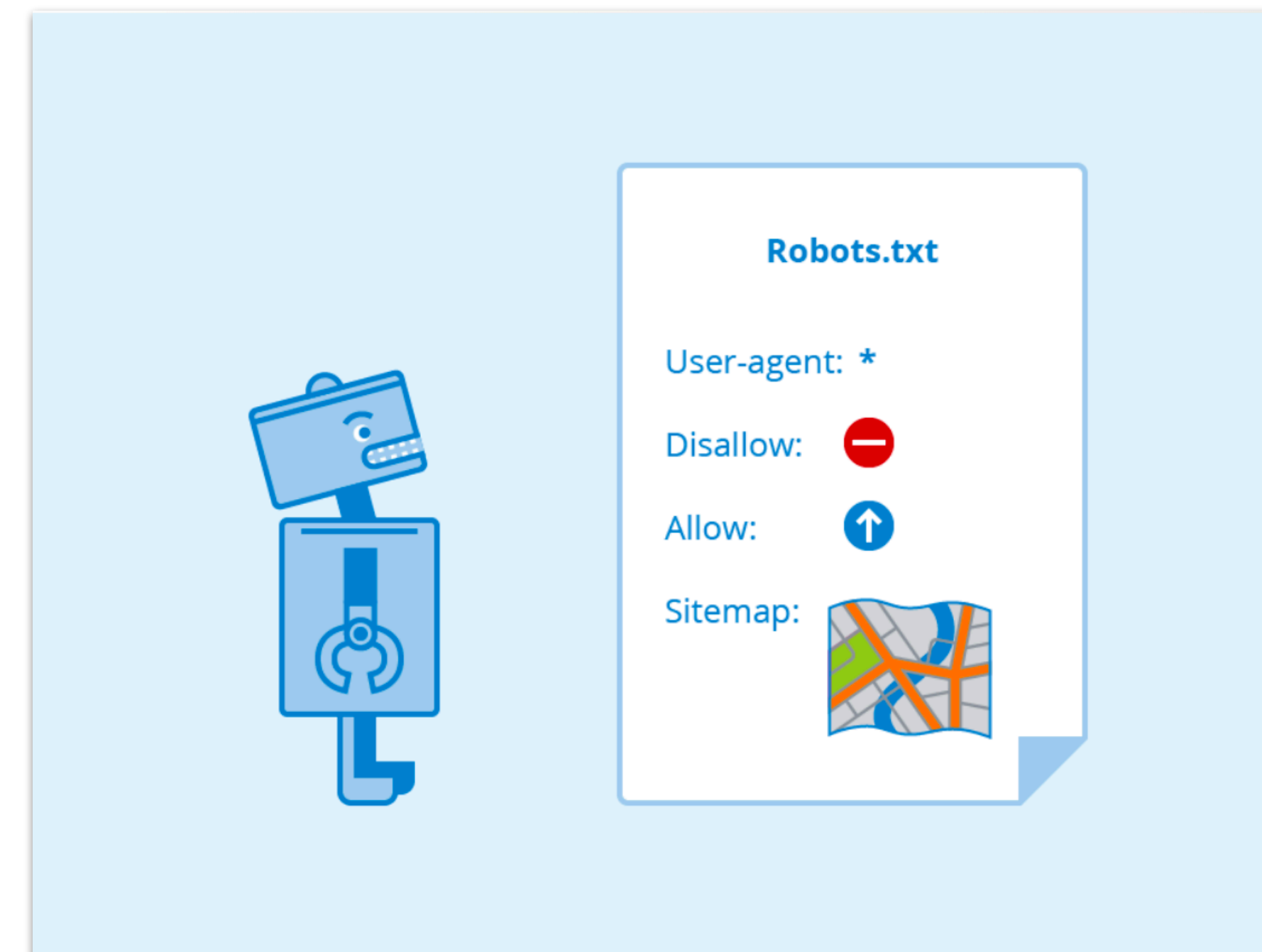
# O que é o robots.txt?

Robots.txt é instruir os robôs da web sobre como rastrear páginas em seu site.

O arquivo robots.txt faz parte do protocolo de exclusão de robôs (REP), um grupo de padrões da web que regulam como os robôs rastreiam a web, acessam e indexam conteúdo para servir esse conteúdo para os usuários.

Para acessar este arquivo use o método abaixo:

[http://\\_domain\\_name/robots.tx](http://_domain_name/robots.tx)



# HTML

## (Hypertext Markup Language)

- ✓ HTML permite formatar texto, adicionar gráficos, criar links, inserir formulários, quadros e tabelas, etc., e salvar tudo em um arquivo de texto que qualquer navegador pode ler e exibir.
- ✓ É uma linguagem de marcação padrão para criar páginas da Web
- ✓ Faz uso de várias tags para formatar o conteúdo
- ✓ HTML é a linguagem mais usada para escrever Web

# Principais Tags HTML

A declaração **<!DOCTYPE html>** define este documento como sendo HTML5

O elemento **<html>** é o elemento raiz de uma página HTML

O elemento **<head>** contém meta-informações sobre o documento

O elemento **<title>** especifica um título para o documento

O elemento **<body>** contém o conteúdo da página visível

O elemento **<h1>** define um título grande

O elemento **<h6>** define um título menor

O elemento **<p>** define um parágrafo

# Beautiful Soup

Beautiful Soup é uma biblioteca Python para extrair dados de arquivos HTML e XML.

- Você pode usá-lo para extrair tabelas, listas, parágrafos e também pode colocar filtros para extrair informações de páginas da web.
- O Beautiful Soup 4 é mais rápido, tem mais recursos e funciona com analisadores de terceiros como lxml e html5lib.

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

**pip install beautifulsoup4**