

```
[1]: !pip install selenium -q
[2]: from selenium import webdriver
from selenium.webdriver.chrome.service import Service
[3]: !pip install beautifulsoup4 -q
[4]: from bs4 import BeautifulSoup
[5]: !pip install pandas -q
[6]: import pandas as pd
[7]: import time
from datetime import date
```

Instanciando o WebDriver

```
[15]: srv = Service('chromedriver.exe')
opt = webdriver.ChromeOptions()
driver = webdriver.Chrome(service=srv, options=opt)
```

Logando em sua Conta no Linkedin

```
[115]: # Abrindo a página de Login do Linkedin
driver.get('https://www.linkedin.com/login-pt')
time.sleep(1)

# Inserindo o usuário
usuario = driver.find_element('id', 'username')
usuario.send_keys('SEU-EMAIL')

# Inserindo a senha
senha = driver.find_element('id', 'password')
senha.send_keys('SUA-SENHA')

# Clicando no botão 'Entrar' para acessar a conta no Linkedin
driver.find_element("xpath", "//button[@type='submit']").click()
```

Acessando um perfil através da URL do Usuário

```
[19]: perfil_url = 'https://www.linkedin.com/in/larissa-akemi-iuki-573321153/'
driver.get(perfil_url)
```

Uma Feature interessante do LinkedIn é a possibilidade de baixar as informações do perfil em um formato de currículo em PDF, então como um extra adicionaremos essa funcionalidade ao nosso Scraping, ou seja, para cada perfil teremos um 'Currículo' em PDF que pode ser explorado com outras ferramentas de extração de dados em documentos.

Alguns perfis possuem diferentes layouts de página, portanto veremos várias partes do código com TRY/EXCEPT para buscar generalizar o código para mais formatos de página.

```
[21]: try:
    # Clicando no botão '...' ou 'Mais' na seção de "Introdução" do perfil
    driver.find_element('xpath', '/html/body/div[5]/div[3]/div/div/div[2]/div/main/section[1]/div[2]/div[3]/div/button').click()
    time.sleep(1) # aguardar 1 segundo

    # Clicando no botão "Salvar como PDF"
    driver.find_element('xpath', '/html/body/div[5]/div[3]/div/div/div[2]/div/main/section[1]/div[2]/div[3]/div/div/ul/li[3]/div').click()

except:
    # Clicando no botão '...' ou 'Mais' na seção de "Introdução" do perfil
    driver.find_element('xpath', '/html/body/div[5]/div[3]/div/div/div[2]/div/main/section[1]/div[2]/div[3]/div/div[2]/button').click()
    time.sleep(1) # aguardar 1 segundo

    # Clicando no botão "Salvar como PDF"
    driver.find_element('xpath', '/html/body/div[5]/div[3]/div/div/div[2]/div/main/section[1]/div[2]/div[3]/div/div/ul/li[3]/div').click()
```

Quais Informações devemos extrair? Voltando a nossa situação problema, digamos que a equipe de RH nos passou uma lista de quais informações são chaves para a tomada de decisão sobre o candidato, que são:

Nome Candidato; Título do Candidato; Organização Atual; Localização do Candidato; Biografia; Título da Última Experiência Profissional; Nome da Organização da Última Experiência Profissional; Tipo de Contratação da Organização; Data de Início da Experiência; Data de Término da Experiência; Duração da Experiência; Instituição referente à Última Formação Acadêmica; Tipo da Formação Acadêmica; Título da Formação Acadêmica; Duração da Formação Acadêmica; Número de Seguidores; Idiomas; Prêmios e Honrarias; Quantidade de Recomendações. Como um extra para gestão de base de dados, vamos acrescentar as seguintes informações:

Data de Ingestão (quando os perfis foram inseridos na base); Disponibilidade da URL para consultas manuais (caso o recrutador queira acessar o perfil do candidato);

Obtendo o Código Fonte HTML da Página

```
[23]: # Lendo a página do início ao fim
inicio = time.time()
posicao_inicial_rolamento = 0
posicao_final_rolamento = 1000

while True:
    driver.execute_script(f"window.scrollTo({posicao_inicial_rolamento},{posicao_final_rolamento})")

    posicao_inicial_rolamento = posicao_final_rolamento
    posicao_final_rolamento += 1000

    # Aguardando 3 segundos
    time.sleep(3)

    fim = time.time()

    # Executar o script por 20 segundos
```

```

if round(fim - inicio) > 20:
    break

[25]: # Salvando o código fonte da página em uma variável
src = driver.page_source

# Utilizando o código fonte para gerar um objeto BeautifulSoup
soup = BeautifulSoup(src, 'lxml')

```

Extraindo as Informações da Introdução do Perfil

Para acessarmos as informações dentro do Objeto com o código fonte, usamos o find(), passando o elemento e a classe de onde está localizada a seção Introdução (para localizar esses parâmetros basta inspecionar os elementos na página do driver).

Uma das complicações do processo de extração está na identificação de onde estão as informações que queremos, pois muitos elementos no código fonte da página mudam conforme a sua seção de login, dessa forma localizar os elementos corretos que não sofrem alterações é um trabalho árduo.

```

[27]: # Obtendo o HTML de extração da introdução
intro = soup.find('div', {'class': 'mt2 relative'})

# Visualizando o código HTML
print(intro)

<div class="mt2 relative">
<div class="MKqlftfYTrDzofnnmBnohPNkZsPc">
<div class="XDUWzpRGnKPVqabnydEEthnZBvhYZBKGNc">
<span class="artdeco-hoverable-trigger artdeco-hoverable-trigger--content-placed-bottom artdeco-hoverable-trigger--is-hoverable ember-view" id="ember38" tabindex="-1">
<a aria-describedby="artdeco-hoverable-artdeco-gen-42" aria-label="Larissa Akemi tem verificações" class="ember-view RVkdjeuLkbLxQGxaArXphLAgsHdrrqEyAtCYq" href="/in/larissa-akemi-iuki-573321153/overlay/about-this-profile/" id="ember39">
<h1 class="text-heading-xlarge inline t-24 v-align-middle break-words">Larissa Akemi Iuki</h1>
<svg aria-hidden="true" class="v-align-middle t-black--light" data-supported-dps="24x24" data-test-icon="verified-medium" height="24" role="none" viewBox="0 24 24" width="24" xmlns="http://www.w3.org/2000/svg">
<!-- -->
<use height="24" href="#verified-medium" width="24"/>
</svg>
<!-- --> </a>
<div class="ember-view" id="artdeco-gen-42"><div class="ember-view" id="ember41"></div></div>
</span>
<!-- --><!-- --><!-- --> </div>
<!-- --> <div class="text-body-medium break-words" data-generated-suggestion-target="urn:li:fsu_profileActionDelegate:1460363610">
    Desenvolvedora Python Jr para IA no Grupo Regazzo | Analista de dados e Inteligência Artificial
</div>
<!-- --><!-- --><!-- --> </div>
<ul class="bxQhtCILGhQwAiTsikhTQdkTKDYxZjI">
<li class="vHIverptNZAGrLDAmqTEGLQiQjJbzg">
<button aria-label="Empresa atual: Grupo Regazzo. Dê um clique para pular para o módulo de experiência" class="JMpjRVRaMGQSAyezvmpAvyLxJAwxBrmzCo text-align-left" type="button">

<span class="XplngfwCxyMKlnDvIHdwDvfwsmMmtCQug hoverable-link-text break-words text-body-small t-black">
<div class="AatKoQwEMdqLKJpUYfBnnUVrTqZmSyyeIUjh0 inline-show-more-text--is-collapsed inline-show-more-text--is-collapsed inline-show-more-text--is-collapsed-with-line-clamp" style="-webkit-line-clamp:2;" tabindex="-1">
    Grupo Regazzo
<!-- --> </div>
</span>
</button>
</li>
<li class="JvHIverptNZAGrLDAmqTEGLQiQjJbzg">
<button aria-label="Formação acadêmica: FAE Centro Universitário. Dê um clique para pular para o módulo de formação" class="JMpjRVRaMGQSAyezvmpAvyLxJAwxBrmzCo text-align-left" type="button">

<span class="XplngfwCxyMKlnDvIHdwDvfwsmMmtCQug hoverable-link-text break-words text-body-small t-black">
<div class="AatKoQwEMdqLKJpUYfBnnUVrTqZmSyyeIUjh0 inline-show-more-text--is-collapsed inline-show-more-text--is-collapsed inline-show-more-text--is-collapsed-with-line-clamp" style="-webkit-line-clamp:2;" tabindex="-1">
    FAE Centro Universitário
<!-- --> </div>
</span>
</button>
</li>
</ul>
<div class="MKqlftfYTrDzofnnmBnohPNkZsPc mt2">
<!-- --> <span class="text-body-small inline t-black--light break-words">
    Curitiba, Paraná, Brasil
</span>
<span class="MCMyQZQvGJrlbbDQXiuKftAnOphLmxrGQ t-black--light">
<a class="ember-view link-without-visited-state cursor-pointer text-heading-small inline-block break-words" href="/in/larissa-akemi-iuki-573321153/overlay/contact-info/" id="top-card-text-details-contact-info">
        Informações de contato
    </a>
</span>
</div>
</div>

```

Agora que temos o código localizado da Seção Introdução, podemos extrair as informações: Nome do Candidato, Título Profissional, Localização, Organização Atual e Formação Acadêmica. Como podem haver perfis de candidatos desempregados e também que não possuem formação acadêmica, esses campos precisarão ser condicionado

```

[37]: if intro:
    # Localizando e extraíndo o nome do candidato
    nome_loc = intro.find('h1', {'class': 'text-heading-xlarge inline t-24 v-align-middle break-words'})
    nome = nome_loc.get_text().strip() if nome_loc else 'Nome não encontrado'

    # Localizando e extraíndo o título profissional do candidato
    titulo_loc = intro.find('div', {'class': 'text-body-medium break-words'})
    titulo = titulo_loc.get_text().strip() if titulo_loc else 'Título não encontrado'

    # Localizando e extraíndo o endereço de residência
    localizacao_loc = intro.find('span', {'class': 'text-body-small inline t-black--light break-words'})
    localizacao = localizacao_loc.get_text().strip() if localizacao_loc else 'Localização não encontrada'

    # Localizando e extraíndo a organização de trabalho
    trabalho_loc = intro.find('div', {'class': 'text-body-medium break-words'})
    trabalho = trabalho_loc.get_text().strip() if trabalho_loc else 'Trabalho não encontrado'

    # Localizando e extraíndo a instituição da formação acadêmica
    estudo_loc = intro.find_all('div', {'class': 'AatKoQwEMdqLKJpUYfBnnUVrTqZmSyyeIUjh0 inline-show-more-text--is-collapsed inline-show-more-text--is-collapsed inline-show-more-text--is-collapsed inline-show-more-text--is-collapsed'})
    estudo = estudo_loc[1].get_text().strip() if len(estudo_loc) > 1 else 'Estudo não encontrado'

```

```
# Exibindo os resultados obtidos até agora
print('Nome:', nome,
      '\nTítulo:', titulo,
      '\nLocalização:', localizacao,
      '\nTrabalha em:', trabalho,
      '\nEstudou em:', estudo)
else:
    print("A introdução não foi encontrada no HTML fornecido.")

Nome: Larissa Akemi Iuki
Título: Desenvolvedora Python Jr para IA no Grupo Regazzo | Analista de dados e Inteligência Artificial
Localização: Curitiba, Paraná, Brasil
Trabalha em: Grupo Regazzo
Estudou em: FAE Centro Universitário
```

Extraindo a Quantidade de Seguidores da Página

```
[39]: # Imprimir o HTML para diagnóstico
print("HTML Completo da Página:")
print(soup.prettify())
```

```
[41]: # Visualizando o código de seguidores
seguidores_xpath = soup.find('li', {'class': 'text-body-small t-black--light inline-block'}).get_text(strip=True)

print('O perfil possui:', seguidores_xpath)

O perfil possui: 1.704 seguidores
```

Extraindo as Informações da Biografiaaa

Essa informação está em uma Seção chamada 'Sobre' no perfil e como também há perfis que não possuem uma Biografia, também teremos que condicionar essa informação.

Lembra que comentei sobre os diferentes layouts das páginas dos usuários quando fizemos o Download do Perfil em PDF? Bom, o mesmo problema acontece aqui, em alguns casos o código abaixo funciona corretamente, porém em outros acaba pegando informações erradas, ou seja, é necessário tratar a base de dados gerada após o processo de Web Scraping.

```
[61]: # Obtendo o HTML de extração da biografia do candidato
sobre_descricao = soup.find('body','(class':'render-mode-BIGPIPE nav-v2 ember-application icons-loaded boot-complete'))\n\n# Visualizando o código HTML
print(sobre_descricao)
```

```
<body class="render-mode-BIGPIPE nav-v2 ember-application icons-loaded boot-complete" data-t-link-to-event-attached="true" dir="ltr">
<!-- Start of PMBR-7487 -->
<script nonce="" src="https://static.licdn.com/aero-v1/sc/h/988vmt8bv2rfmpquw6nnswc5t"></script>
<!-- End of PMBR-7487 -->
<!-- HUED-11420 -->
<div id="artdeco-toasts__wormhole"> <section aria-label="Mensagem toast" class="artdeco-toasts" id="artdeco-toasts">
<header class="artdeco-toasts__header">
<h2 class="artdeco-toasts__title">
    0 notificações no total
  </h2>
</header>
<!-- -->
<div class="artdeco-toasts__toasts" role="alert">
<!-- --> </div>
</section>
</div>
<!-- EMBER_CLI_FASTBOOT_BODY -->
<div class="app-boot-bg-skeleton" id="app-boot-bg-loader">
```

<div class="AatKoQwEmdqLKJpUYfBnnUVrTqZmSyyeIUjho inline-show-more-text--is-collapsed inline-show-more-text--is-collapsed-with-line-clamp full-width" style="webkit-line-clamp:4;" tabindex="1">
<!--Como Analista de Dados e Engenheira Ambiental e Sanitária, busco soluções inovadoras e impactantes para os problemas reais, usando dados e evidências. Sou formada pela FAE Centro Universitário e me capacitei com cursos de Data Analytics, Data Science e Machine Learning pela Tera, uma escola de tecnologia e negócios.<!-->

<!-->Na TAG IMF, uma empresa de soluções financeiras, realizei relatórios, dashboards e análises de indicadores, processos, estratégias e regras de negócio, contribuindo para a melhoria da qualidade e da eficiência dos serviços prestados aos clientes. Antes disso, tive experiências em Business Intelligence e Business Development, onde desenvolvi habilidades de coleta, organização e interpretação de dados, além de gestão de projetos e vendas. Também realizei um intercâmbio na Austrália, onde estudei e trabalhei por um ano e três meses, aprimorando meu inglês e minha visão global. Meu objetivo é continuar aprendendo e me desafiando na área de Análise de Dados.<!--><!-->Como Analista de Dados e Engenheira Ambiental e Sanitária, busco soluções inovadoras e impactantes para os problemas reais, usando dados e evidências. Sou formada pela FAE Centro Universitário e me capacitei com cursos de Data Analytics, Data Science e Machine Learning pela Tera, uma escola de tecnologia e negócios.

Na TAG IMF, uma empresa de soluções financeiras, realizei relatórios, dashboards e análises de indicadores, processos, estratégias e regras de negócio, contribuindo para a melhoria da qualidade e da eficiência dos serviços prestados aos clientes. Antes disso, tive experiências em Business Intelligence e Business Development, onde desenvolvi habilidades de coleta, organização e interpretação de dados, além de gestão de projetos e vendas. Também realizei um intercâmbio na Austrália, onde estudei e trabalhei por um ano e três meses, aprimorando meu inglês e minha visão global. Meu objetivo é continuar aprendendo e me desafiando na área de Análise de Dados. <!-- -->/span>

```
[71]: # Extrair texto de ambas as seções
biografia_visivel = sobre_descricao2.find('span', {'aria-hidden': 'true'}).get_text(separator=' ', strip=True)
biografia_oculta = sobre_descricao2.find('span', {'class': 'visually-hidden'}).get_text(separator=' ', strip=True)

# Concatenar biografias
biografia_completa = biografia_visivel + ' ' + biografia_oculta
```

```
print('Biografia Completa:', biografia_completa)
```

Biografia Completa: Como Analista de Dados e Engenheira Ambiental e Sanitária, busco soluções inovadoras e impactantes para os problemas reais, usando dados e evidências. Sou formada pela FAE Centro Universitário e me capacitei com cursos de Data Analytics, Data Science e Machine Learning pela Tera, uma escola de tecnologia e negócios. Na TAG IMF, uma empresa de soluções financeiras, realizei relatórios, dashboards e análises de indicadores, processos, estratégias e regras de negócio, contribuindo para a melhoria da qualidade e da eficiência dos serviços prestados aos clientes. Antes disso, tive experiências em Business Intelligence e Business Development, onde desenvolvi habilidades de coleta, organização e interpretação de dados, além de gestão de projetos e vendas. Também realizei um intercâmbio na Austrália, onde estudei e trabalhei por um ano e três meses, aprimorando meu inglês e minha visão global. Meu objetivo é continuar aprendendo e me desafiando na área de Análise de Dados. Como Analista de Dados e Engenheira Ambiental e Sanitária, busco soluções inovadoras e impactantes para os problemas reais, usando dados e evidências. Sou formada pela FAE Centro Universitário e me capacitei com cursos de Data Analytics, Data Science e Machine Learning pela Tera, uma escola de tecnologia e negócios.

Na TAG IMF, uma empresa de soluções financeiras, realizei relatórios, dashboards e análises de indicadores, processos, estratégias e regras de negócio, contribuindo para a melhoria da qualidade e da eficiência dos serviços prestados aos clientes. Antes disso, tive experiências em Business Intelligence e Business Development, onde desenvolvi habilidades de coleta, organização e interpretação de dados, além de gestão de projetos e vendas. Também realizei um intercâmbio na Austrália, onde estudei e trabalhei por um ano e três meses, aprimorando meu inglês e minha visão global. Meu objetivo é continuar aprendendo e me desafiando na área de Análise de Dados.

Extraindo as Informações referentes à Última Experiência Profissional

Essa é uma das seções que começam a complicar bastante a extração do Web Scraping, já que também temos perfis com diferentes layouts nessa seção.

Como é possível observar na imagem acima, há três formas de uma pessoa inserir as informações sobre suas experiências, as vezes com informações básicas, outras com descrições das atividades realizadas e também casos como o primeiro item, onde dentro de uma mesma organização a pessoa passou por diversos cargos. Todas essas mudanças de estrutura tornam difícil a generalização do processo de Web Scraping, caindo sempre no mesmo problema que é nem sempre conseguir pegar as informações corretas.

A seção Experiência possui uma página exclusiva, o que torna mais fácil extrair as informações do que se fossem extraí-las da página principal, então vamos acessar essa página referente a Experiência Profissional.

Como estamos abrindo uma nova página, é importante executar o roolamento dela até o final para pegar todas as informações:

```
[73]: # Acessando a página de Experiência Profissional
driver.get('https://www.linkedin.com/in/larissa-akemi-iuki-573321153/details/experience/')

# Lendo a página de inicio ao fim
inicio = time.time()
posicao_inicial_rolamento = 0
posicao_final_rolamento = 1000

while True:
    driver.execute_script(f"window.scrollTo({posicao_inicial_rolamento},{posicao_final_rolamento})")

    posicao_inicial_rolamento = posicao_final_rolamento
    posicao_final_rolamento += 1000

    # Aguardando 3 segundos
    time.sleep(3)

    fim = time.time()

    # Executar o script por 20 segundos
    if round(fim - inicio) > 20:
        break

# Salvando o código fonte da página em uma variável
src_experiencia = driver.page_source

# Utilizando o código fonte para gerar um objeto BeautifulSoup
soup_experiencia = BeautifulSoup(src_experiencia, 'lxml')

print(soup_experiencia)
```

```
<html class="theme theme--mercado app-loader--default artdeco windows" lang="pt"><head>
<script nonce="">function(i,n){void 0!=i.addEventListener&&void 0!=i.hidden&&(n.liVisibilityChangeListener=function(){i.hidden&&(n.liHasWindowHidden=10),i.addEventListener("visibilitychange",n.liVisibilityChangeListener)})(document,window);</script>
<meta content="script-src attr='none'; require-trusted-types-for 'script'; trusted-types 'allow-duplicates' default jSecure highcharts dompurify" data-distribution="enforce" data-report-to="https://www.linkedin.com/security/csp?a=voyager-web&#038;m=bpr" data-sanitizer="jSecure" http-equiv="Content-Security-Policy" name="trusted-types"/>
<title>(7) Experiência | Larissa Akemi Iuki | LinkedIn</title>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta class="mercado-icons-sprite" content="https://static.licdn.com/aero-v1/sc/h/2brfqc2ezdcl00afxryk15mid" id="artdeco-icons/static/images/sprite-asset-name="asset-url"/>
<meta content="" name="description"/>
<meta content="notranslate" name="google"/>
<meta content="voyager-web" name="service"/>
<meta content="#ffffff" id="theme-color-meta-tag" name="theme-color"/>
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0" name="viewport"/>
<meta content="https://static.licdn.com/aero-v1/sc/h/baseCDNU1"/>
<link href="https://static.licdn.com/aero-v1/sc/h/5bukxbhy9xsil5mb7c2wulfbx" id="favicon-svg" rel="icon" type="image/svg+xml"/>
```

```
[91]: # Encontrar todas as entradas de experiência
experiencias = soup_experiencia.find_all('li', class_='pvs-list__paged-list-item')

print(experiencias)
```

```
<li class="pvs-list__paged-list-item artdeco-list__item pvs-list__item--line-separated pvs-list__item--one-column" id="profilePagedListComponent-ACoAAC
Tjf-cBExuzi7qghn3o5SpjDViJneHcvdc-EXPERIENCE-VIEW-DETAILS-profile-ACoAACTjf-cBExuzi7qghn3o5SpjDViJneHcvdc-NONE-pt-BR-0">
<div>
<!--> <div class="dXpwMfrKvOOHEhxAPDefaVfRmIppM rPaeWespKszgJATXUohrjEMeNoda LTxItSphOrYqrNvGpuJlJFLepxpbFAJvtIeg" data-view-name="profile-component-entity">
<div>
<a class="optional-action-target-wrapper display-flex" href="https://www.linkedin.com/company/3614814/" target="_self">
<div class="ivm-image-view-model pvs-entity__image">
<div class="ivm-view-attr__img-wrapper">
<!-->
<!--> 
</div>
</div>
</a>
</div>
<div class="display-flex flex-column full-width align-self-center">
```

```
[95]: # Encontrar todas as entradas de experiência
experiencias = soup_experiencia.find_all('li', class_='pvs-list__paged-list-item')

# Iterar sobre cada experiência e extraír as informações
for experiencia in experiencias:
    # Extraír título do cargo
    titulo_cargo_tap = experiencia.find('div', class_='display-flex align-items-center mr1 t-bold')
```

```

    titulo_cargo = titulo_cargo_tag.find('span', {'aria-hidden': 'true'}).get_text(strip=True) if titulo_cargo_tag else 'Cargo não encontrado'

    # Extrair nome da empresa
    empresa_tag = experiencia.find('span', class_='t-14 t-normal')
    empresa = empresa_tag.find('span', {'aria-hidden': 'true'}).get_text(strip=True) if empresa_tag else 'Empresa não encontrada'

    # Extrair datas de emprego
    datas_tag = experiencia.find('span', class_='pvs-entity__caption-wrapper')
    datas = datas_tag.get_text(strip=True) if datas_tag else 'Datas não encontradas'

    # Extrair Localização
    localizacao_tags = experiencia.find_all('span', class_='t-14 t-normal t-black--light')
    localizacao = localizacao_tags[1].get_text(strip=True) if len(localizacao_tags) > 1 else 'Localização não encontrada'

    # Extrair descrições de atividades
    descricao_tags = experiencia.find_all('span', {'aria-hidden': 'true', 'class': 'None'})
    descricoes = [descricao.get_text(separator=' ', strip=True) for descricao in descricao_tags]

    # Extrair competências
    competencias_tags = experiencia.find_all('div', class_='display-flex align-items-center t-14 t-normal t-black')
    competencias = 'Competências não encontradas'
    for comp_tag in competencias_tags:
        if 'Competências:' in comp_tag.get_text():
            competencias_span = comp_tag.find('span', {'aria-hidden': 'true'})
            competencias = competencias_span.get_text(strip=True) if competencias_span else competencias

    # Filtrar descrições para remover duplicatas ou itens desnecessários
    descricoes_filtradas = list(dict.fromkeys([d for d in descricoes if 'Competências:' not in d]))

    # Imprimir as informações extraídas
    print(f"Titulo do Cargo: {titulo_cargo}")
    print(f"Empresa: {empresa}")
    print(f"Datas de Emprego: {datas}")
    print(f"Localização: {localizacao}")
    print("Descrições de Atividades:")
    for descricao in descricoes_filtradas:
        print(f"- {descricao}")
    print(f"Competências: {competencias}")
    print("-" * 50)

Título do Cargo: Desenvolvedora Python Jr para IA
Empresa: Grupo Regazzo · Tempo integral
Datas de Emprego: maio de 2024 - o momento · 4 meses
Localização: Localização não encontrada
Descrições de Atividades:
- Desenvolvedora Python Jr para IA
- Grupo Regazzo · Tempo integral
- Realizou testes com agentes, validando suas aplicações na área de inteligência artificial. Utiliza como framework o Langchain e LangGraph para desenvolver os projetos, além de validar diversos vector stores como Pinecone, Deeplake e Azure Search. Minha atuação inclui a criação de soluções inovadoras que optimizam processos empresariais e aumentando a eficiência operacional.
Competências: Competências não encontradas
-----
Título do Cargo: Desenvolvimento profissional
Empresa: Pausa na carreira
Datas de Emprego: dez de 2022 - abr de 2024 · 1 ano 5 meses
Localização: Sydney, New South WalesSydney, New South Wales
Descrições de Atividades:
- Desenvolvimento profissional

```

Extraindo as Informações referente à Formação Acadêmica mais Recente

Agora vamos extrair as informações da seção Formação Acadêmica, e aqui também temos mais de um formato de estrutura, ou seja, é necessário criar duas funções para pegar os tipos: informações básicas sobre a formação acadêmica e a estrutura em que o usuário descreve as atividades realizadas durante a formação.

Para a Formação Acadêmica, também temos uma página exclusiva, portanto vamos acessá-la:

```

[104]: # Acessando a página de Formação Acadêmica
driver.get('https://www.linkedin.com/in/larissa-akemi-iuki-573321153/details/education/')

# Lendo a página do inicio ao fim
inicio = time.time()
posicao_inicial_rolamento = 0
posicao_final_rolamento = 1000

while True:
    driver.execute_script(f"window.scrollTo({posicao_inicial_rolamento},{posicao_final_rolamento})")

    posicao_inicial_rolamento = posicao_final_rolamento
    posicao_final_rolamento += 1000

    # Aguardando 3 segundos
    time.sleep(3)

    fim = time.time()

    # Executar o script por 20 segundos
    if round(fim - inicio) > 20:
        break

# Salvando o código fonte da página em uma variável
src_educacao = driver.page_source

# Utilizando o código fonte para gerar um objeto BeautifulSoup
soup_educacao = BeautifulSoup(src_educacao, 'lxml')

print(soup_educacao)

<html class="theme theme--mercado app-loader--default artdeco windows" lang="pt"><head>
<script nonce="">!function(i,n){void 0!=i.addEventListener&&void 0!=i.hidden&&(n.liVisibilityChangeListener=function(){i.hidden&&(n.liHasWindowHidden=!0),i.addEventListener("visibilitychange",n.liVisibilityChangeListener)})(document>window);</script>
<meta content="script-src attr '_none' require-trusted-types-for 'script'; trusted-types 'allow-duplicates' default jSecure highcharts dompurify" data-d
isposition="enforce" data-report-to="https://www.linkedin.com/security/csp?a=voyager-web&m=bpr" data-sanitizer="jSecure" http-equiv="Content-Securit
y-Policy" name="trusted-types"/>
<title>(8) Formação acadêmica | Larissa Akemi Iuki | LinkedIn</title>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta class="mercado-icons-sprite" content="https://static.licdn.com/aero-v1/sc/h/2brfqc2ezdcl00afxryk15mid" id="artdeco-icons/static/images/sprite-as
et" name="asset-url"/>
<meta content="" name="description"/>
<meta content="notranslate" name="google"/>
<meta content="voyager-web" name="service"/>
<meta content="#ffffff" id="theme-color-meta-tag" name="theme-color"/>
```

```

<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0" name="viewport"/>
<meta content="https://static.linkedin.com/aero-v1/sc/h" name="baseCDNUrl"/>
<link href="https://static.linkedin.com/aero-v1/sc/h/5bukxbhy9xsi15mb7c2wulfbx" id="favicon-svg" rel="icon" type="image/svg+xml"/>

```

```

[106]: # Encontrar todas as entradas de educação
educacoes = soup_educacao.find_all('li', class_='pvs-list__paged-list-item')

# Iterar sobre cada educação e extrair as informações
for educacao in educacoes:
    # Extrair nome da instituição
    instituicao_tag = educacao.find('div', class_='display-flex align-items-center t-11 hoverable-link-text t-bold')
    instituicao = instituicao_tag.find('span', {'aria-hidden': 'true'}).get_text(strip=True) if instituicao_tag else 'Instituição não encontrada'

    # Extrair curso
    curso_tag = educacao.find('span', class_='t-14 t-normal')
    curso = curso_tag.find('span', {'aria-hidden': 'true'}).get_text(strip=True) if curso_tag else 'Curso não encontrado'

    # Extrair datas de estudo
    datas_tag = educacao.find('span', class_='pvs-entity__caption-wrapper')
    datas = datas_tag.get_text(strip=True) if datas_tag else 'Datas não encontradas'

    # Extrair descrição do curso
    descricao_tag = educacao.find('div', class_='display-flex align-items-center t-14 t-normal t-black')
    descricao = descricao_tag.find('span', {'aria-hidden': 'true'}).get_text(strip=True) if descricao_tag else 'Descrição não encontrada'

    # Extrair competências
    competencias_tags = educacao.find_all('div', class_='display-flex align-items-center t-14 t-normal t-black')
    competencias = 'Competências não encontradas'
    for comp_tag in competencias_tags:
        if 'Competências:' in comp_tag.get_text():
            competencias_span = comp_tag.find('span', {'aria-hidden': 'true'})
            competencias = competencias_span.get_text(strip=True) if competencias_span else competencias

    # Imprimir as informações extraídas
    print(f"Instituição: {instituicao}")
    print(f"Curso: {curso}")
    print(f"Datas de Estudo: {datas}")
    print(f"Descrição do Curso: {descricao}")
    print(f"Competências: {competencias}")
    print("-" * 50)

```

Instituição: FAE Centro Universitário

Curso: Bacharelado, Engenharia Ambiental e Sanitária

Datas de Estudo: 2014 - 2018

Descrição do Curso: Competências:Desenvolvimento de novos negócios

Competências: Competências:Desenvolvimento de novos negócios

Instituição: The Sydney Business & Travel Academy (SBTA)

Curso: English, Exchange

Datas de Estudo: dez de 2022 - abr de 2023

Descrição do Curso: Durante um ano em Sydney, tive a oportunidade de mergulhar no aprendizado da língua inglesa em um ambiente imersivo. Frequentei aulas diárias em uma escola de idiomas renomada, onde aprimorei minhas habilidades de conversação, compreensão auditiva, leitura e escrita. Além das aulas formais, participei de atividades extracurriculares e interagi com falantes nativos, o que contribuiu significativamente para minha fluência e confiança no idioma. Essa experiência enriquecedora não apenas aprimorou minha proficiência no inglês, mas também ampliou minha compreensão da cultura e da vida cotidiana na Austrália.

Competências: Competências não encontradas

Instituição: Tera

Curso: Data Analytics, Technology

Datas de Estudo: nov de 2021 - abr de 2022

Descrição do Curso: No curso de Data Analyst da Tera, explorei diversas ferramentas e conceitos essenciais para análise de dados. Aprendi a utilizar SQL para manipulação e extração de dados, além de ferramentas como Google Data Studio, Power BI, Tableau e Google Analytics para visualização e análise de dados. O curso também abordou conceitos de ciência de dados, proporcionando uma compreensão sólida dos processos e técnicas para análise de dados. Essa experiência me permitiu adquirir habilidades práticas e conhecimentos fundamentais para atuar como analista de dados em diversos contextos profissionais.

Competências: Competências:Visualização de dados · SQL · Google Data Studio · Ciência de dados · Power bi · Google Analytics · Tableau

Instituição: Tera

Curso: Data Science & Machine Learning, Technology

Datas de Estudo: 2020 - 2021

Descrição do Curso: No curso de Ciência de Dados e Machine Learning da Tera, explorei um conjunto abrangente de ferramentas e conceitos fundamentais para a análise avançada de dados. Aprendi a utilizar ferramentas de visualização como Google Data Studio, Power BI e Tableau, bem como técnicas de análise de dados e ciência de dados. Além disso, adquiri habilidades em Python para manipulação de dados, aprendizagem estatística e técnicas avançadas de Machine Learning. Essa experiência me proporcionou uma base sólida para aplicar técnicas avançadas de análise de dados e machine learning em diversos projetos e cenários profissionais.

Competências: Competências:Visualização de dados · Google Data Studio · Ciência de dados · Power bi · Google Analytics · Tableau · Python · Aprendizagem estatística · Machine Learning

Instituição: Universidade Positivo

Curso: Ensino Técnico, Técnico em segurança do trabalho

Datas de Estudo: 2015 - 2016

Descrição do Curso: Descrição não encontrada

Competências: Competências não encontradas

Extraindo as Informações referentes as Competências

```

[17]: # Acessando a página de Competências
driver.get('https://www.linkedin.com/in/larissa-akemi-573321153/details/skills/')

# Lendo a página do inicio ao fim
inicio = time.time()
posicao_inicial_rolamento = 0
posicao_final_rolamento = 1000

while True:
    driver.execute_script(f"window.scrollTo({posicao_inicial_rolamento},{posicao_final_rolamento})")

    posicao_inicial_rolamento = posicao_final_rolamento
    posicao_final_rolamento += 1000

    # Aguardando 3 segundos
    time.sleep(3)

    fim = time.time()

    # Executar o script por 20 segundos
    if round(fim - inicio) > 20:
        break

# Salvando o código fonte da página em uma variável
src_competencia = driver.page_source

# Utilizando o código fonte para gerar um objeto BeautifulSoup
soup_competencia = BeautifulSoup(src_competencia, 'lxml')

```

```
[18]: print(soup_competencia)
<div>
<div aria-hidden="true" class="display-flex tabindex=-1">
<!-- -->
</div>
</div>
<div class="display-flex flex-column full-width align-self-center">
<div class="display-flex flex-row justify-space-between">
<a class="optional-action-target-wrapper display-flex flex-column full-width" data-field="skill_page_skill_topic" href="https://www.linkedin.com/search/results/all/?keywords=Tecnologia+da+informa%C3%A7%C3%A3o&origin=PROFILE_PAGE_SKILL_NAVIGATION" target=_self">
<div class="display-flex flex-wrap align-items-center full-height">
<div class="display-flex">
<div class="display-flex full-width">
<div class="display-flex align-items-center mr1 hoverable-link-text t-bold">
<span aria-hidden="true"><!-- -->Tecnologia da informa%C3%A7%C3%A3o<!-- --><span class="visually-hidden"><!-- -->Tecnologia da informa%C3%A7%C3%A3o<!-- --></span>


[57]: # Encontrar todos os blocos de competências



```
skill_blocks = soup_competencia.find_all('div', class_='display-flex align-items-center mr1 hoverable-link-text t-bold')

Iterar sobre cada bloco e extrair o nome da competência
skills = []
for block in skill_blocks:
 skill_name_tag = block.find('span', {'aria-hidden': 'true'})
 if skill_name_tag:
 skill_name = skill_name_tag.get_text(strip=True)
 skills.append(skill_name)

Exibir as competências extraídas
for i, skill in enumerate(skills, start=1):
 print(f'Competência {i}: {skill}')

Competência 1: Sistemas de inteligência artificial
Competência 2: LangChain
Competência 3: Langraph
Competência 4: Prompt de comando
Competência 5: IA Prompting
Competência 6: Inteligência artificial
Competência 7: Aprendizagem estatística
Competência 8: Machine Learning
Competência 9: Vendas
Competência 10: Capacidade de organização
Competência 11: Resolução de problemas
Competência 12: Tecnologia da informação
Competência 13: Desenvolvimento de novos negócios
Competência 14: Liderança educacional
Competência 15: Estatística
Competência 16: CSS
Competência 17: Analítica de dados
Competência 18: Atendimento ao cliente
Competência 19: Atividades de integração de equipe
Competência 20: Google Analytics
Competência 21: Sistemas de inteligência artificial
Competência 22: Prompt de comando
Competência 23: Inteligência artificial
Competência 24: Aprendizagem estatística
Competência 25: Machine Learning
Competência 26: Vendas
Competência 27: Tecnologia da informação
Competência 28: Desenvolvimento de novos negócios
Competência 29: Liderança educacional
Competência 30: Estatística
Competência 31: Analítica de dados
Competência 32: Atendimento ao cliente
Competência 33: Atividades de integração de equipe
Competência 34: Ciência de dados
Competência 35: Visualização de dados
Competência 36: Coleta de dados
Competência 37: Análise de dados
Competência 38: Aprendizado de máquina
Competência 39: Educação ambiental
Competência 40: Planejamento urbano
Competência 41: CSS
Competência 42: Google Analytics
Competência 43: Google Data Studio
Competência 44: Power bi
Competência 45: Tableau
Competência 46: Python
Competência 47: SQL
Competência 48: Microsoft Excel
Competência 49: AutoCAD
Competência 50: Microsoft Word
Competência 51: Microsoft PowerPoint
Competência 52: C++
Competência 53: HTML
Competência 54: JavaScript
Competência 55: Microsoft Office
Competência 56: Capacidade de organização
Competência 57: Resolução de problemas
Competência 58: Comunicação
Competência 59: Trabalho em equipe
Competência 60: LangChain
Competência 61: IA Prompting
Competência 62: Alicia Saturnino
Competência 63: Simone Lindroth
Competência 64: Bruna Eguti
Competência 65: Caio Canuto
Competência 66: Suellen Martins
```


```

Extraindo as Informações sobre os Idiomas

```
[87]: # Acessando a página de Idiomas
driver.get('https://www.linkedin.com/in/larissa-akemi-iuki-573321153/details/languages/')

# Lendo a página do início ao fim
inicio = time.time()
posicao_inicial_rolamento = 0
posicao_final_rolamento = 1000

while True:
    driver.execute_script(f>window.scrollTo({posicao_inicial_rolamento},{posicao_final_rolamento})
```

```

posicao_inicial_rolamento = posicao_final_rolamento
posicao_final_rolamento += 1000

# Aguardando 3 segundos
time.sleep(3)

fim = time.time()

# Executar o script por 20 segundos
if round(fim - inicio) > 20:
    break

# Salvando o código fonte da página em uma variável
src_idiomas = driver.page_source

# Parseando o HTML usando BeautifulSoup
soup_language = BeautifulSoup(src_idiomas, 'html.parser')

print(soup_language)

<html class="theme theme--mercado app-loader--default artdeco windows" lang="pt"><head>
<script nonce="">!function(i,n)(void 0!=i.addEventListener&&void 0==i.hidden&&(n.liVisibilityChangeListener=function(){i.hidden&&(n.liHasWindowHidden=!0),i.addEventListener("visibilitychange",n.liVisibilityChangeListener)})(document>window);</script>
<meta content="script-src attr 'none' require-trusted-types-for 'script'; trusted-types 'allow-duplicates' default jSecure highcharts dompurify" data-disposition="enforce" data-report-to="https://www.linkedin.com/security/csp?a=voyager-web&m=bpr" data-sanitizer="jSecure" http-equiv="Content-Security-Policy" name="trusted-types"/>
<title>(15) Idiomas | Larissa Akemi Iuki | LinkedIn</title>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta class="mercado-icons-sprite" content="https://static.licdn.com/aero-v1/sc/h/2brfqc2ezdc100afxfrykl5mid" id="artdeco-icons/static/images/sprite-asset" name="asset-url"/>
<meta content="" name="description"/>
<meta content="notranslate" name="google"/>
<meta content="voyager-web" name="service"/>
<meta content="#ffffff" id="theme-color-meta-tag" name="theme-color"/>
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0" name="viewport"/>
<meta content="https://static.licdn.com/aero-v1/sc/h/" name="baseCDNUrl"/>
<link href="https://static.licdn.com/aero-v1/sc/h/5bukxbhy9xsil5mb7c2wulfbx" id="favicon-svg" rel="icon" type="image/svg+xml"/>
```

```
[109]: # Iniciar a lista de idiomas
idiomas = []

# Buscar todas as seções de idioma corretamente
idiomas_secoes = soup_language.find_all('div', class_='display-flex flex-wrap align-items-center full-height')

for secao in idiomas_secoes:
    # Tentar extrair o nome do idioma
    idioma_nome_div = secao.find('div', class_='display-flex align-items-center mr1 t-bold')
    if idioma_nome_div:
        # Pegue o texto de 'aria-hidden' para evitar duplicações
        idioma_nome = idioma_nome_div.find('span', {'aria-hidden': 'true'}).get_text(strip=True)
    else:
        idioma_nome = 'Nome não encontrado'

    # Tentar extrair o nível do idioma
    idioma_nivel_span = secao.find_next('span', class_='t-14 t-normal t-black--light')
    if idioma_nivel_span:
        # Pegue o texto de 'aria-hidden' para o nível
        idioma_nivel = idioma_nivel_span.find('span', {'aria-hidden': 'true'}).get_text(strip=True)
    else:
        idioma_nivel = 'Nível não encontrado'

    # Adicionar à lista de idiomas se encontrado
    if idioma_nome != 'Nome não encontrado' and idioma_nivel != 'Nível não encontrado':
        idiomas.append((idioma_nome, idioma_nivel))

# Exibir o resultado
for idioma, nivel in idiomas:
    print(f'Idioma: {idioma}, Nível: {nivel}')


Idioma: Inglês, Nível: Nível básico a intermediário
Idioma: Japonês, Nível: Nível básico
Idioma: Português, Nível: Fluente ou nativo
```

Click to add a cell.