O Iridium é uma 'search engine' simples, construída em Python, que foi desenvolvida como trabalho prático final da disciplina de PDSII na UFMG. A proposta do trabalho em questão era de criar um sistema de consultas para armazenar dados e retornar as informações requisitadas via *queries*. O processo de recuperação de informação é o processo de conversão de dados através de uma *Máquina de Busca*, que recebe uma expressão de busca, a analisa, e retorna os dados que foram requisitados inicialmente. Tal busca deve ser realizada, por ordem da proposta, através de uma ordenação por *índice invertido*.

O esqueleto do programa consiste em duas classes principais – a primeira funciona como uma espécie de 'barreira' para a entrada de dados. Ela recebe o texto e o transforma em um objeto, o que diminui consideravelmente o número de erros – uma vez que existiria uma padronização das entradas textuais. O 'filtro' transforma letras *uppercase* em *lowercase* padrão e também retira acentos (apesar de podermos assumir, pela proposta, que o texto não conteria acentos) e outras inconsistências textuais comuns em buscas, como sinais de pontuação, sem perda de qualidade e precisão da consulta. Após o processamento, o objeto resultante é encaminhado para o índice invertido, a segunda classe principal. O índice invertido armazena os dados mapeando os termos às suas ocorrências em um determinado arquivo. Implementamos no através da classe *InvertedIndex*, criada com base no link disponível no GitHub do projeto, sob a seção de referências.

A Estrutura de Dados utilizada no projeto é o dicionário, disponível na linguagem Python. O dicionário é bem semelhante à lista no sentido de que, assim como ela, é também um conjunto de objetos. É uma estrutura de dados muito flexível, pois pode ser editada *on-the-go,* pode ter seu tamanho reduzido ou aumentado dependendo das necessidades do programa e também pode conter outras listas no seu interior. A diferença do dicionário para a lista, e o motivo principal de sua utilização como estrutura de dados principal do projeto é o seu acesso – no dicionário, os elementos de seu interior são acessados através de chaves, o que torna uma busca muito fácil.

A ordenação dos dados foi feita através do *cosine ranking*, onde existe uma classificação de dois vetores de acordo com o grau de similaridade entre eles.

Para saber se o programa estava funcionando de acordo com as expectativas, utilizamos a database para testes disponibilizada pelo professor na plataforma do Moodle (*The 20 Newsgroups data set*) de modo a obter consistência com os resultados esperados. Um link para a database está disponível no GitHub do projeto.

O projeto foi realizado na linguagem Python devido à sua ampla gama de bibliotecas, que possibilitaram um projeto enxuto e eficiente. Durante a construção do trabalho, aprendemos bastante sobre coisas que não teríamos acesso normalmente durante a grade curricular, como a linguagem Python, a utilização do GitHub e como trabalhar em equipe em um ambiente de desenvolvimento de software. Por isso, agradecemos a oportunidade de poder praticar tudo o que foi visto em sala e colocar esse conhecimento no mundo real, nos preparando para o mercado de trabalho futuramente.