**Introdução**

A escolha de um estabelecimento de ensino superior e o seu respetivo curso é uma tarefa que abrange todos os estudantes que ingressam num plano curricular. Atualmente, esta escolha é feita com base na informação proveniente do site da Direção Geral do Ensino Superior (DGES) que, por sua vez, apresenta a mesma de forma antiquada através de menus e tabelas.

O objetivo deste projeto é facilitar a escolha do estabelecimento e curso superior através de uma interface mais gráfica, com mais opções de pesquisa e com melhor apresentação de informação sobre os temas anteriormente referidos. Para isso, o projeto foi repartido em duas grandes partes:

- Um web scrawler contendo alguns scripts para popular as tabelas da base de dados

- Uma aplicação web para a consulta das tabelas anteriormente preenchidas pelo crawler.

Para o efeito, então, o workflow da operação começa no web crawler. Este irá percorrer páginas chave do site da DGES em busca de informação relativas ás instituições, cursos e a relação entre estes os dois – isto é, quais são os cursos que pertencem a uma determinada instituição. O algoritmo também irá percorrer o site <https://www.pordata.pt/> de modo a capturar informações sobre a qualidade de vida em cada cidade relevante e irá preencher as tabelas cidades e distrito com base no sítio web <http://pt.gpspostcode.com/codigo-postal/portugal/#111>

Depois da base de dados estar completamente preenchida, a nossa aplicação web irá, então, disponibilizar toda a informação com recurso ao API do Google Maps. Com o seu auxilio, pretendemos que o utilizador possa selecionar um conjunto das seguintes opções de pesquisa:

- Pesquisar por instituição/curso

- Pesquisar por nota mínima

- Pesquisar por distrito

- Pesquisar por cidade

- Pesquisar por qualidade de vida

- Pesquisar por poder de compra

- Pesquisar por ranking

Cada um dos resultados da pesquisa será indicado no mapa, bem como informações relevantes sobre a pesquisa. Ao clicar no resultado, o utilizador será levado para uma página onde poderá encontrar mais informações sobre a instituição/curso.

**Scrapper**

A framework de web crawling que escolhemos é a *scrapy* – uma framework para python. Utilizamos esta ferramenta para retirar informação de sítios web públicos e preencher as tabelas na base de dados relativas ás instituições com essa mesma informação. Isto é:

* Cidades
* Cursos
* Distritos
* Instituicoes
* Instituicoes\_has\_cursos
* Tipos\_curso
* Tipos\_ensino

**Scrapy – Arquitetura**

O Scrappy foi desenvolvido com a Twisted, uma framework de networking orientada por eventos. Deste modo, o Scrappy utiliza código assíncrono para suportar processamentos em paralelo, fazendo-o uma framework extremamente rápida e capaz

**Fluxo de dados**

**Diagram

Description automatically generated**

O fluxo de dados no Scrapy é controlado pelo motor de execução e funciona assim:

1. O motor recebe as solicitações iniciais para fazer *crawl* da *Spider* que, é o script que o utilizador da framework escreve para efetuar um scrape.
2. O motor agenda as solicitações no Agendador (Scheduler) e pede as próximas solicitações a serem rastreadas.
3. O Agendador retorna as próximas Solicitações ao Motor
4. O Motor envia as Solicitações para o Downloader, passando pelos Middlewares do Downloader
5. Assim que a página termina o download, o Downloader gera uma Resposta contendo o HTML e demais código proveniente da página em questão. Envia, depois, essa informação para o Motor passando pelos Middlewares do Downloader de novo
6. O Motor recebe a Resposta do Downloader e envia-a para a Spider, passando pelo middleware da Spider.
7. A Spider processa a Resposta e retorna itens retirados para o Motor, passando pelo Middleware da Spider
8. O Motor envia itens processados para as pipelines de item e, em seguida, envia solicitações processadas para o agendador e pede pelas próximas solicitações possíveis de rastreamento (novo url)
9. O processo repete-se da etapa 1 até que não haja mais solicitações do Agendador

**Componentes**

* Motor Scrapy: Responsável por controlar o fluxo de dados entre todos os componentes do sistema e desencadeia eventos quando certas ações ocorrem
* Agendador**:** Recebe solicitações do Motor e coloca-os numa fila para os processar quando o Motor o requisita.
* Downloader: Responsável por descarregar páginas web e encaminha-las para o Motor, que, por sua vez, as encaminha para a Spider
* Spider: Classes escritas por utilizadores da framework para analisar respostas e extrair dados
* Pipeline do item:Responsável pelo processamento dos itens depois de serem extraídos pela Spider. Tipicamente inclui tratamento da informação, validação e persistência.
* Middlewares do Downloader:Processam as solicitações entre do Motor para o Downloader e as respostas que passam do Downloader para o Motor
* Middlewares da Spider: Processam informação proveniente do Motor para a Spider ou vice-versa.