



Jadiel Eudes Mendonça Barbosa

Raspagem de Dados Jurídicos Utilizando Scrapy

Recife

2021

Jadiel Eudes Mendonça Barbosa

Raspagem de Dados Jurídicos Utilizando Scrapy

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Silvana Bocanegra

Recife

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

B238r

Barbosa, Jadiel Eudes Mendonça

Raspagem de dados jurídicos utilizando scrapy / Jadiel Eudes Mendonça Barbosa. - 2021.
16 f. : il.

Orientadora: Silvana Bocanegra.
Inclui referências.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em
Sistemas da Informação, Recife, 2021.

1. Raspagem de dados. 2. Dados jurídicos. 3. Rastreador web. 4. Scrapy. 5. Spiders. I. Bocanegra, Silvana, orient. II.
Título

CDD 004

Jadiel Eudes Mendonça Barbosa

Raspagem de Dados Jurídicos Utilizando Scrapy

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 20 de Dezembro de 2021.

BANCA EXAMINADORA

Victor Wanderley

Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Cleviton Monteiro

Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Rodrigo Soares

Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Raspagem de Dados Jurídicos Utilizando Scrapy

Jadiel Eudes Mendonça Barbosa ¹

¹Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil

jadiel132@hotmail.com

Resumo. *A raspagem de dados é uma técnica computacional na qual através de um programa é realizada extração de dados que estão escondidos em páginas da web. Dessa forma, este trabalho acadêmico tem como objetivo utilizar técnicas de raspagem de dados para extrair dados de processos jurídicos dos sites dos tribunais com o intuito de auxiliar empresas contratantes a tomarem decisões estratégicas junto a seus departamentos jurídicos.*

Palavras-chave: *Raspagem de Dados, Dados Jurídicos, Rastreador Web, Scrapy, Spiders*

Abstract. *web scraping is a computational technique that uses a program to extract data that are hidden in web pages. In this way, this academic work aims to use how web scraping techniques to extract data from legal processes from the websites of the courts in order to help contracting companies to take strategic decisions with their legal departments.*

Keywords: *Web Scraping, Legal Data, Web Crawler, Scrapy, Spiders*

1. Introdução

O processo de raspagem de dados que será descrito neste trabalho acadêmico, tem como foco a área de gestão jurídica, para isso, são desenvolvidos robôs que fazem varreduras diárias nos sites dos tribunais para capturar ou atualizar dados cadastrados anteriormente.

A Intelivix é uma empresa B2B (business-to-business), ou seja, é uma empresa que presta serviço para outras empresas. Seus clientes são grandes empresas que estão espalhadas por todo o território nacional e possuem grandes volumes de processos sendo tramitados na justiça. Os processos após serem extraídos do sites dos tribunais pelos robôs são tratados e disponibilizados em painéis para os clientes, ao qual poderão tomar decisões estratégicas ao cruzar e relacionar informações destes processos jurídicos.

1.1. Problema

Muitas empresas mesmo possuindo escritórios de advocacia, enfrentam problemas para tomar ciência dos processos em que são réu, podendo ser pela dificuldade de acesso aos sites dos tribunais, endereço incorreto para receber notificação da justiça, ou alguma limitação que os advogados possam ter para encontrar tais processos. Tais dificuldades acarretam em um maior problema, que é o tempo de descoberta. Quanto mais cedo um processo for encontrado maior será o tempo que a empresa terá para tomar uma decisão e montar uma defesa, podendo evitar a tramitação destes processos em instâncias superiores e reduzir seus custos advocatícios.

1.2. Objetivos

Desta forma, o objetivo é monitorar e atualizar as spiders existentes, assim como desenvolver novas spiders quando necessário para que possam englobar todos os sites dos tribunais e assim extrair dados de processos da forma mais rápida possível. A spider ou robô da web, é uma classe do framework scrapy (seção 3.2) que descreve todo o fluxo da raspagem dos dados (web scraping). Desta maneira, os dados que são extraídos poderão passar por transformações e serão organizados para que possa auxiliar o setor jurídico do cliente na tomada de decisão.

Meu objetivo é trabalhar junto com a equipe utilizando-se de várias ferramentas para monitorar os logs de execução das spiders. Com isso é possível identificar as linhas de código em que ocorreram erros para então corrigi-los. Além disso, contribuir na atualização das spiders quando houver mudanças no código fonte nos sites dos tribunais, novas regras de negócio ou novos requisitos. Por fim, desenvolver novas spiders para capturar dados de novos sistemas.

2. Referencial teórico

De início é necessário entendermos a diferença entre dados e informação para compreender o que se busca com as técnicas de extração de dados. Dito isto, o que entendemos por dados é que são elementos brutos, que estão espalhados pela internet e que por si só não apresentam nenhuma compreensão ou significado. Desta forma, dados não possuem valor para auxiliar uma tomada de decisão. No entanto, informação é um conjunto de dados que foram transformados, ordenados e organizados para gerar algum significado e compreensão, no qual nos dar embasamento para tomar uma decisão a respeito de um determinado assunto.

Com isso, podemos dizer que a internet é um aglomerado de dados com diferentes tipos de formatos e fontes, podendo ser encontrados em ambientes estruturados ou não estruturados, o que pode dificultar na coletas desses dados. Copiar e colar é uma forma de coletar dados da internet, porém é metódico e repetitivo, que exige uma grande demanda de tempo, o que torna o trabalho cansativo e inviável.

As páginas web geralmente são encontradas no formato *HTML*. *HTML* ou Linguagem de Marcação de HiperTexto, não é uma linguagem de programação, é apenas o bloco de construção mais básico da web, que possibilita a organização e formatação de textos e documentos. Hipertexto é um texto que tem referências em links que conectam páginas da internet que podem estar dentro do mesmo site ou em vários sites e que podem ser acessados imediatamente. O *HTML* possui seus elementos de "marcação" de texto que estão estruturados em tags com sintaxe de abertura e fechamento como `<tag>` e `</tag>` [Mozilla 2021a].

Essas páginas web são formadas por uma série de documentos hipermídia e são acessados através de URLs (Uniform Resource Locator, localizador de recurso uniforme). Esses documentos de hipermídia precisam trafegar pela internet entre cliente e servidor, a transferência desses arquivos se dar por meio de um protocolo chamado HTTP ((Hypertext Transfer Protocol). O cliente faz requisições a uma página web que contém documentos como, textos, imagens e vídeos e essa página web fica hospedada em um servidor. A comunicação HTTP é bidirecional, permitindo dois recursos para a diminuição do tráfego de rede e o aumento de desempenho. Com a medida que a internet foi evoluindo e o

tráfego de dados sensíveis ficou mais constante, houve a necessidade de proteger esses dados, então foi criado o HTTPS. O HTTPS ((Hypertext Transfer Protocol, Secure) é o HTTP usando o SSL (Secure Socket Layer, camada de soquete seguro) ou o TSL((Transport Layer Security, segurança da camada de transporte) que é uma camada de segurança no qual os dados são criptografados antes de trafegarem na internet [Torres 2021].

A prática de extrair dados da internet, ou raspagem de dados como também é chamada, já foi conhecida como screen scraping, data mining, web harvesting, entre outras variações. Entretanto atualmente, num consenso geral o termo web scraping é o mais utilizado [Mitchell 2019].

Web scraping é a técnica de extrair dados de forma automatizada de um ou mais sites, estruturando-os e salvando-os em arquivos, como planilhas ou banco de dados, no qual facilitará uma possível análise ou visualização desses dados. Essas extrações são feitas por meio de expressões regulares com a ajuda de seletores CSS ou XPath.

Web crawler (rastreadores) é um programa que vai descobrir URLs, o crawler vai acessar uma ou mais URLs, descobre e extrai hiperlinks associados de forma recursiva. Na extração de dados da web, há normalmente uma combinação de web crawler com web scraping, no qual o crawler irá encontrar as URLs e baixar seus conteúdos para que então possa ser realizada a raspagem dos dados [Najork 2009].

Então de forma geral, como está ilustrado na figura 1, as URLs são acessadas e as páginas são baixadas, logo após é realizado a raspagem dos dados, no qual os dados são tratados e persistidos em alguma base de dados.

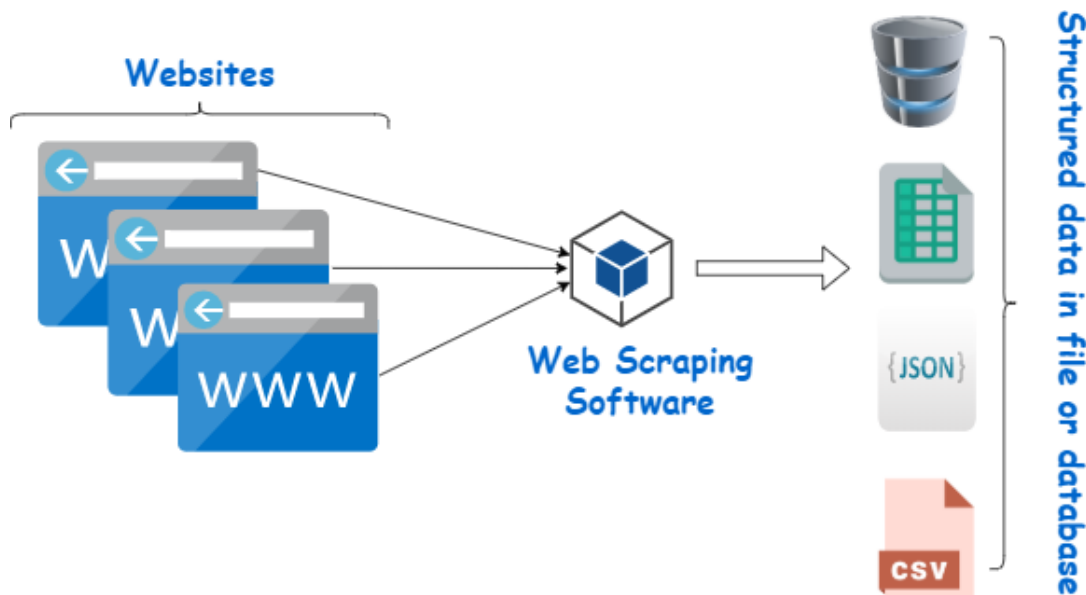


Figura 1. Representação geral de web scraping.

<https://www.webharvy.com/articles/what-is-web-scraping.html>

3. Ferramentas utilizadas

Este tópico irá retratar sobre as ferramentas que são utilizadas na empresa, deste a descoberta de um processo, passando por sua extração até o armazenamento.

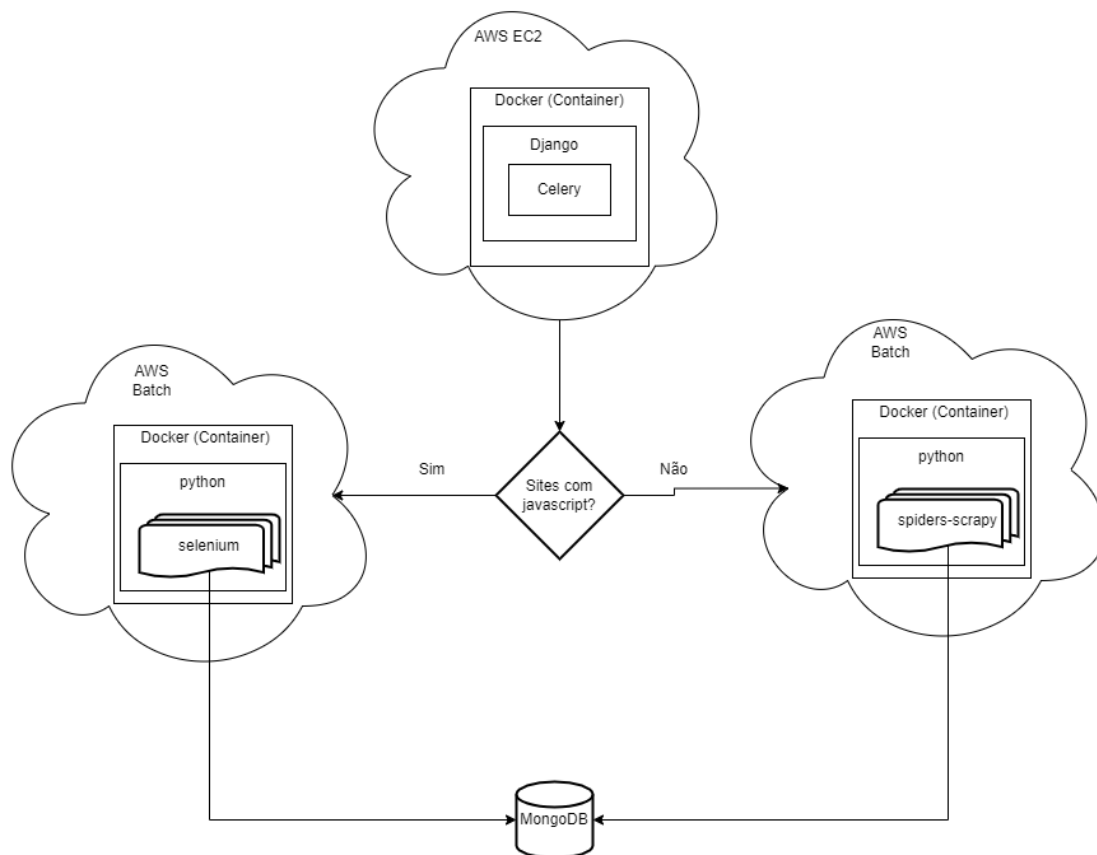


Figura 2. Visão geral da arquitetura do projeto.

O projeto django roda dentro de um container docker no Amazon EC2 e com o framework celery gerencia a fila de spiders que serão executadas. As spiders scrapy assim como os raspadores em selenium rodam dentro de uma container docker no Amazon Batch. Quando um site possui javascript, é rodado o projeto selenium, quando não, a raspagem é feita pelas spiders scrapy (Figura 2).

3.1. Python

Python é uma linguagem de alto nível de tipagem dinâmica e forte, com uma abordagem simples. É uma linguagem interpretada, que oferece recursos de orientação a objeto, multiparadigma, além de recursos poderosos em sua biblioteca padrão, tornando-se ideal para o desenvolvimento de scripts e aplicativos multiplataformas [Python 2021].

3.2. Scrapy

Scrapy é um framework poderoso de código aberto desenvolvido em python. Esse framework integra muito bem web scraping e web crawler para extração de dados das páginas web, podendo ser utilizado também para extrair dados em APIs [Scrapy 2021]. Abaixo está listado o que é necessário para montar um raspador scrapy.

- **Spiders** - São classes no qual é definido e estruturado o comportamento de rastreamento e análise de páginas web de um ou vários sites.

- **Selectors** - São mecanismos para extrair dados de dentro do HTML, selecionando um nó (tag HTML) específico. Os seletores são os XPath e as expressões CSS.
- **Items** - São objetos python para os quais são retornados os dados extraídos pelas spider. O retorno é na estrutura chave-valor, como um dicionário.
- **Item Loaders** - São carregadores de itens, ou seja, ele pega o dado que foi raspado e atribui ao item.
- **Item Pipeline** - É uma classe que irá executar uma ação em um item já raspado. Ações comuns a serem executadas pelo pipeline são as limpezas e validações sobre os dados extraídos e seu armazenamento em um banco de dados.

3.3. Selenium

Selenium é um framework de código aberto e multiplataforma. Foi desenvolvido com o objetivo de testar aplicações web de forma automatizada, simulando o usuário utilizando um navegador.

3.4. Docker

Docker é uma plataforma open source desenvolvida em Go pela google. Facilita o desenvolvimento ágil simplificando a metodologia de DevOps, tendo como enviar, testar e implementar código rapidamente [Tecnologia 2021].

O docker é baseado em containers, na criação, execução e publicação(deploy) desses containers. Um container é a forma como uma aplicação e suas dependências serão empacotadas.

Os containers compartilham o kernel, bibliotecas e arquivos binários do SO(sistema operacional) ao qual estão hospedados. Esses arquivos que são compartilhados servem somente como leitura, o que torna a virtualização do sistema mais rápido.

3.5. Django

Django é um framework web full stack desenvolvido em python e de código aberto. Este framework tem como objetivo tornar o desenvolvimento das aplicações web mais rápido e tem uma arquitetura baseada no modelo **MVT** ou **Model-View-Template** [Roveda 2021].

- **Model** - é a camada responsável pela comunicação com o banco de dados.
- **View** - é a camada responsável por enviar uma resposta para uma requisição recebida.
- **Template** é a camada na qual o usuário final irá visualizar os dados da aplicação.

3.6. MongoDB

O mongodb é um banco de dados de código aberto e multiplataforma classificado como NoSQL. O mongo é orientado a documentos, ou seja, os dados são guardados em documentos semelhantes ao JSON, numa estrutura de chave-valor, diferentemente dos bancos de modelos relacionais, em que os dados são registrados em linhas e colunas. Além disso, o mongodb tem como objetivo oferecer escalabilidade, desempenho e alta disponibilidade [Guedes 2021].

3.7. AWS

A AWS - Amazon Web Services é um conjunto de produtos e serviços de computação em nuvem. A AWS já soma mais de 200 desses serviços, como AWS Batch, Amazon EC2, Amazon S3 e AWS Lambda, que são os serviços utilizados neste trabalho [AWS 2021].

O AWS Batch gerencia dinamicamente tarefas de computação enviadas em lotes, alocando recursos necessários como memória e cpu de acordo com o tamanho dos lotes.

O Amazon Elastic Compute Cloud ou Amazon EC2 é um serviço que disponibiliza capacidade computacional, no qual o cliente tem poder de escolha de processador, rede, memória, armazenamento, entre outros.

O Amazon Simple Storage Service ou Amazon S3 é um serviço de armazenamento de objetos, no qual é oferecido escalabilidade, disponibilidade, segurança e performance.

O Amazon Lambda é um serviço de computação orientada a eventos que pode ser acionada por mais de 200 serviços que são ofertados pela Amazon.

O Amazon Simple Notification Service ou Amazon SNS é um serviço de entrega de mensagens de editores para assinantes. A comunicação entre editores e assinantes é feita de forma assíncrona, no qual a mensagem é enviada para um tópico, que é um canal de comunicação e um ponto de acesso lógico.

3.8. GitHub

GitHub é um serviço de computação em nuvem para armazenamento de códigos. O git é uma ferramenta para gerenciar o versionamento destes códigos. O Github utiliza o git para manter os códigos atualizados.

O Github actions é uma ferramenta de automatização de fluxos de trabalhos, que possui os componentes CI/CD ou continuous integration e continuous deployment [GitHub 2021].

- **Continuous Integration** - Cria, testa e integra algum fluxo de trabalho de forma automatizada em um repositório compartilhado.
- **Continuous Deployment** - Realiza alterações em códigos prontos para o ambiente de produção ou que estão diretamente no cliente.

3.9. Celery

Celery é uma fila de tarefas assíncronas que tem foco no processamento em tempo real, porém também possui agendamento de tarefas. As filas de tarefas são mecanismos para distribuir os trabalhos em threads ou máquinas. Essas filas de tarefas são monitoradas constantemente para adicionar novos trabalhos para serem executados conforme a necessidade [Solem 2021].

4. Abordagem proposta

A raspagem de dados dos sites dos tribunais brasileiros desde trabalho, é realizado por meio das spiders do framework scrapy. A spider necessita de uma URL inicial ou uma

lista de URLs iniciais para iniciar o rastreamento dessas URLs, no qual, deve haver uma função de retorno de chamada que irá obter as respostas baixadas dessas solicitações.

Nas funções de retorno que serão feitas as análises da página que foi baixada, essas análises geralmente são feitas utilizando seletores. Os dados analisados deverão ser carregados nos itens com a ajuda de carregadores de itens(item loaders), no qual, o item será retornado para o pipeline para ser persistido em um banco de dados, que neste trabalho é utilizado o MongoDB.

A empresa no qual este trabalho acadêmico é baseado, possui mais de 200 spiders que fazem raspagem de dados nos sites dos tribunais diariamente, em busca de novos processos, atualizar processos antigos e certidões eletrônicas. As spiders rodam em um container docker utilizando o AWS Batch, serviço da Amazon Web Services.

Alguns sites de tribunais adicionam algumas dificuldades para evitar que robôs façam varreduras e operações automatizadas. Como estes acessos aos sites dos tribunais são para obter informações públicas e os robôs não estão violando nenhuma lei ou mecanismo de segurança, a raspagem de dados está de acordo com o que é permitido pela lei de nº 12.537/11 (Lei de Acesso a Informação) e pela lei de nº 12.965/14 (Lei do Marco Civil da Internet) [Reinis 2021]. Uma dessas dificuldades é o bloqueio de IP de servidor, como as spiders rodam na AWS, o IP que chega ao site dos tribunais é o de servidor. Uma solução para este problema é utilizar serviço de proxy. O proxy funciona como um intermediário entre o usuário e os serviços de internet por ele acessado. Desta forma, todo o tráfego que antes pertencia ao servidor é direcionado para IPs residenciais, conseguindo assim, acesso ao site desejado. Outro problema encontrado é que alguns sites possuem captcha e recaptcha.

O Captcha (figura 3) (Completely Automated Public Turing test to tell Computers and Humans Apart, Teste de Turing público completamente automatizado para distinguir entre computadores e pessoas) é uma medida de segurança para tentar evitar o acesso de robôs, no qual, são geradas imagens aleatórias de letras e/ou números e a pessoa tem que digitar o que está vendo na imagem [Google 2021]. O reCaptcha (figura 4) foi baseado no captcha e também é uma medida de segurança que visa bloquear spam e dificultar acesso de robôs, porém, ao invés de digitar o que se vê na imagem, é mostrado várias imagens de um determinado objeto, no qual, a pessoa tem que selecionar as imagens referente ao objeto que foi dito.

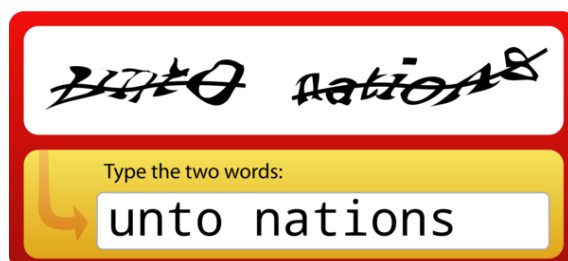


Figura 3. Resolução de captcha.

<https://rockcontent.com/br/blog/captcha/>



Figura 4. Resolução de recaptcha.
<https://www.google.com/recaptcha/about/>

Para a resolução de captcha e recaptcha, foi adquirido um serviço da API 2captcha. 2captcha é um serviço de reconhecimento de imagem e captcha movido por humanos, no qual tem como objetivo a resolução mais rápida e precisa possível de captchas [2captcha 2021].

Também há páginas web escritas em javascript, que é um problema para o framework scrapy, pois o scrapy não executa scripts que tiverem presentes no HTML, apenas faz o download da página como é enviada do servidor. Javascript é uma linguagem de programação interpretada, multi-paradigma e dinâmica, conhecida como linguagem de scripting para páginas Web. Com o javascript, pode-se programar o comportamento de uma página Web a partir da ocorrência de um evento [Mozilla 2021b].

Uma alternativa para renderizar páginas escritas em javascript é utilizando scrapy-splash. Splash é um serviço de renderização de javascript escrito em lua. No entanto, a equipe encontrou algumas dificuldades no projeto, a primeira é que o splash é desenvolvido em lua, o segundo problema é que o splash tem a sua api rodando em um container docker, isso acaba sendo um problema pois as spiders já rodam dentro de um container docker no AWS Batch e a equipe não encontrou uma solução adequada para possuir dois containers docker se comunicando dentro do AWS Batch. Por conta dessas dificuldades com o splash, a solução adotada pela equipe foi criar spiders em selenium. Como o selenium utiliza um navegador para simular um usuário acessando uma página, não tem problema para renderizar páginas escritas em javascript. Contudo, o selenium também traz alguns problemas para a equipe, um dos problemas é modelar o projeto em selenium para ficar similar as spiders feitas em scrapy, outro problema é a dificuldade de adicionar

serviço de proxy com o selenium, pois diferente do scrapy, não há uma integração viável e simples.

Para manter um serviço de qualidade com capturas de processos em tempo hábil, é necessário um constante monitoramento das spiders, pois os sites dos tribunais estão sempre com mudanças que geralmente alteram o fluxo de extração da spider. Além das ferramentas já citadas anteriormente, algumas outras auxiliam na manutenibilidade do projeto, é o caso do Sentry, Mitmproxy e o Power BI.

O Sentry é uma API que gera relatórios de erros e exceções, bem como problemas de desempenho no projeto. O Power BI é uma coleção de serviços que tem como objetivo transformar os dados em informações coerentes com uma visualização que envolva o usuário. O Mitmproxy é um conjunto de ferramentas que fornece um proxy de interceptação que pode ser usado para interceptar, inspecionar modificar e reproduzir tráfegos da internet de qualquer protocolo que seja protegido por SSL/TSL [Aldo Cortesi 2021].

Então de forma geral o fluxo (figura 5) acontece da seguinte forma: As spiders com seus termos ou números de processos definidos são selecionadas para o processo de raspagem de dados através do projeto django, que envia lotes de jobs para rodar nos servidores da AWS. Quando os jobs são finalizados e não produzem erros, as spiders que são de certidão eletrônica, fazem downloads dos PDFs e os enviam para o Amazon S3, quando é uma spider de processo, os processos que são extraídos são salvos no MongoDB. Porém quando é retornado erro, é enviado a lista de spiders com erro para o sentry, no qual mostrará mais detalhado em qual local aconteceu o erro. Além disso, há um relatório mais geral de erros, utilizando o Power BI, que traz informações como quantidade de dias que uma spider passou quebrada, quantidade de erros mais frequentes, sistemas mais problemáticos, entre outros, no qual a equipe consegue ter um controle maior e uma melhor tomada de decisão para corrigir um problema. Vale ressaltar a importância do Github Actions também, pois com constantes alterações e correções nas spiders, o trabalho repetitivo e cansativo de atualizar a imagem do projeto diariamente para que sempre esteja a versão mais atual disponível acaba sendo facilitado.

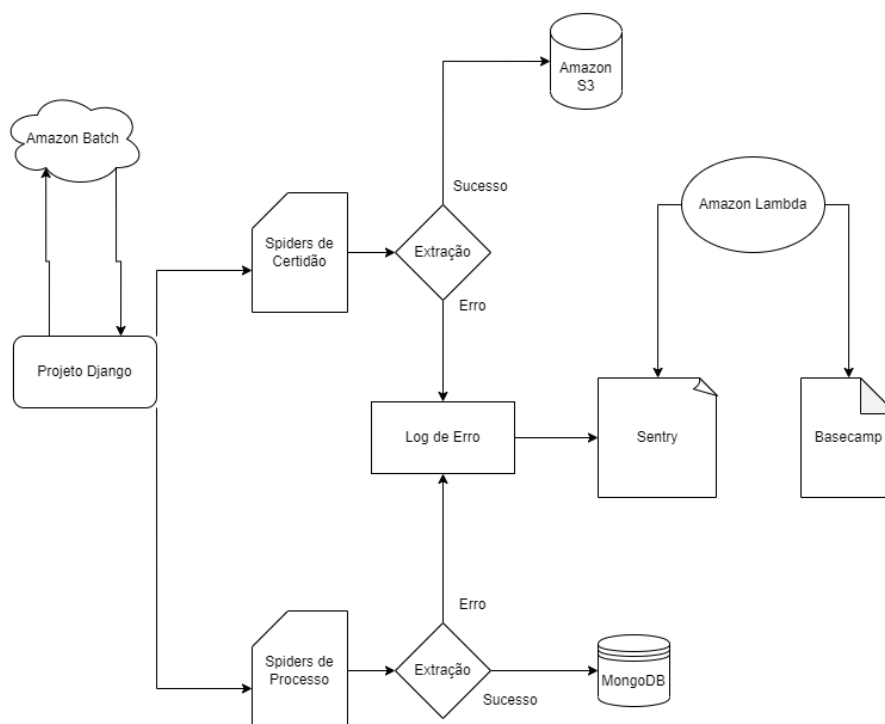


Figura 5. Fluxo de extração.

A extração do processo em si ocorre da seguinte forma: É realizada uma requisição para a URL inicial (figura 6). Depois de preencher os dados necessários para realizar a consulta, é retornado um processo ou uma lista de processos. O processo é dividido em 4 partes, cada um com sua extração, com seus itens e carregadores de itens. Temos então os dados iniciais do processo como pode ser visto na figura 7, as partes envolvidas no processo e seus respectivos polos, no qual, nome_da_parte é referente a pessoa ou empresa envolvida no processo, documento é o CPF ou CNPJ da parte, papel_da_parte refere-se ao papel da pessoa ou empresa envolvida no processo, por fim, advogado_da_parte é o nome do advogado que defende a parte envolvida no processo e o documento do advogado pode ser o CPF ou OAB (figura 8), as movimentações ou andamento do processo, que é o que aconteceu no processo (figura 9) e por fim, os documentos que estão anexados ao processo (figura 10). Ao final do processo de extração, como antes mencionado, os dados são organizados e salvos no MongoDB como pode ser visto na figura 11.

PJe Consulta pública


Processo	Processo	Última movimentação
<input type="text" value="_____.____.8.17.____"/> Processo referência Numeração única <input checked="" type="radio"/> Livre <input type="radio"/> <input type="text"/> Nome da Parte <input type="text"/> Nome do advogado <input type="text"/> Classe Judicial <input type="text"/> CPF <input checked="" type="radio"/> CNPJ <input type="radio"/> <input type="text"/> OAB (000000 A UF) <input type="text"/> - UF <input type="text"/>	resultados encontrados A presente consulta não retornará qualquer resultado em caso de informações prestadas incorretamente ou de processos sob sigilo de justiça, conforme art. 1º, parágrafo único, da Resolução nº 121 do Conselho Nacional de Justiça.	
<input type="button" value="PESQUISAR"/>		

Figura 6. URL inicial de consulta de processo do sistema PJE.

Dados do Processo			
Número Processo	Data da Distribuição	Classe Judicial	Assunto
	21/03/	PROCEDIMENTO DO JUIZADO ESPECIAL CÍVEL (436)	DIREITO CIVIL (899) - Obrigações (7681) - Espécies de Contratos (9580) -
Jurisdição		Órgão Julgador	
Olinda - Juizados		1º Juizado Especial Cível e das Relações de Consumo de Olinda	

Figura 7. Dados iniciais de um processo.

Polo ativo

Participante	Situação
Nome_da_parte - documento (Papel_da_parte)	Ativo
Advogado_da_parte - documento (Advogado)	Ativo

2 resultados encontrados

Polo Passivo

Participante	Situação
Nome da parte - documento (Papel da parte)	Ativo
Nome_da_parte - documento (Papel_da_parte)	Ativo
Advogado_da_parte - documento (Advogado)	Ativo
Advogado_da_parte - documento (Advogado)	Ativo

Figura 8. Partes envolvidas em um processo.

Movimentações do Processo

Movimento	Documento
05/12/2019 16:53:38 - Arquivado Definitivamente	
05/12/2019 16:53:21 - Expedição de Certidão.	
05/12/2019 16:52:36 - Expedição de Outros documentos.	
05/12/2019 12:35:05 - Ato ordinatório praticado	

Figura 9. Movimentações de um processo.

Documentos juntados ao processo

Documento	Certidão
22/11/2019 08:59:48 - DESPACHO (DESPACHO)	
10/10/2018 09:08:48 - DECISÃO (DECISÃO)	
10/09/2018 09:49:44 - DESPACHO (DESPACHO)	
22/08/2018 22:58:20 - SENTENÇA (SENTENÇA)	

4 resultados encontrados

Figura 10. Documentos de um processo.

Key	Value	Type
(1) ObjectId("61a4cf443d5641ee...")	{ 24 fields }	Object
_id		ObjectId
diferenciador		String
fonte	pje_v2	String
numero		String
assuntos	[4 elements]	Array
classe_cnj	EMBARGOS DE DECLARAÇÃO CÍVEL (16...	String
comarca	Recife - Turma Recursal - Cível	String
data_distribuicao		String
desembargadores	[1 element]	Array
eletronico	true	Boolean
estado	PE	String
instancia	2 Grau	String
inteligivix_id	[1 element]	Array
juizo	Terceira Turma Recursal	String
justica	estadual	String
raw	{ 4 fields }	Object
spider		String
spider_id		Int32
tasks	[1 element]	Array
termos	[1 element]	Array
termos_cliente	[1 element]	Array
partes	[4 elements]	Array
andamentos	[14 elements]	Array
documentos	[6 elements]	Array

Figura 11. Dados de um processo salvo no MongoDB.

5. Conclusões

O objetivo do projeto é encontrar e extrair dados de processos jurídicos da maneira mais rápida possível. Isso requer monitoramento constante das spiders e dos sites dos tribunais. A equipe possui várias ferramentas para controlar os erros que surgem, conseguindo ser mais eficiente na resolução do problema e mantendo o projeto sempre atualizado e disponível para realizar extrações. Desta forma, meu objetivo também foi alcançado, pois para que tais processos sejam extraídos, é necessário dar manutenção, atualizar e desenvolver novas spiders para abranger novos sistemas.

De qualquer maneira, há algumas coisas que podem ser melhoradas, porém demandam uma maior quantidade de tempo para a sua resolução e ficam como implementações futuras no projeto. Uma dessas melhorias é a utilização do framework splash para integrar com o scrapy e poder renderizar páginas que são implementadas em javascript, com isso não haveria necessidade de manter o projeto selenium, evitando perder tempo com algumas modelagens de projeto para que possa ter a mesma estrutura do scrapy, além de ter garantido a integração com um serviço de proxy nas spiders desenvolvidas em scrapy. Uma outra melhoria seria desenvolver o próprio resolvidor de captcha, no qual iria diminuir um pouco do custo de gastos fixos do projeto, já que o serviço de captcha é pago. No geral, o trabalho atende as necessidades dos clientes, que é obter o maior número de processos jurídicos no menor tempo possível.

Referências

- 2captcha (2021). Api 2captcha. <https://2captcha.com/2captcha-api>. Acessado: 26/11/2021.
- Aldo Cortesi, Maximilian Hils, T. K. (2021). mitmproxy - an interactive https proxy. <https://mitmproxy.org/#mitmweb>. Acessado: 28/11/2021.
- AWS (2021). O que é aws? como funciona amazon web services. <https://aws.amazon.com/pt/what-is-aws/>. Acessado: 18/11/2021.
- GitHub, I. (2021). Continuous integration and continuous delivery (ci/cd) fundamentals | github resources. [//resources.github.com/ci-cd/](https://resources.github.com/ci-cd/). Acessado: 13/11/2021.
- Google (2021). O que é o captcha? - ajuda do administrador do google workspace. <https://support.google.com/a/answer/1217728?hl=pt-br>. Acessado: 26/11/2021.
- Guedes, M. (2021). O que é mongodb? | blog treinaweb. <https://www.treinaweb.com.br/blog/o-que-e-mongodb>. Acessado: 12/11/2021.
- Mitchell, R. (2019). *Web Scraping com Python: Coletando dados na Web moderna*, pages 9, 125. Novatec Editora Ltda.
- Mozilla (2021a). Html: Linguagem de marcação de hipertexto | mdn. <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acessado: 15/11/2021.
- Mozilla (2021b). Sobre javascript - javascript | mdn. https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript. Acessado: 26/11/2021.
- Najork, M. (2009). Web crawler architecture.
- Python, S. F. (2021). The python tutorial — python 3.10.0 documentation. <https://docs.python.org/3/tutorial/index.html>. Acessado: 08/11/2021.
- Reinis, O. A. (2021). Big data e acesso à informação - a legalidade do uso de bots (robôs). <https://oareinis.jusbrasil.com.br/artigos/590149343/big-data-e-acesso-a-informacao-a-legalidade-do-uso-de-bots-robos>. Acessado: 20/12/2021.
- Roveda, U. (2021). O que é django, para que serve e como usar este framework. <https://kenzie.com.br/blog/django/>. Acessado: 11/11/2021.
- Scrapy (2021). Scrapy à primeira vista - documentação do scrapy 2.5.1. <https://docs.scrapy.org/en/latest/intro/overview.html>. Acessado: 09/11/2021.
- Solem, A. (2021). Celery - fila de tarefas distribuída - documentação do celery 5.2.1. <https://docs.celeryproject.org/en/stable/>. Acessado: 16/11/2021.
- Tecnologia, P. (2021). Container docker: o que é e quais são as vantagens de usar? - panorama positivo - tudo sobre tecnologia da informação. <https://www.meupositivo.com.br/panoramapositivo/container-docker/>. Acessado: 10/11/2021.
- Torres, G. (2021). *Rede de Computadores*, pages 342–344. SF Editorial.