Diário de bordo

1ª Semana

Nos primeiros dias entramos no site do Spotify e verificamos se poderÍamos pegar o HTML da página, só que em dica do nosso professor, Eduardo Heredia, procuramos no site do Spotify uma API que nos desse os dados que precisaríamos.
Começamos a ler e testar a API do Spotify, utilizando o próprio terminal, porém percebemos que para cada usuário ao pegar seus dados, precisaríamos de um token que só com a autorização do mesmo poderíamos pegar as informações, assim, seria impossível conseguir token/autorização de milhares de usuários. Portanto, voltamos a estaca zero, que seria pegar as informações do HTML do site play.spotify.com.
Começamos a testar pegar o HTML no terminal utilizando o comando wget, porém ocorre que ao colocar play.spotify.com no wget ele retornava que o browser estava errado e acabei sendo redirecionado para outra página.
De fato, ocorria que o site tinha um verificador de browsers, se não for nenhum dos que eles determinam compatíveis com o site, eles redirecionam falando que seu navegador é inválido. Assim é preciso quando começar a "Crawlear", é necessário enganar o HTML da página identificando que o seu crawler é um navegador conhecido e suportado pelo site para que consiga entrar na página inicial.
Outra coisa que descobrimos nessa semana é que será preciso fazer o crawler entrar com um login e um senha, para conseguirmos ver todas as informações ali contidas
Também forma começados os estudos com Node.JS mostrando o processo de instalação, como funciona e tratamento de múltiplos requests.

3ª Semana

Descobrimos como fazer um scrap de um pagina da web em node, seguimos um tutorial muito bom feito pela scotch.io. Ele ensinava como fazer um request através de uma URL, como buscar no HTML através do JQuery os dados que você quer, como transformar os arquivos em JSON e colocar em um arquivo separado.

Explicando biblioteca por biblioteca:
Express: é um framework com muitas utilidades para serviços web, neste caso utilizamos para definir o caminho para execução do crawler.
File System (fs): É um biblioteca nativa de NodeJS, que tem como principal objetivo ter acesso as principais funções do sistema, neste caso utilizamos para salvar os dados que pegamos em um arquivo .json.
Request: Serve para tornar simples as chamadas de requisições de dados. Utilizamos para acessar o html de uma página através da URL.
Cheerio: Serve para poder utilizar JQuery do lado do servidor. Utilizamos para buscar no html os dados.

Publicado no github como: aprendizados/nodejs_webscraping
Link do tutorial: (https://scotch.io/tutorials/scraping-the-web-with-node-js)

Também tentamos colocar o login, mas antes era necessário identificar o navegador como um compatível com o spotify e depois de algumas pesquisas foi visto que o certo era alterar o user-agent para um identificador de qualquer navegador válido. Neste caso, utilizamos o Google Chrome.

Agora é necessário logar. A parte complicada é que é utilizado um padrão OAuth, que utiliza da criptografia RSA-SHA1 e também é necessário usar a api do Spotify para gerar este código, para que possa logar e pegar o html.
https://github.com/request/request#oauth-signing
https://github.com/andreareginato/simple-oauth2
https://developer.spotify.com/web-api/authorization-guide/#authorization-code-flow

---

4ª Semana

Após orientação do professor orientador Eduardo Heredia, o grupo chegou a conclusão de que deveria-se mudar a plataforma pela qual estava sendo explorada. Devido as limitações da API do Spotify o grupo remodelou o projeto e o foco da pesquisa, alterando de ramo musical para vagas de empregos no Brasil.
Mediante as alterações o projeto acabou ficando com alguns atrasos e o cronograma teve de ser alterado para o novo modelo de projeto.
Após todas as alterações feitas, foi refeito o modelo conceitual, a partir deste o grupo fez o modelo lógico e desenvolveu o site do crawler.
Nesta semana também foi registrado o domínio que hospedará a visualização de dados, além disto, o grupo terminou o desenvolvimento do crawler que pega os dados necessários para a plataforma no site da infoJobs. Também foi dado inicio ao crawler para pegar os dados no site da Catho e do Manager.