



RegEx

Prof. Osvaldo Kotaro Takai
Profa. Ana Cristina dos Santos

Objetivos

Acompanhe, a seguir, os objetivos de aprendizagem:

- Entender que os Bots, às vezes, precisam buscar por padrões de texto utilizando Expressões Regulares.
- Recomendo:
 - [Expressões Regulares do Zero](#)
 - Para aprender a construir expressões regulares.
 - <https://regexr.com/>:
 - Para testar suas expressões regulares.
 - Use o browser [Firefox](#).

O que vamos fazer?

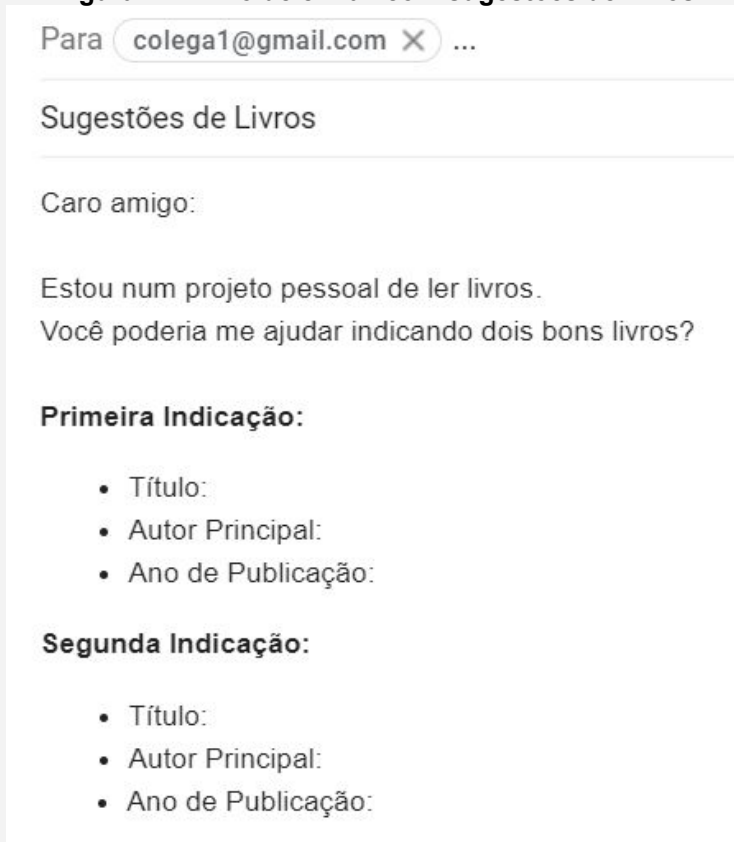
- A ideia é criar um Bot que:
 - Obtenha indicações de livros realizadas por seus colegas e guardá-las num arquivo texto.

Como fazer?

Preparação: enviar e receber e-mails.

- Usando a sua nova conta do Gmail, envie o e-mail ao lado para os seus colegas.
- Retorne os e-mails recebidos de seus colegas indicando dois livros seguindo os campos sugeridos.

Figura 1 - Envio de e-mail com sugestões de livros



Para colega1@gmail.com X ...

Sugestões de Livros

Caro amigo:

Estou num projeto pessoal de ler livros.
Você poderia me ajudar indicando dois bons livros?

Primeira Indicação:

- Título:
- Autor Principal:
- Ano de Publicação:

Segunda Indicação:

- Título:
- Autor Principal:
- Ano de Publicação:

Fonte: do autor, 2021.

Como fazer?

RegEx: entendendo os RegEx que serão utilizados.

- Você receberá e-mails da forma ao lado.
- Note que a parte do e-mail dentro do retângulo não contém indicações de livros. Portanto, devemos eliminá-lo.
- Assim, precisamos encontrar um padrão para dizer:
 - A partir deste ponto do texto, não nos interessa, e queremos eliminá-los.

Figura 2 - Verificação das informações do e-mail

Caro amigo:

Seguem as minhas indicações:

Primeira Indicação:

- Título: Pipeline de liderança
- Autor Principal: CHARAN, Ram
- Ano de Publicação: 2018

Segunda Indicação:

- Título: Organizações exponenciais: Porque elas são 10 vezes melhores, mais rápidas e mais baratas que a sua (e o que fazer a respeito)
- Autor Principal: ISMAIL, S
- Ano de Publicação: 2018

Em dom., 15 de dez. de 2019 às 10:52, fulano beltrano da silva <fulanob.silva.rpa@gmail.com> escreveu:

> Caro amigo:

>

> Estou num projeto pessoal de ler livros em 2020.

> Você poderia me ajudar indicando dois bons livros?

>

> *Primeira Indicação:*

>

- > - Título:
- > - Autor Principal:
- > - Ano de Publicação:

>

> *Segunda Indicação:*

>

- > - Título:
- > - Autor Principal:
- > - Ano de Publicação:

Fonte: do autor, 2021.

Como fazer?

- A palavra “Em” não pode ser esse padrão, pois pode ter nomes de livros que tenham “Em”.
- A data não pode também, pois ela é muito específica.
- Escolhemos “osvaldo takai”, neste caso.
 - No seu caso, o seu nome.

Figura 2 - Verificação das informações do e-mail

Caro amigo:

Seguem as minhas indicações:

Primeira Indicação:

- Título: Pipeline de liderança
- Autor Principal: CHARAN, Ram
- Ano de Publicação: 2018

Segunda Indicação:

- Título: Organizações exponenciais: Porque elas são 10 vezes melhores, mais rápidas e mais baratas que a sua (e o que fazer a respeito)
- Autor Principal: ISMAIL, S
- Ano de Publicação: 2018

Em dom., 15 de dez. de 2019 às 10:52, fulano beltrano da silva <fulanob.silva.rpa@gmail.com> escreveu:

> Caro amigo:

>

> Estou num projeto pessoal de ler livros em 2020.

> Você poderia me ajudar indicando dois bons livros?

>

> ***Primeira Indicação:***

>

> - Título:

> - Autor Principal:

> - Ano de Publicação:

>

> ***Segunda Indicação:***

>

> - Título:

> - Autor Principal:

> - Ano de Publicação:

Fonte: do autor, 2021.

Como fazer?

- O RegEx para fazer isso é:
(fulano beltrano da silva(.|\s)*)
- O RegEx diz o seguinte:
 - Busque no texto “fulano beltrano da silva” seguido de qualquer de caractere “(.|\s)*”.
 - “.” indica caractere normal.
 - “\s” indica caractere em branco.
 - “*” indica zero ou mais vezes.
- Os caracteres encontrados serão substituídos por uma string vazia “”.

Figura 2 - Verificação das informações do e-mail

Caro amigo:

Seguem as minhas indicações:

Primeira Indicação:

- Título: Pipeline de liderança
- Autor Principal: CHARAN, Ram
- Ano de Publicação: 2018

Segunda Indicação:

- Título: Organizações exponenciais: Porque elas são 10 vezes melhores, mais rápidas e mais baratas que a sua (e o que fazer a respeito)
- Autor Principal: ISMAIL, S
- Ano de Publicação: 2018

Em dom., 15 de dez. de 2019 às 10:52, fulano beltrano da silva <fulanob.silva.rpa@gmail.com> escreveu:

> Caro amigo:

>

> Estou num projeto pessoal de ler livros em 2020.

> Você poderia me ajudar indicando dois bons livros?

>

> ***Primeira Indicação:***

>

> - Título:

> - Autor Principal:

> - Ano de Publicação:

>

> ***Segunda Indicação:***

>

> - Título:

> - Autor Principal:

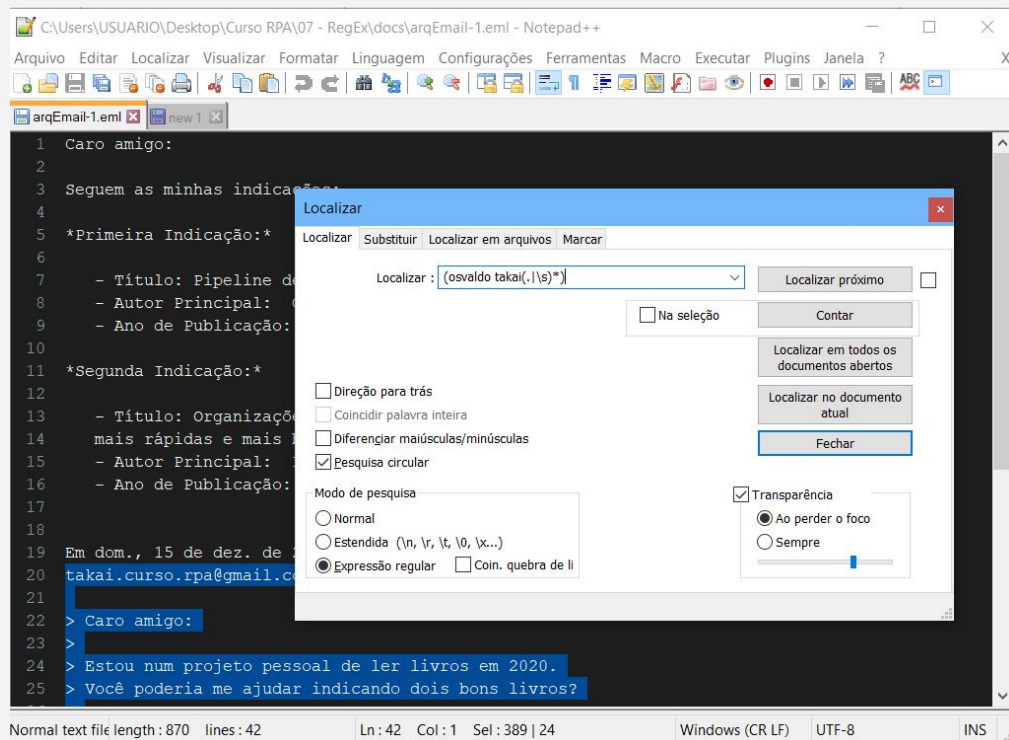
> - Ano de Publicação:

Fonte: do autor, 2021.

Como fazer?

- Teste usando o notepad++ ou no <https://regexr.com/> (Firefox).

Figura 3 - Teste no Notepad++



Fonte: do autor, 2021.

Figura 4 - Teste no regexr.com



Fonte: do autor, 2021.

Como fazer?

- Agora, do texto que sobrou, queremos eliminar todos os caracteres de mudança de linhas “\r” e “\n” para conseguir extrair títulos longos e que possuam mudanças de linhas, como no título da **segunda indicação**. 2
- Queremos também, eliminar os “*” para extrair somente os dados dos campos necessários.

Figura 5 - E-mail

Caro amigo:

Seguem as minhas indicações:

Primeira Indicação:

- Título: ***Pipeline de liderança***
- Autor Principal: ***CHARAN, Ram***
- Ano de Publicação: ***2018***

Segunda Indicação:

- Título: **Organizações exponenciais: Porque elas são 10 vezes melhores, mais rápidas e mais baratas que a sua (e o que fazer a respeito)** 2
- Autor Principal: ISMAIL, S
- Ano de Publicação: 2018

Em dom., 15 de dez. de 2019 às 10:52,

Fonte: do autor, 2021.

Como fazer?

- O RegEx para fazer isso é:
(\r|\n|*)
- O “*” indica o caracteres “*” para não confundir com a indicação de zero ou mais vezes.
- Os caracteres encontrados serão substituídos por uma *string* vazia “”.

Caro amigo:Seguem as minhas indicações:Primeira Indicação: -
Título: Pipeline de liderança - Autor Principal: CHARAN, Ram - Ano
de Publicação: 2018Segunda Indicação: - Título: Organizações
exponenciais: Porque elas são 10 vezes melhores, mais rápidas e
mais baratas que a sua (e o que fazer a respeito) - Autor Principal:
ISMAIL, S - Ano de Publicação: 2018Em dom., 15 de dez. de 2019
às 10:52,

Teste usando o notepad++ ou no <https://regexr.com/> (Firefox).

Como fazer?

- Queremos limpar o texto removendo os múltiplos espaços em branco.

Caro amigo:Seguem as minhas indicações:Primeira Indicação: -
Título: Pipeline de liderança - Autor Principal: CHARAN, Ram - Ano
de Publicação: 2018Segunda Indicação: - Título: Organizações
exponenciais: Porque elas são 10 vezes melhores, mais rápidas e
mais baratas que a sua (e o que fazer a respeito) - Autor Principal:
ISMAIL, S - Ano de Publicação: 2018Em dom., 15 de dez. de 2019
às 10:52,

Como fazer?

- O RegEx para isso é:
 \s+
 - “\s” indica espaço em branco.
 - “+” indica uma ou mais vezes.
- O texto encontrado será substituído por um caractere branco “ ”.

Caro amigo:Seguem as minhas indicações:Primeira Indicação: -
Título: Pipeline de liderança - Autor Principal: CHARAN, Ram - Ano de
Publicação: 2018Segunda Indicação: - Título: Organizações
exponenciais: Porque elas são 10 vezes melhores, mais rápidas e
mais baratas que a sua (e o que fazer a respeito) - Autor Principal:
ISMAIL, S - Ano de Publicação: 2018Em dom., 15 de dez. de 2019 às
10:52,

Teste usando o notepad++ ou no <https://regexr.com/> (Firefox).

Como fazer?

- Finalmente vamos extrair todos os Títulos encontrados no texto.
 - O RegEx para isso é:
Título:\s*(.+?)\s*-\sAutor
- Ou seja:
 - Buscar por todas as ocorrências de “Título:”;
 - Seguidos por zero ou mais caracteres em branco: “\s*”;
 - Seguidos por um ou mais caracteres que conterá o título: “(.+?)”;
 - “?” Indica lazy ao invés de greedy.
- Finalizado por “\s*-\sAutor)”

Caro amigo:Seguem as minhas indicações:Primeira Indicação: -
Título: Pipeline de liderança - Autor Principal: CHARAN, Ram - Ano
de Publicação: 2018Segunda Indicação: - **Título: Organizações
exponenciais: Porque elas são 10 vezes melhores, mais rápidas e
mais baratas que a sua (e o que fazer a respeito) - Autor** Principal:
ISMAIL, S - Ano de Publicação: 2018Em dom., 15 de dez. de 2019 às
10:52,

Teste usando o notepad++ ou no <https://regexr.com/> (Firefox).

Como fazer?

- O mesmo para extrair todos os Autores Principais encontrados.
 - O RegEx para isso é:
Autor Principal:\s*(.+?)\s*-\sAno
 - Ou seja:
 - Buscar por todas as ocorrências de “Autor Principal:”;
 - Seguidos por zero ou mais caracteres em branco: “\s*”;
 - Seguidos por um ou mais caracteres que conterà o autor principal: “(.+?)”;
 - “?” Indica lazy ao invés de greedy.
 - Finalizado por “\s*-\sAno”

Caro amigo:Seguem as minhas indicações:Primeira Indicação: -
Título: Pipeline de liderança - Autor Principal: **CHARAN, Ram** - Ano de
Publicação: 2018Segunda Indicação: - Título: Organizações
exponenciais: Porque elas são 10 vezes melhores, mais rápidas e
mais baratas que a sua (e o que fazer a respeito) - Autor Principal:
ISMAIL, S - Ano de Publicação: 2018Em dom., 15 de dez. de 2019 às
10:52,

Teste usando o notepad++ ou no <https://regexr.com/> (Firefox).

Como fazer?

- Similar para extrair todos os Anos de Publicação encontrados.
 - O RegEx para isso é:
Ano de Publicação:\s*(\d{4})
 - Ou seja:
 - Buscar por todas as ocorrências de “Ano de Publicação:”;
 - Seguidos por zero ou mais caracteres em branco: “\s*”;
 - Seguidos por quatro dígitos que conterá o ano de publicação: “(\d{4})”.

Caro amigo:Seguem as minhas indicações:Primeira Indicação: -
Título: Pipeline de liderança - Autor Principal: CHARAN, Ram - Ano de
Publicação: **2018**Segunda Indicação: - Título: Organizações
exponenciais: Porque elas são 10 vezes melhores, mais rápidas e
mais baratas que a sua (e o que fazer a respeito) - Autor Principal:
ISMAIL, S - Ano de Publicação: **2018**Em dom., 15 de dez. de 2019 às
10:52,

Teste usando o notepad++ ou no <https://regexr.com/> (Firefox).

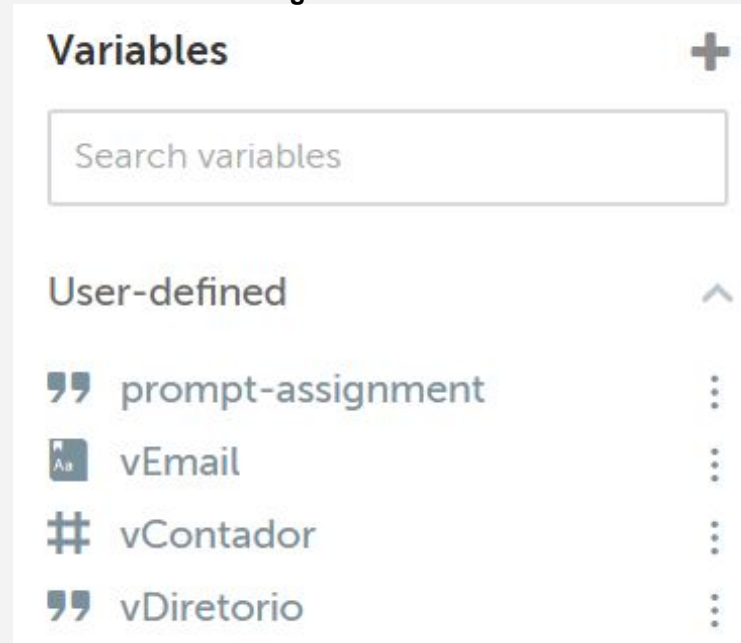
Como fazer?

Os Bots

- Crie três Bots:
 - Bot 1: Que guarda os e-mails com as sugestões de livros como um **arquivo** texto num **diretório** especificado por você.
 - Bot 2: Que processa esses arquivos textos e extraia as **indicações** de livros no seguinte formato:
 - Título/tAutor Principal/tAno/n
 - \t representa o caractere de tabulação (TAB)
 - \n representa o caractere *line feed* (ENTER)
 - Bot 3: Que executa Bot 1 e depois Bot 2.

Bot 1: Variáveis

Figura 6 - Variáveis



Fonte: do autor, 2021.

Bot 1: Variáveis

Variável vEmail

- Usado para armazenar os conteúdos de um e-mail.

Figura 7 - vEmail

The screenshot shows the 'Edit variable' dialog box for a variable named 'vEmail'. The dialog has a title bar 'Edit variable' and two buttons: 'Cancel' and 'Apply'. It contains several fields and options:

- Type:** A dropdown menu set to 'Dictionary'.
- Subtype:** A dropdown menu set to 'String'.
- Name:** A text field containing 'vEmail' with a character limit of 50.
- Description (optional):** A text field containing 'An unordered group of key/value pairs' with a character limit of 255.
- Use as input:** An unchecked checkbox.
- Use as output:** An unchecked checkbox.
- Constant (read-only):** An unchecked checkbox.
- Default value (optional):** A text field containing 'This dictionary is empty'.
- Bottom bar:** A light blue bar with a plus sign icon.

Fonte: do autor, 2021.

Bot 1: Variáveis

Variável vContador

- Usado para numerar o arquivo texto:
 - **arqEmail-1.eml** para o primeiro e-mail.
 - **arqEmail-2.eml** para o segundo e-mail e assim por diante.

Figura 8 - vContador

The screenshot shows the 'Edit variable' dialog box for a variable named 'vContador'. The 'Type' is set to 'Number'. The 'Name' field contains 'vContador' with a character limit of 50. The 'Description (optional)' field contains 'Digits' with a character limit of 255. There are three checkboxes: 'Use as input' (unchecked), 'Use as output' (unchecked), and 'Constant (read-only)' (unchecked). The 'Default value (optional)' field contains '0' with increment and decrement buttons.

Edit variable	
Type	# Number
Name	vContador (Max characters = 50)
Description (optional)	Digits (Max characters = 255)
<input type="checkbox"/> Use as input	
<input type="checkbox"/> Use as output	
<input type="checkbox"/> Constant (read-only)	
Default value (optional)	0 (+/-)

Fonte: do autor, 2021.

Bot 1: Variáveis

Variável **vDiretorio**

- Guarda o caminho completo para o diretório dos arquivos textos:
 - Conteúdos dos e-mails e
 - Conteúdo das indicações.

Figura 8 - vDiretorio

The screenshot shows the 'Edit variable' dialog box for a variable named 'vDiretorio'. The 'Type' is set to 'String'. The 'Name' field contains 'vDiretorio' with a note 'Max characters = 50'. The 'Description (optional)' field contains 'Text' with a note 'Max characters = 255'. There are three checkboxes: 'Use as input' (unchecked), 'Use as output' (unchecked), and 'Constant (read-only)' (unchecked). The 'Default value' field contains the path 'C:\Users\USUARIO\Desktop\Curso RPA\07 - RegEx\docs'.

Edit variable Cancel Apply

Type
String

Name
vDiretorio
Max characters = 50

Description (optional)
Text
Max characters = 255

☐ Use as input

☐ Use as output

☐ Constant (read-only)

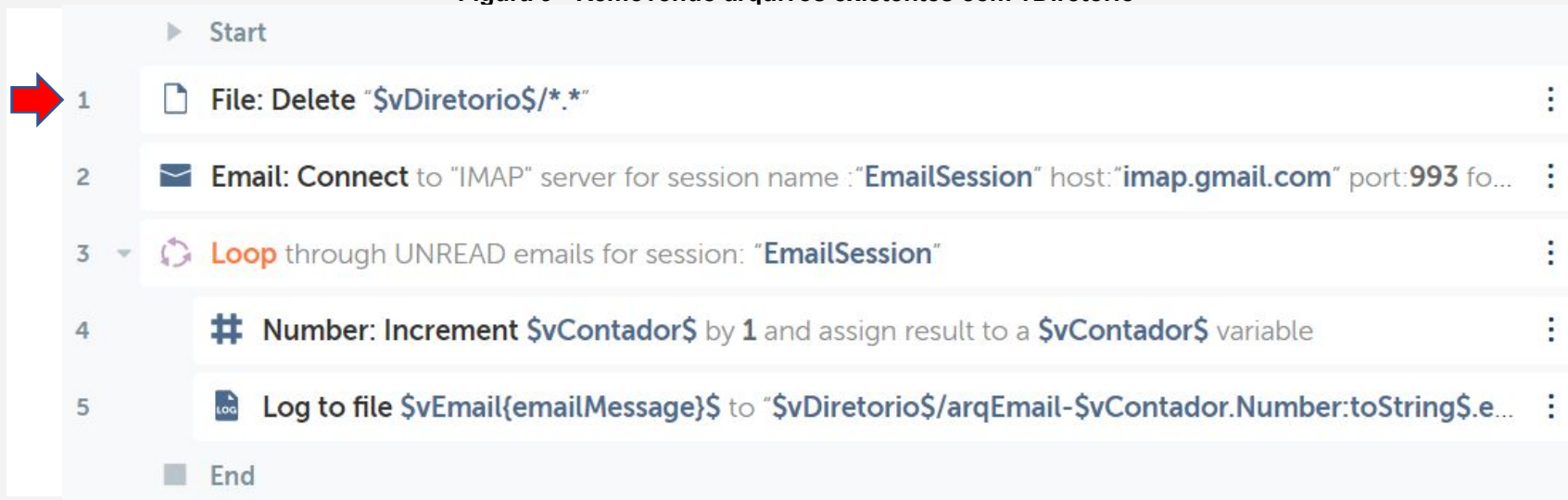
Default value
C:\Users\USUARIO\Desktop\Curso RPA\07 - RegEx\docs

Fonte: do autor, 2021.

Bot 1: Ação 1

- Remove quaisquer arquivos existentes no diretório especificado pela variável vDiretorio.

Figura 9 - Removendo arquivos existentes com vDiretorio



Fonte: do autor, 2021.

Bot 1: Ação 1

Ação File: Delete

- A versão atual desta ação exige que se defina o filtro de Data: Created que, no caso foi definida com “Is within last (days)” = 1.
- Ou seja, o arquivo que será removido foi criado há 1 dia.

Figura 10 - File: Delete

File: Delete
Deletes a file

File
\$vDiretorio\$/*.* (x) Browse...
e.g. C:\MyDoc*.doc

☐ Size
Size (KB)
##

☒ Date
Created ▼

☒ Is within last (days)
1 (x)

☐ Is between
Start date (MM/DD/YY)

End date (MM/DD/YY)
##

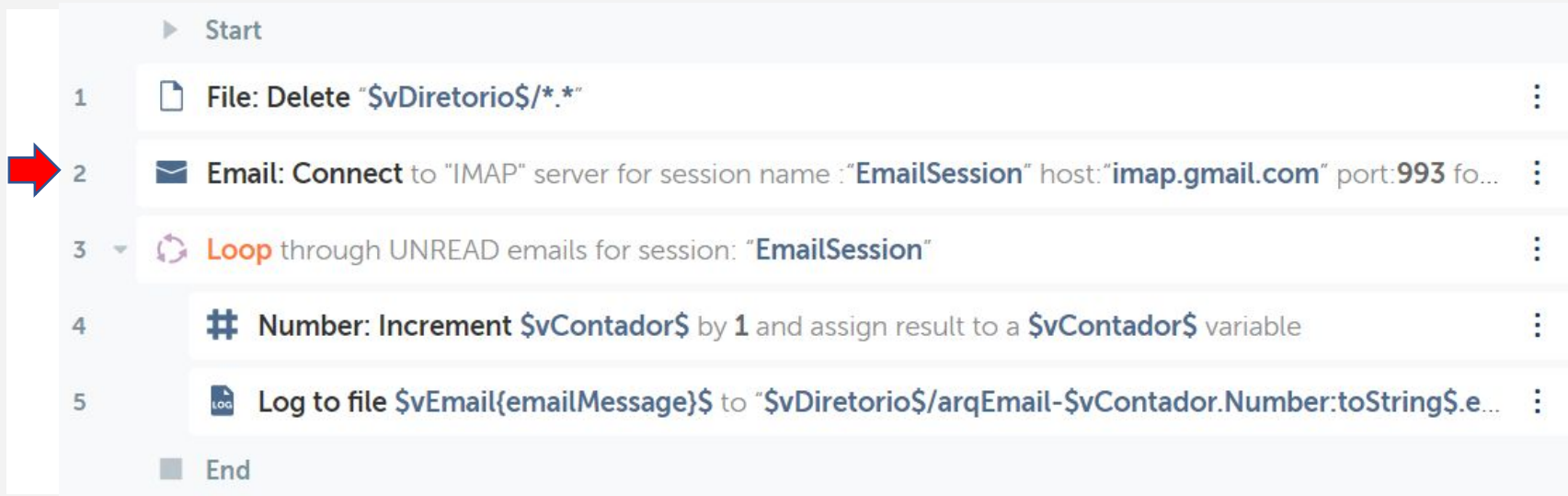
☐ Is before (MM/DD/YY)

Fonte: do autor, 2021.

Bot 1: Ação 2

- Conecta à conta do Gmail.
 - Lembre-se da configuração desta ação aprendida na lição 05 - *Consultar E-mail*.
- No nome da sessão criada é **EmailSession**.

Figura 11 - Email: Connect

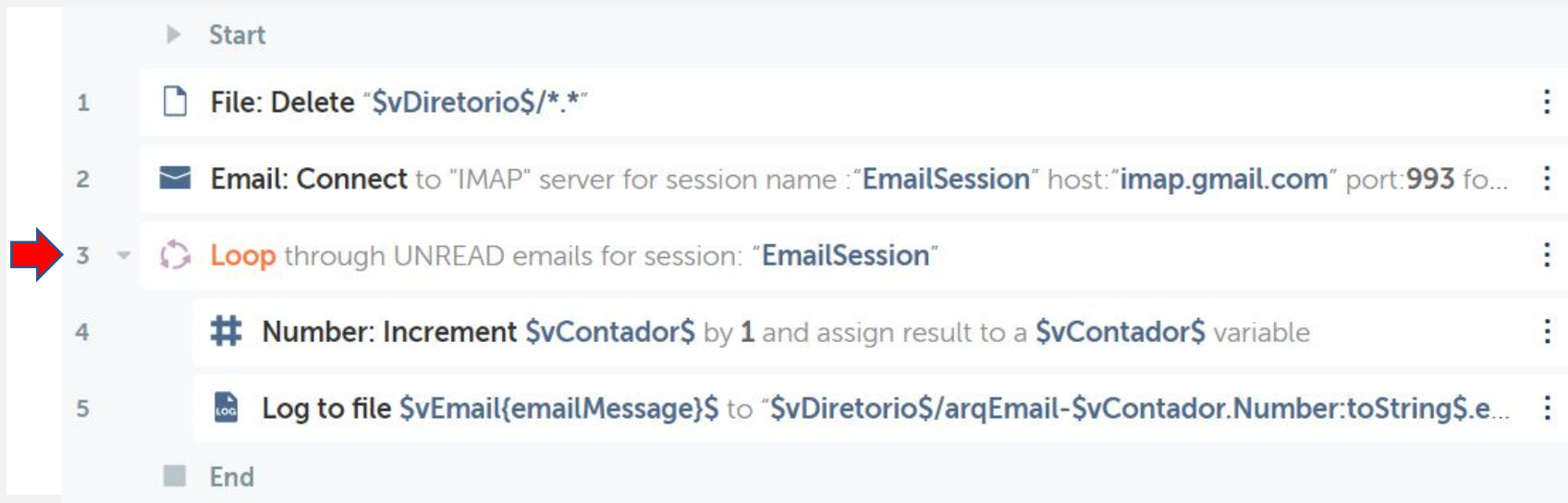


Fonte: do autor, 2021.

Bot 1: Ação 3

- Permite visitar todos os e-mails não lidos existentes na caixa de entrada (*inbox*), cujo assunto seja “Sugestões de Livros”.
- Garanta que o nome da sessão seja a mesma da ação 2: **EmailSession**.

Figura 12 - Loop

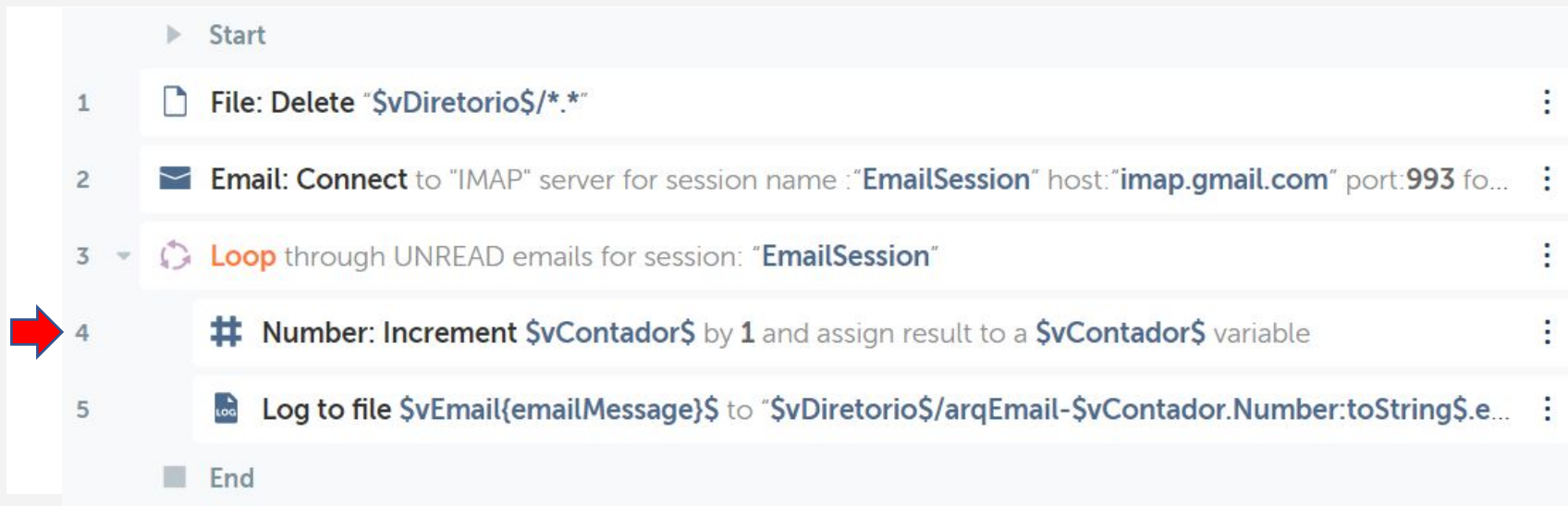


Fonte: do autor, 2021.

Bot 1: Ação 4

- Incrementa a variável `vContador`, para que o primeiro arquivo texto seja `arqEmail-1.eml`.

Figura 13 -Number: Increment

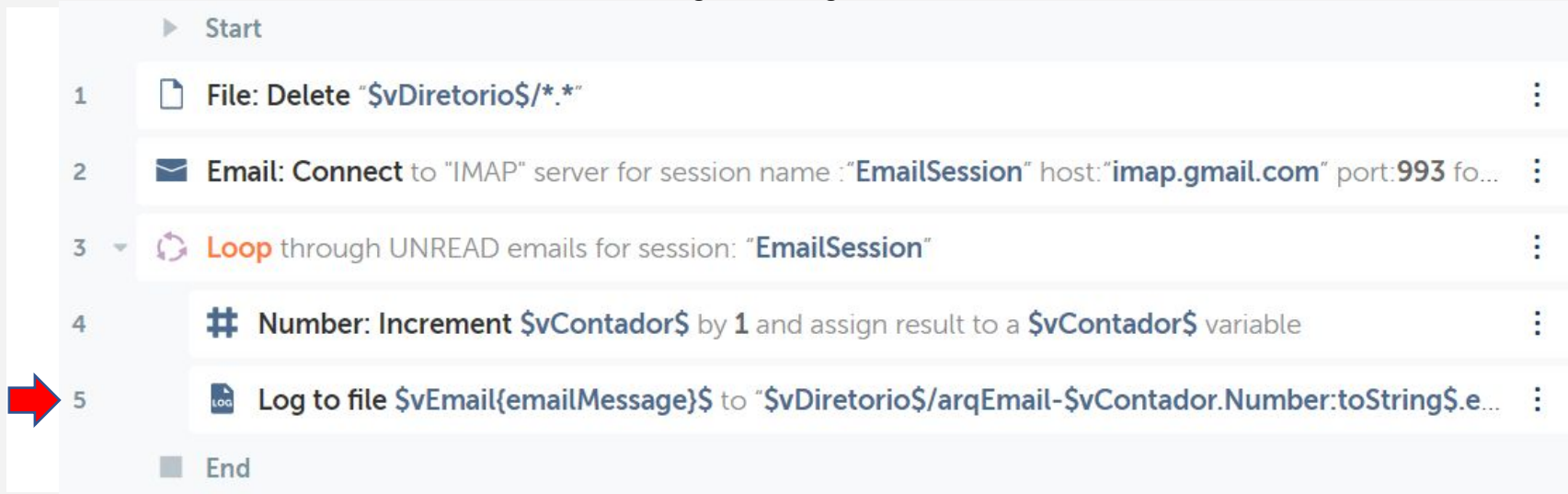


Fonte: do autor, 2021.

Bot 1: Ação 5

- Salva a mensagem do e-mail no arquivo texto **arqEmail-*.eml** no diretório **vDiretorio**.

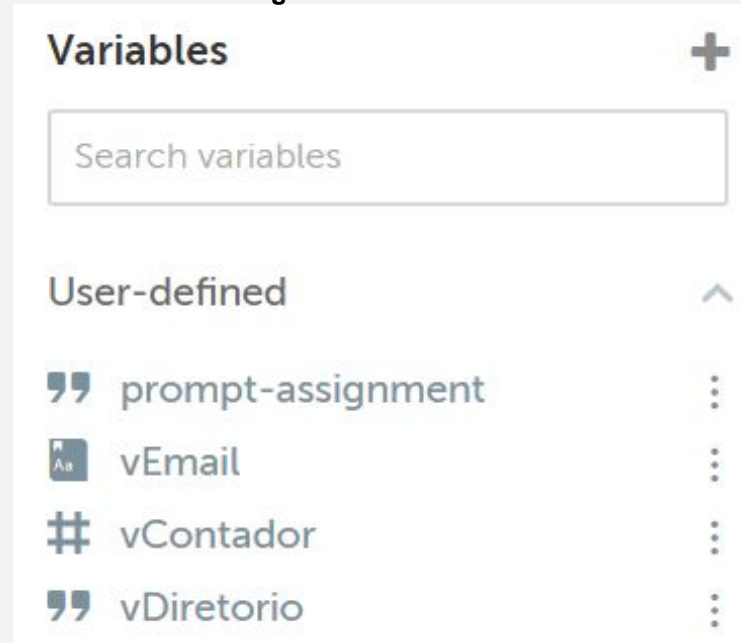
Figura 13 -Log to file



Fonte: do autor, 2021.

Bot 2: Variáveis

Figura 14 - Variáveis 2



Fonte: do autor, 2021.

Bot 2: Variável vDiretorio

- Caminho completo para o diretório dos arquivos textos:
 - Conteúdos dos e-mails (criado pelo **Bot 1**) e
 - Conteúdo das indicações (que será criado por este bot, **Bot 2**).

Figura 15 - vDiretorio 2

The screenshot shows a web-based 'Edit variable' interface. At the top right are 'Close' and 'Apply' buttons. The 'Name' field contains 'vDiretorio'. The 'Description (optional)' field is empty. There are three checkboxes: 'Constant (read-only)' (unchecked), 'Use as input' (unchecked), and 'Use as output' (unchecked). The 'Type' dropdown menu is set to 'String'. The 'Default value (optional)' field contains the path 'C:\Users\USUARIO\Desktop\Curso RPA\07 - RegEx\docs'.

Edit variable Close Apply

Name
vDiretorio

Description (optional)

☐ Constant (read-only)

☐ Use as input ☐ Use as output

Type
String ▼

Default value (optional)
C:\Users\USUARIO\Desktop\Curso RPA\07 - RegEx\docs

Fonte: do autor, 2021.

Bot 2: Ação 1

- Define em Python, o código que visita cada **arqEmail-*.eml** no diretório **vDiretorio** e extrai os Títulos, Autores Principais e Anos de Publicação dos livros indicados.

Figura 16 - Python script: Open



Fonte: do autor, 2021.

Bot 2: Script em Python

```
# -*- coding: utf-8  
import re  
import os
```

- Especifica que o Script em Python irá trabalhar com a codificação utf-8.
- Importa a biblioteca de RegEx.
- Importa a biblioteca para manipular arquivos no sistema operacional.

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):
```

```
    indicacoes = ""
```

```
    for arqEmail in os.listdir(diretorio):
```

```
        if arqEmail.endswith(".eml"):
```

```
            arquivo = open(diretorio + "\\ " + arqEmail, 'r', encoding='utf8')
```

```
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)
```

```
            arquivo.close()
```

```
arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')
```

```
arquivoIndicacoes.write(indicacoes)
```

```
arquivoIndicacoes.close()
```

- **criarArqDeIndicacoes** é o nome da função que será chamada pelo Bot 2.
- **diretorio** é o parâmetro desta função e contém o caminho completo do diretório que contém os arquivos textos.

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):
```

```
    indicacoes = ""
```

```
    for arqEmail in os.listdir(diretorio):
```

```
        if arqEmail.endswith(".eml"):
```

```
            arquivo = open(diretorio + "\\ " + arqEmail, 'r', encoding='utf8')
```

```
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)
```

```
            arquivo.close()
```

```
arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')
```

```
arquivoIndicacoes.write(indicacoes)
```

```
arquivoIndicacoes.close()
```

- A variável string, indicações, é criada para conter todas as indicações de livros, ou seja, várias linhas da forma:
Título\tAutor Principal\tAno de Publicação\n

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):
```

```
    indicacoes = ""
```

```
    for arqEmail in os.listdir(diretorio):
```

```
        if arqEmail.endswith(".eml"):
```

```
            arquivo = open(diretorio + "\\\" + arqEmail, 'r', encoding='utf8')
```

```
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)
```

```
            arquivo.close()
```

```
arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')
```

```
arquivoIndicacoes.write(indicacoes)
```

```
arquivoIndicacoes.close()
```

- É um loop que executa, o seu *bloco de comandos* várias vezes, uma vez para cada arquivo encontrado no **diretório**.

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):
```

```
    indicacoes = ""
```

```
    for arqEmail in os.listdir(diretorio):
```

```
        if arqEmail.endswith(".eml"):
```

```
            arquivo = open(diretorio + "\\\" + arqEmail, 'r', encoding='utf8')
```

```
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)
```

```
            arquivo.close()
```

```
arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')
```

```
arquivoIndicacoes.write(indicacoes)
```

```
arquivoIndicacoes.close()
```

- Se o arquivo encontrado finalizar com a extensão “.eml” então executa o *bloco de comandos* abaixo.

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):  
    indicacoes = ""  
    for arqEmail in os.listdir(diretorio):  
        if arqEmail.endswith(".eml"):  
            arquivo = open(diretorio + "\\\" + arqEmail, 'r', encoding='utf8')  
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)  
            arquivo.close()  
  
    arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')  
    arquivoIndicacoes.write(indicacoes)  
    arquivoIndicacoes.close()
```

- Abre o arquivo para leitura na codificação utf-8.

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):
```

```
    indicacoes = ""
```

```
    for arqEmail in os.listdir(diretorio):
```

```
        if arqEmail.endswith(".eml"):
```

```
            arquivo = open(diretorio + "\\ " + arqEmail, 'r', encoding='utf8')
```

```
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)
```

```
            arquivo.close()
```

```
arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')
```

```
arquivoIndicacoes.write(indicacoes)
```

```
arquivoIndicacoes.close()
```

- Chama a função **obterLivros** que retorna todas as indicações de livros contidas no arquivo recém aberto.
- As indicações retornadas são concatenadas às indicações anteriores: (+=).

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):  
    indicacoes = ""  
    for arqEmail in os.listdir(diretorio):  
        if arqEmail.endswith(".eml"):  
            arquivo = open(diretorio + "\\\" + arqEmail, 'r', encoding='utf8')  
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)  
            arquivo.close()  
  
    arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')  
    arquivoIndicacoes.write(indicacoes)  
    arquivoIndicacoes.close()
```

- Fecha o arquivo recém aberto.

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):  
    indicacoes = ""  
    for arqEmail in os.listdir(diretorio):  
        if arqEmail.endswith(".eml"):  
            arquivo = open(diretorio + "\\\" + arqEmail, 'r', encoding='utf8')  
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)  
            arquivo.close()
```

```
arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')  
arquivoIndicacoes.write(indicacoes)  
arquivoIndicacoes.close()
```

- Cria ou recria um arquivo texto com o nome **indicações.txt** vazio no formato utf-8 no **diretório** especificação.
 - utf-8 é a melhor codificação para o alfabeto latino.

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):  
    indicacoes = ""  
    for arqEmail in os.listdir(diretorio):  
        if arqEmail.endswith(".eml"):  
            arquivo = open(diretorio + "\\\" + arqEmail, 'r', encoding='utf8')  
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)  
            arquivo.close()  
  
    arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')  
    arquivoIndicacoes.write(indicacoes)  
    arquivoIndicacoes.close()
```

- Grava no arquivo **indicações.txt** as indicações contidas na variável **indicações**.

Bot 2: Script em Python

```
def criarArqDeIndicacoes(diretorio):  
    indicacoes = ""  
    for arqEmail in os.listdir(diretorio):  
        if arqEmail.endswith(".eml"):  
            arquivo = open(diretorio + "\\\" + arqEmail, 'r', encoding='utf8')  
            indicacoes += obterLivros(arquivo.read(), diretorio, arqEmail)  
            arquivo.close()  
  
    arquivoIndicacoes = open(diretorio + "\\indicacoes.txt", 'w+', encoding='utf8')  
    arquivoIndicacoes.write(indicacoes)  
    arquivoIndicacoes.close()
```

- Fecha o arquivo **indicacoes.txt**.

Bot 2: Script em Python

```
def obterLivros(fonte, diretorio, arqEmail):
```

```
    fonte = re.sub("(osvaldo takai(.\s)*)", "", fonte) # Remove texto desnecessário.
```

```
    fonte = re.sub("(\\n|\\s*)", "", fonte) # Elimina os "pula linhas" e "".
```

```
    fonte = re.sub("\\s+", " ", fonte) # Remover múltiplos espaços.
```

```
    # Apenas para depuração
```

```
    arquivoAux = open(diretorio + "\\ " + arqEmail + "-DEBUG.txt", 'w+', encoding='utf8')
```

```
    arquivoAux.write(fonte)
```

```
    arquivoAux.close()
```

```
    aux1 = obterLista(fonte, "Título:\\s*(.+?)\\s*\\sAutor") # Obtém os Títulos.
```

```
    aux2 = obterLista(fonte, "Autor Principal:\\s*(.+?)\\s*\\sAno") # Obtém os Principais.
```

```
    aux3 = obterLista(fonte, "Ano de Publicação:\\s*(\\d{4})") # Obtém os Anos.
```

```
    livros = ""
```

```
    for i in range(len(aux1)):
```

```
        livros += aux1[i].strip() + "\\t" + aux2[i].strip() + "\\t" + aux3[i].strip() + "\\n"
```

```
    return livros
```

- Define a função interna **obterLivros**.
- Recebe como parâmetros:
 - fonte (conteúdo do arquivo texto do e-mail lido)
 - Diretorio (para guardar o arquivo de depuração)
 - arqEmail (nome do arquivo texto do e-mail lido para depuração)

Bot 2: Script em Python

```
def obterLivros(fonte, diretorio, arqEmail):
```

```
    fonte = re.sub("(osvaldo takai(?:\s*))", "", fonte) # Remove texto desnecessário.
```

```
    fonte = re.sub("(\\n|\\s*)", "", fonte) # Elimina os "pula linhas" e "".
```

```
    fonte = re.sub("\\s+", " ", fonte) # Remover múltiplos espaços.
```

```
# Apenas para depuração
```

```
arquivoAux = open(diretorio + "\\\" + arqEmail + "-DEBUG.txt", 'w+', encoding='utf8')
```

```
arquivoAux.write(fonte)
```

```
arquivoAux.close()
```

```
aux1 = obterLista(fonte, "Título:\\s*(.+?)\\s*-\\sAutor") # Obtém os Títulos.
```

```
aux2 = obterLista(fonte, "Autor Principal:\\s*(.+?)\\s*-\\sAno") # Obtém os Principais.
```

```
aux3 = obterLista(fonte, "Ano de Publicação:\\s*(\\d{4})") # Obtém os Anos.
```

```
livros = ""
```

```
for i in range(len(aux1)):
```

```
    livros += aux1[i].strip() + '\\t' + aux2[i].strip() + '\\t' + aux3[i].strip() + '\\n'
```

```
return livros
```

- Utiliza o RegEx para remover da fonte (que contém todo o conteúdo do e-mail) a parte que não contém as indicações de livros.

Bot 2: Script em Python

```
def obterLivros(fonte, diretorio, arqEmail):
    fonte = re.sub("(osvaldo takai(.\s)*)", "", fonte) # Remove texto desnecessário.
    fonte = re.sub("(\\n\\s*)", "", fonte) # Elimina os "pula linhas" e "*".
    fonte = re.sub("\\s+", " ", fonte) # Remover múltiplos espaços.

    # Apenas para depuração
    arquivoAux = open(diretorio + "\\ " + arqEmail + "-DEBUG.txt", 'w+', encoding='utf8')
    arquivoAux.write(fonte)
    arquivoAux.close()

    aux1 = obterLista(fonte, "Título:\\s*(.+?)\\s*-\\sAutor") # Obtém os Títulos.
    aux2 = obterLista(fonte, "Autor Principal:\\s*(.+?)\\s*-\\sAno") # Obtém os Principais.
    aux3 = obterLista(fonte, "Ano de Publicação:\\s*(\\d{4})") # Obtém os Anos.

    livros = ""
    for i in range(len(aux1)):
        livros += aux1[i].strip() + "\\t" + aux2[i].strip() + "\\t" + aux3[i].strip() + "\\n"
    return livros
```

- Remove da fonte os *carriage return*, *line feed* e asteriscos.

Bot 2: Script em Python

```
def obterLivros(fonte, diretorio, arqEmail):
    fonte = re.sub("(osvaldo takai(.\s)*)", "", fonte) # Remove texto desnecessário.
    fonte = re.sub("(\\n|\\s*)", "", fonte) # Elimina os "pula linhas" e "".
    fonte = re.sub("\\s+", " ", fonte) # Remover múltiplos espaços.

    # Apenas para depuração
    arquivoAux = open(diretorio + "\\ " + arqEmail + "-DEBUG.txt", 'w+', encoding='utf8')
    arquivoAux.write(fonte)
    arquivoAux.close()

    aux1 = obterLista(fonte, "Título:\\s*(.+?)\\s*\\sAutor") # Obtém os Títulos.
    aux2 = obterLista(fonte, "Autor Principal:\\s*(.+?)\\s*\\sAno") # Obtém os Principais.
    aux3 = obterLista(fonte, "Ano de Publicação:\\s*(\\d{4})") # Obtém os Anos.

    livros = ""
    for i in range(len(aux1)):
        livros += aux1[i].strip() + "\\t" + aux2[i].strip() + "\\t" + aux3[i].strip() + "\\n"
    return livros
```

- Substitui vários espaços em branco por um único espaço em branco.

Bot 2: Script em Python

```
def obterLivros(fonte, diretorio, arqEmail):
    fonte = re.sub("(osvaldo takai(\\s)*)", "", fonte) # Remove texto desnecessário.
    fonte = re.sub("(\\n|\\s*)", "", fonte) # Elimina os "pula linhas" e "".
    fonte = re.sub("\\s+", " ", fonte) # Remover múltiplos espaços.

    # Apenas para depuração
    arquivoAux = open(diretorio + "\\ " + arqEmail + "-DEBUG.txt", 'w+', encoding='utf8')
    arquivoAux.write(fonte)
    arquivoAux.close()

    aux1 = obterLista(fonte, "Título:\\s*(.+?)\\s*\\s*Autor") # Obtém os Títulos.
    aux2 = obterLista(fonte, "Autor Principal:\\s*(.+?)\\s*\\s*Ano") # Obtém os Principais.
    aux3 = obterLista(fonte, "Ano de Publicação:\\s*(\\d{4})") # Obtém os Anos.

    livros = ""
    for i in range(len(aux1)):
        livros += aux1[i].strip() + "\\t" + aux2[i].strip() + "\\t" + aux3[i].strip() + "\\n"
    return livros
```

- Para verificar que a limpeza anterior funcionou corretamente, este trecho de código salva a fonte resultante no arquivo com o mesmo nome do arqEmail seguido do sufixo “-DEBUG.txt”.

Bot 2: Script em Python

```
def obterLivros(fonte, diretorio, arqEmail):
    fonte = re.sub("(osvaldo takai(\\.\\s)*)", "", fonte) # Remove texto desnecessário.
    fonte = re.sub("(\\n\\s*)", "", fonte) # Elimina os "pula linhas" e "".
    fonte = re.sub("\\s+", " ", fonte) # Remover múltiplos espaços.

    # Apenas para depuração
    arquivoAux = open(diretorio + "\\ " + arqEmail + "-DEBUG.txt", 'w+', encoding='utf8')
    arquivoAux.write(fonte)
    arquivoAux.close()

    aux1 = obterLista(fonte, "Título:\\s*(.+?)\\s*\\sAutor") # Obtém os Títulos.
    aux2 = obterLista(fonte, "Autor Principal:\\s*(.+?)\\s*\\sAno") # Obtém os Principais.
    aux3 = obterLista(fonte, "Ano de Publicação:\\s*(\\d{4})") # Obtém os Anos.

    livros = ""
    for i in range(len(aux1)):
        livros += aux1[i].strip() + "\\t" + aux2[i].strip() + "\\t" + aux3[i].strip() + "\\n"
    return livros
```

- Obtém as listas de Títulos, Autores Principais e Anos de Publicação.

Bot 2: Script em Python

```
def obterLivros(fonte, diretorio, arqEmail):
    fonte = re.sub("(osvaldo takai(.\s)*)", "", fonte) # Remove texto desnecessário.
    fonte = re.sub("(\\n\\s*)", "", fonte) # Elimina os "pula linhas" e "".
    fonte = re.sub("\\s+", " ", fonte) # Remover múltiplos espaços.

    # Apenas para depuração
    arquivoAux = open(diretorio + "\\ " + arqEmail + "-DEBUG.txt", 'w+', encoding='utf8')
    arquivoAux.write(fonte)
    arquivoAux.close()

    aux1 = obterLista(fonte, "Título:\\s*(.+?)\\s*-\\sAutor") # Obtém os Títulos.
    aux2 = obterLista(fonte, "Autor Principal:\\s*(.+?)\\s*-\\sAno") # Obtém os Principais.
    aux3 = obterLista(fonte, "Ano de Publicação:\\s*(\\d{4})") # Obtém os Anos.

    livros = ""
    for i in range(len(aux1)):
        livros += aux1[i].strip() + "\\t" + aux2[i].strip() + "\\t" + aux3[i].strip() + "\\n"
    return livros
```

- Monta a string contendo as indicações de livros no formato:
Título\\tAutor Principal\\tAno de Publicação\\n

Bot 2: Script em Python

```
def obterLivros(fonte, diretorio, arqEmail):
    fonte = re.sub("(osvaldo takai(.\s)*)", "", fonte) # Remove texto desnecessário.
    fonte = re.sub("(\\n\\s*)", "", fonte) # Elimina os "pula linhas" e "".
    fonte = re.sub("\\s+", " ", fonte) # Remover múltiplos espaços.

    # Apenas para depuração
    arquivoAux = open(diretorio + "\\ " + arqEmail + "-DEBUG.txt", 'w+', encoding='utf8')
    arquivoAux.write(fonte)
    arquivoAux.close()

    aux1 = obterLista(fonte, "Título:\\s*(.+?)\\s*-\\sAutor") # Obtém os Títulos.
    aux2 = obterLista(fonte, "Autor Principal:\\s*(.+?)\\s*-\\sAno") # Obtém os Principais.
    aux3 = obterLista(fonte, "Ano de Publicação:\\s*(\\d{4})") # Obtém os Anos.

    livros = ""
    for i in range(len(aux1)):
        livros += aux1[i].strip() + "\\t" + aux2[i].strip() + "\\t" + aux3[i].strip() + "\\n"
    return livros
```

- Retorna a string das indicações montada.

Bot 2: Ação 2

- Executa a função `criarArqDeIndicacoes` passando como parâmetro o diretório `vDiretorio`.

Figura 17 - Python script: Execute function



Fonte: do autor, 2021.

Bot 3

- Este Bot chama o Bot 1 e Bot 2 nesta sequência.

Figura 18 - Bot 3



Fonte: do autor, 2021.

Salve e Execute

- Dúvidas?

A black and white photograph of a person's hands typing on a laptop keyboard. A cup of tea with a tea bag is visible on the desk next to the laptop. The image is split vertically, with the left side being lighter and the right side being darker.

Usando Python

FIM

Faculdade
IMPACTA