

ATIVIDADE SOMATIVA 2

Olá estudante,

Chegou o momento de finalizarmos o nosso projeto. Retomaremos o grafo da semana 4, e agora, implementaremos o algoritmo de Dijkstra para determinar o caminho mais curto ou mais relevante entre diferentes livros em um grafo.

Esta fase do projeto não só fortalece a compreensão de algoritmos avançados e estruturas de dados, mas também ilustra como a teoria pode ser aplicada para melhorar sistemas de recomendação em aplicações práticas.

O que você deve desenvolver?

1. **Implementação do algoritmo de Dijkstra:** use Java para implementar o algoritmo de Dijkstra, adaptando-o para funcionar com um grafo de livros que criamos na atividade somativa da semana 4.
2. **Determinação de caminhos relevantes:** aplique o algoritmo para encontrar os caminhos mais curtos no grafo, que podem indicar recomendações interessantes para os usuários com base em suas preferências e interesses.
3. **Aplicação prática:** integre essa funcionalidade ao sistema de biblioteca virtual para oferecer mostrar quais são os livros com a menor distância para um livro qualquer. Quanto menor a distância, mais próximo este livro é, e maior é a chance de ser uma boa recomendação. Use o algoritmo a seguir como base para o seu trabalho:

```
public static Map<Livro, Integer> djikstraSimples(HashMap<Livro, Set<Livro>>
grafo, Livro origem) {

    Map<Livro, Integer> distancias = new HashMap<>();

    Queue<Livro> fila = new LinkedList<>();

    distancias.put(origem, 0); // como não temos pesos entre os nós, o peso padrão
    é 0
```

```

        fila.add(origem);

    }

    while (!fila.isEmpty()) {

        Livro atual = fila.poll();

        int distanciaAtual = distancias.get(atual);

        for (Livro vizinho : grafo.getDefault(atual, new HashSet<>())) {

            if (!distancias.containsKey(vizinho)) {

                distancias.put(vizinho, distanciaAtual + 1);

                fila.add(vizinho);

            }

        }

    }

    return distancias;

}

```

O que eu devo entregar?

Precisaremos do código-fonte do seu projeto dentro de uma pasta zipada. Isso significa que precisaremos de todos os arquivos com final *.java* que você criou. Os arquivos com final *.class* **não devem** ser incluídos.