

1 - Oi quero perguntar algumas coisas antes de ajustar o código: 1 - Você consulta nossa conversa por inteiro para melhorar nosso robo como por exemplo ver os ultimos scripts que deram certo; 2 - O agente faz sozinho uma interação com o eproc sozinho com base num comando, como por exemplo essa que estamos fazendo, procurar um processo, uma apelação com as palavras chaves, sem precisar de script? 3 = Se o agente faz sozinho, como ele sabe onde "apertar" o botão certo na tela para por exemplo quegar nos localidore, ele consulta os scripts da pagina e se ele consulta, ele não pode registrar isso para aprimorar nosso script do robo? E, na programação pythom é possível separar os blocos de programação com numeração, por exemplo: 1 - bloco do script: 2 - outro bloco do script, assim acho que seria mais fácil para mim que não sou programador achar o bloco que precisa para substituir.

O ChatGPT disse:

2 - 1. Instalar as dependências

1. Abra o **Prompt de Comando** (ou o terminal integrado do VS Code) na pasta onde está o seu projeto.

Se estiver usando um ambiente virtual (recomendado), ative-o:

```
venv\Scripts\activate
```

2.

Execute o comando para instalar as bibliotecas necessárias:

```
pip install selenium requests PyPDF2
```

3. Isso fará o download e a instalação do Selenium (para controlar o navegador), Requests (caso você precise baixar PDFs) e PyPDF2 (para ler o texto dos PDFs ao aplicar filtros por palavras-chave).

2. Ajustar o caminho do ChromeDriver

1. Localize em seu computador onde está o arquivo `chromedriver.exe`. Se você já copiou para `C:\ChromeDriver\`, então esse é o caminho a ser usado.
2. Abra o arquivo `eproc_extractor_gui.py` em um editor de texto (pode ser o VS Code, Notepad ou qualquer outro).

Procure a linha que contém:

```
CHROME_DRIVER_PATH = r"C:\ChromeDriver\chromedriver.exe"
```

3. Se o arquivo está em outro local, substitua esse caminho pela localização correta, mantendo as barras invertidas (\) e o sufixo `.exe`.
4. Salve o arquivo.

3. Executar o script

Volte ao terminal (Prompt de Comando) e navegue até a pasta onde está o script:

```
cd "C:\Users\emanuelamara1\Documents\Emma\Python"
```

1.

Execute o script:

```
python eproc_extractor_gui.py
```

2.

- O Chrome abrirá a página inicial do Eproc.
- Faça o login manualmente (informando usuário e senha).
- Assim que a tela principal do Eproc aparecer, você verá uma janela do programa pedindo:
 1. **Localizador** (digite ou cole exatamente o nome da fila em “Meus localizadores”).
 2. **Tipo de recurso** (Apelação ou Agravo de Instrumento).
 3. **Tipo de peça** (Sentença, Apelação, Contrarrazões ou Parecer).
 4. **Expressão de busca** com os conectores (“casa ou gato” e “piz*”, por exemplo), se desejar filtrar por palavras.
 5. **Aplicar filtro** (marque se quiser que a busca de palavras influencie no download).

- Confirme as opções e aguarde. O robô percorrerá todos os processos da fila selecionada e salvará os PDFs em `C:\recursos` (a pasta é criada automaticamente se ainda não existir).

Seguindo esses passos, o script estará pronto para uso sem precisar alterar o código toda vez que você trocar de localizador, de tipo de documento ou de palavras-chave.

Você disse:

O que é ambiente virtual que fala no passo 2?

O ChatGPT disse:

Quando falei em “ambiente virtual” (ou *virtual environment*, abreviado venv) eu me referia a uma cópia isolada do interpretador Python e de todas as bibliotecas que você instala via `pip`. É uma forma de manter as dependências do projeto separadas do resto do sistema, evitando conflitos entre versões de pacotes diferentes.

Em termos práticos, você faz assim:

Criar o ambiente (só uma vez por projeto):

Abra o prompt de comando dentro da pasta do projeto e digite:

```
python -m venv venv
```

1. Isso cria uma subpasta chamada `venv` contendo o Python e as bibliotecas deste projeto.

Ativar o ambiente (sempre que for trabalhar nesse projeto):

Ainda no prompt, execute:

`venv\Scripts\activate`

2. (em Linux/macOS, seria `source venv/bin/activate`).

Ao ativar, você verá algo como `(venv)` no início da linha; isso indica que o ambiente isolado está em uso. As bibliotecas instaladas com `pip` vão para dentro dessa pasta, sem mexer no Python do sistema.

3. **Usar o Python normalmente:**

Agora instale as dependências com `pip install selenium requests PyPDF2`, ajuste o caminho do ChromeDriver no script e execute `python eproc_extractor_gui.py`. Tudo acontecerá dentro do `venv`.

4. **Desativar quando terminar** (opcional):

No terminal, basta digitar `deactivate` para voltar ao Python “global”.

Usar `venv` é recomendado porque deixa o projeto mais organizado e evita conflitos de versões, mas se você preferir pode instalar as bibliotecas globalmente e rodar o script sem criar um ambiente virtual.

Você disse:



e esse ai que estava no print

O ChatGPT disse:

Trabalhou por 45 segundos

Sim, essa tela é a página inicial do Eproc. Para acessar os localizadores, clique no menu 'Localizadores' à esquerda, depois em 'Meus localizadores' e selecione 'Urgentes Duda'. Isso deve ajudá-lo a navegar adequadamente na interface.

Pronto

Essa tela do “Painel do Magistrado” é a página inicial que aparece após o login. O menu de “Localizadores” está nessa coluna à esquerda (logo abaixo de “Impedimento de Magistrados”). Para chegar ao localizador “Urgentes Duda”, você deve clicar em “**Localizadores**”, depois em “**Meus localizadores**” e, então, selecionar a fila “0.06 Urgentes Duda”. É exatamente esse caminho que o script utiliza automaticamente depois que você faz o login.

Você disse:

Você assumiu o controle

O ChatGPT disse:

Parou depois de 40 segundos

Você disse:

Pelo jeito ele não entrou no menu localizador e depois meus localizadores. Mas porque se no script original ele conseguia?

O ChatGPT disse:

Trabalhou por 2 minutos

Você notou essa diferença porque, na versão mais recente, eu deixei de clicar no menu lateral e passei a abrir diretamente a página de “Meus localizadores” usando aquele link com **hash=...**. É um atalho que costuma funcionar quando o hash está válido, mas pode não levar você exatamente para o mesmo lugar em todas as sessões.

O script original, que você testou antes, de fato navegava pelo menu: clicava em “**Localizadores**”, depois “**Meus localizadores**”, e então selecionava o item “0.06 Urgentes Duda”. Na versão com GUI eu usei o link direto, porque foi mais rápido. Se a sua sessão ou perfil mudou, esse link pode não servir e a rotina deixa de abrir a lista — por isso você voltou a ver a tela inicial e não a página de localizadores.

Para ter mais robustez, basta ajustar o código para fazer os cliques no menu em vez de usar o atalho. Assim ele reproduz exatamente o caminho manual e dispensa qualquer **hash** de URL. Se quiser que eu faça essa alteração, posso atualizar o script para clicar em **Localizadores** → **Meus localizadores** antes de procurar o “Duda” — aí o comportamento volta a ser igual ao do script original que funcionava para você.

Você disse:

Oi quero perguntar algumas coisas antes de ajustar o código: 1 - Você consulta nossa conversa por inteiro para melhorar nosso robo como por exemplo ver os ultimos scripts que deram certo; 2 - O agente faz sozinho uma interação com o eproc sozinho com base num comando, como por exemplo essa que estamos fazendo, procurar um processo, uma apelação com as palavras chaves, sem precisar de script? 3 = Se o agente faz sozinho, como ele sabe onde "apertar" o botão certo na tela para por exemplo clicar nos localizadores, ele consulta os scripts da pagina e se ele consulta, ele não pode registrar isso para aprimorar nosso script do robo? E, na programação python é possível separar os blocos de programação com numeração, por exemplo: 1 - bloco do script: 2 - outro bloco do script, assim acho que seria mais fácil para mim que não sou programador achar o bloco que precisa para substituir.

O ChatGPT disse:

3 - Na movimentação do processo na parte documento você vai encontrar assim: 1 - SENT1 - sentença do processo; 2 - Apeação APELAÇÃO 3 - CONTRAZAP1 - contrarrazões 4 - Parecer do Ministério Publico: PROMOÇÃO ##### Voee conhece esses "apelidos" que o sistema da?

4 - Na movimentação do processo na parte documento você vai encontrar assim: 1 - SENT1 - sentença do processo; 2 - Apeação APELAÇÃO 3 - CONTRAZAP1 - contrarrazões 4 - Parecer do Ministério Publico: PROMOÇÃO ##### Voee conhece esses "apelidos" que o eproc da para as peças processuais? (não programe ainda)

Sim, esses nomes “apelidos” são exatamente as chaves que o Eproc usa para identificar cada tipo de documento no fluxo de eventos. Por exemplo:

- **SENT1** costuma indicar a sentença;
- **APELAÇÃO1** (ou “APE1”) é o evento da apelação;
- **CONTRAZAP1** corresponde às contrarrazões;
- **PROMOÇÃO1** é o parecer do Ministério Público.

No script, podemos usar essa nomenclatura para localizar com precisão a peça processual que você escolher. Se houver variações ou outros códigos que você encontre no seu dia a dia (por exemplo, “SENT2” ou “APELAÇÃO2”), avise para que possamos incluir todos eles.

5 - Você sabe que o agravo de instrumento e a apelação são recursos diferentes, por exemplo no agravo você não vai achar esses "apelidos" e sim somente Incluir INIC1? Também será que poderíamos separar na hora do menu apelação de agravo de instrumento?

Entendi. Sim, os dois recursos têm tratamento diferente dentro do Eproc:

- Para **apelações**, os documentos vêm marcados com códigos como **SENT1**, **APELAÇÃO1**, **CONTRAZAP1** e **PROMOÇÃO1**. Esses “apelidos” identificam a sentença, a própria apelação, as contrarrazões e o parecer do Ministério Público, respectivamente.

- No caso do **agravo de instrumento**, o sistema geralmente só disponibiliza o documento **INIC1** (petição inicial do agravo) e não usa aqueles outros códigos.

Dá para refletir essa diferença na própria interface: o robô pode abrir um menu onde você escolhe “Apelação” ou “Agravo de Instrumento”. Se você escolher apelação, ele procura e baixa as peças com os códigos SENT1/APE1/CONTRAZAP1/PROMOÇÃO1 (aplicando ou não o filtro de palavras-chave, como discutimos). Se você escolher agravo, ele passa a buscar apenas o **INIC1**.

7 - Também poderia ter uma opção extrair a peça a escolha do usuário sem procurar palavras chaves? (não comece a programar)

8 - Sim, isso é perfeitamente viável. A mesma janela de configuração pode trazer uma opção (como uma caixa de seleção ou botão) para indicar que você quer baixar a peça escolhida **sem aplicar o filtro de palavras-chave**. Se essa opção estiver marcada, o script vai simplesmente abrir o documento correspondente (sentença, apelação, contrarrazões ou parecer do Ministério Público) e salvá-lo, sem analisar o texto.

Em resumo, a interface poderia ter:

- Um campo para selecionar o tipo de peça processual (sentença, apelação, etc.).
- Um campo para definir palavras-chave (com conectores, se você quiser filtrar).

- Uma opção adicional como “baixar sem filtrar por palavras-chave”; se marcada, o campo de palavras-chave é ignorado.

8 - Ainda não comece a programar. Seria possível nessa mesma interface ele perguntar se voce quer a analise da sentença, apelação, contrarrazões, ou parecer do ministério publico. Assim poderia trocar a peça processual de analise ou copia quando o usuário quisesse sem precisar passar pela mudança do código. E possível?

É perfeitamente possível. Em vez de deixar a escolha da peça processual “engessada” no código, a interface pode ter mais um campo (um conjunto de opções ou um menu suspenso) onde você seleciona o tipo de documento a ser analisado ou copiado — por exemplo: “Sentença”, “Apelação”, “Contrarrazões” ou “Parecer do Ministério Público”.

O robô usaria esse valor para direcionar sua busca dentro do processo: ele identificaria o evento correspondente (“SENTENÇA”, “APELAÇÃO”, “CONTRARRAZÕES”, “PARECER”), extrairia ou analisaria o PDF e aplicaria as palavras-chave que você tiver definido. Assim, você consegue mudar de peça processual a qualquer momento sem editar o código.

Quando estiver pronto, podemos alinhar os detalhes (se você quer múltipla seleção ou escolha única, quais são os nomes exatos desses eventos no Eproc, etc.) e incluir essa funcionalidade na mesma janela de configuração.

10 - Outra coisa, ao rodar o robo ele não pode ter uma interface grafica (simples) para perguntar qual localizador quer entrar (claro dentro os

meus localizadores) e ao mesmo tempo pedir as palavras chaves ? (não comece a programar ainda).

Sem problemas — dá para fazer isso, sim. Em vez de editar o código todas as vezes, podemos criar uma janela simples que aparece ao iniciar o robô, na qual você escolha:

- Qual localizador de “Meus localizadores” quer usar (por exemplo, “0.06 Urgentes Duda”). Isso pode ser digitado ou exibido como uma lista para você selecionar.
- Quais palavras-chave ele deve procurar (incluindo conectores como **e/ou** e comodins com *****, como você mencionou). Você poderia digitar algo como **(casa ou gato) e piz***.

A interface pode ser feita em Tkinter ou outro módulo nativo do Python, sem necessidade de nada complexo: dois campos de texto ou um campo de seleção e um campo de palavras-chave. O script então usa esses valores para navegar até o localizador certo e, ao abrir cada apelação ou agravo, só salva o PDF se o texto atender aos critérios informados.

Antes de começarmos a programar, vale alinhar alguns pontos:

- Você prefere selecionar o localizador de uma lista (o próprio script poderia ler os nomes visíveis em “Meus localizadores”) ou digitar o texto do localizador?
- Como você gostaria de digitar as palavras-chave? Por exemplo, usar parênteses, “e/ou” e asterisco como no exemplo **(casa ou gato) e**

piz*?

Confirmando esses detalhes, posso preparar a interface de forma a atender exatamente ao que você precisa.

11 - Parabéns funciona. Não programe ainda, seria possível que o robo (script) pudesse achar o agravo ou a apelação e procurar dentro do texto algumas palavras e somente se achar essas palavras ele faria a copia?

Para fazer isso é possível, sim. A lógica seria: para cada apelação ou agravo, o robô abriria o documento em PDF, extrairia o texto e checaria se contém determinadas palavras-chave antes de salvar o arquivo em disco.

Para implementar isso, vou precisar saber quais palavras você quer que ele procure (por exemplo, nomes de partes, termos jurídicos específicos, etc.). Você poderia me dizer a lista de palavras que o script deve detectar no texto?

12 - E na procura poderia ter conectores como "e" "ou" ou mesmo palavra* que nesse caso com asterisco procuraria todas as palavras que tivessem o início palavra - palavra, palavra. Procurar por exemplo: (casa ou gato) e piz*. É possível?

É possível, sim, implementar essa lógica mais refinada. O que o script precisaria fazer é o seguinte:

1. Depois de localizar a apelação ou o agravo, baixar o PDF **apenas para análise temporária** (ou, se preferir, abrir o documento no próprio

navegador).

2. Extrair o texto desse PDF usando uma biblioteca de leitura de PDF, como **PyPDF2** ou **pdfplumber**. Essas bibliotecas permitem transformar o conteúdo do arquivo em uma string de texto em Python.
3. Avaliar o texto extraído com base nos conectores que você mencionou:
 - **e/AND**: todas as condições precisam ser verdadeiras.
 - **ou/OR**: qualquer uma das condições pode ser verdadeira.
 - **Asterisco (*)**: usar uma correspondência por prefixo (“piz*” encontra “pizza”, “pizzaria”, “pizaras”...).
4. Somente se o texto cumprir as regras (por exemplo, contém “casa” **ou** “gato” **e** alguma palavra começada por “piz”), o script salva o PDF no diretório final; caso contrário, descarta o arquivo.

Alguns pontos importantes:

- Para extrair texto de PDFs, você precisará de uma biblioteca extra (como **PyPDF2**), que não vem instalada por padrão no ambiente remoto. Isso deve ser instalado no seu computador onde o script será executado.
- É necessário definir exatamente quais palavras-chave e operadores você quer usar; no seu exemplo, seria algo como:
 - **Conjunto OR**: ["casa", "gato"]

- **Conjunto AND com wildcard:** `[("piz", True)]` onde `True` indica uso de prefixo (asterisco).

Se você me disser quais palavras e operadores quer utilizar, posso escrever uma versão ajustada do script incluindo essa filtragem (e comentar onde instalar a biblioteca de PDF).

Outros:

`C:\Users\emanuelamaral\prints\ nao_encontrei_localizador.png.`

`cd "C:\Users\emanuelamaral\Documents\Emma\Python"`

`python teste09.py`

`python Buscador04.py > saida.txt 2>&1`

