Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Engenharia de *Software* – *Campus* Lourdes
Algoritmos e Estruturas de Dados II (Laboratório)
Prof. Glender Brás – Semestre 2/2025
Atividade: Estruturas de Dados de Busca em Sistemas de *Software* –
Valor: 3 pontos – **Data:** 01/12/2025

INSTRUÇÕES

- Esta é uma **atividade individual**, em laboratório, da disciplina "Algoritmos e Estruturas de Dados II".
- Utilize seu repositório, criado no **GitHub Classroom**, para resolver as questões de programação propostas e enviar suas respostas para correção. **Nenhuma outra forma de entrega da atividade será aceita**.
- O prazo **limite para entrega** é o dia **01/12**, até final da aula. *Commits* no repositório com data posterior a esse limite serão desconsiderados no momento da correção da atividade.
- Além do **conteúdo do seu repositório de trabalho** e dos **materiais disponíveis no Canvas citados na Tarefa 0**, **não é permitida a consulta a nenhum outro material**, porém a professora está **disponível para esclarecer dúvidas** e debater propostas de soluções vindas dos alunos.
- Sendo uma atividade individual, **espera-se dos alunos comprometimento e ética** na produção de sua solução. Soluções produzidas em conjunto, entregas contendo cópias, mesmo que parciais, em código ou estrutura, e consultas a materiais e ferramentas não permitidos são **faltas graves**, **resultando em nota 0 para todos os envolvidos** e posterior **notificação à Coordenação do Curso de Engenharia de *Software*** por desrespeito às normas acadêmicas.

Tema: Busca e relatórios em sistemas de *software*

Temos nosso sistema de comércio de produtos em que seus dados são armazenados em estruturas de dados apropriadas para buscas. Nosso sistema já gera alguns relatórios relativos a produtos e pedidos. Nesta última atividade acrescentaremos ao nosso sistema de comércio uma nova entidade – fornecedor – e desenvolveremos relatórios sobre fornecedores e produtos vendidos por eles.

Tarefa 0: (preparação)

- **Adicione uma nova ramificação** (*branch*) no seu repositório GitHub Classroom correspondente a esta atividade, com o nome ***atividade0112***, a partir de seu *branch* anterior. Certifique-se de que as classes ***ABB***, ***AVL***, ***No***, ***TabelaHash***, ***Entrada***, ***Lista*** e ***Celula*** estejam neste novo *branch*.
- Em seguida, adicione à pasta raiz do projeto o arquivo "**fornecedores.txt**" fornecido no Canvas.

Tarefa 1: (0,5 ponto)

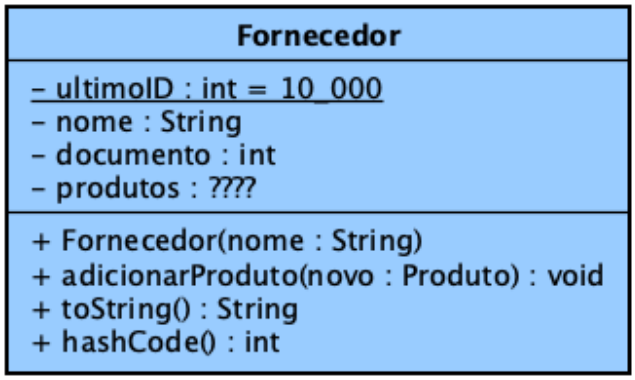
Implemente o código da classe "Fornecedor", de acordo com o diagrama de classes UML ao lado. Perceba que ainda é necessário escolher uma estrutura de dados adequada para armazenar todos os produtos vendidos pelo fornecedor.

O construtor da classe cria um novo fornecedor a partir do nome informado. Esse nome deve conter, pelo menos, duas palavras; caso contrário, a exceção ***IllegalArgumentException*** é lançada. O fornecedor criado recebe um número de documento gerado sequencialmente a partir do contador estático da classe.

O método ***adicionarProduto*** recebe um produto como parâmetro e deve inseri-lo na estrutura de dados escolhida por você para armazenar os produtos de um fornecedor. Não podem ser armazenados produtos nulos, caso em que uma exceção é lançada.

O método ***toString*** retorna uma representação textual do fornecedor, incluindo seu nome, documento e o histórico de produtos associados.

Por fim, o método ***hashCode*** retorna um código *hash* para o fornecedor que, neste caso, corresponde ao documento do fornecedor.



Tarefa 2: (0,5 ponto)
Na classe ***App***, declare e instancie a árvore balanceada AVL em que os fornecedores devem ser inseridos e a tabela *hash* que associará um produto aos seus fornecedores.

Tarefa 3: (1 ponto)
Na classe ***App***, implemente o método ***static <K> AVL<K, Fornecedor> lerFornecedores(String nomeArquivoDados, Function<Fornecedor, K> extratorDeChave)***. Esse método lê os dados de um arquivo-texto, cujo nome é fornecido como parâmetro, e retorna uma árvore balanceada AVL de fornecedores. O arquivo-texto de fornecedores deve estar no formato:

- N (quantidade de fornecedores)
- nome do fornecedor

Deve haver uma linha para cada um dos fornecedores.
No momento da criação de um fornecedor, já devem ser selecionados aleatoriamente até 6 produtos para serem indicados como sendo os produtos vendidos pelo fornecedor em questão. Adicionalmente, o fornecedor criado deve ser associado aos seus produtos na tabela *hash* que relaciona um produto aos seus fornecedores.
Retorna uma árvore com os fornecedores carregados, ou vazia em caso de problemas na leitura do arquivo.

Tarefa 4: (1 ponto)
Na classe ***App***, implemente os códigos dos métodos:

- ***relatorioDeFornecedor***, que deve retornar o relatório completo de um fornecedor escolhido pelo usuário por meio de seu documento identificador.
- ***fornecedoresDoProduto***, que deve gerar, em arquivo, um relatório com todos os fornecedores de um produto escolhido pelo usuário.

Faça ainda as modificações necessárias no ***menu*** e no método ***main*** para que essas opções estejam disponíveis ao usuário da aplicação.

Instruções e observações:

- O projeto deve estar hospedado na tarefa correspondente do GitHub Classroom. Endereço para aceitar a tarefa:
  - **G2 – 7:00:** https://classroom.github.com/a/ETHY6kWq
  - **G1 – 10:50:** https://classroom.github.com/a/DY9IP6XV

- As atividades pontuadas da disciplina podem depender direta ou indiretamente dos códigos desenvolvidos nas aulas. Portanto, é essencial o comprometimento no acompanhamento das atividades semanais.
- Para a correção das atividades pontuadas, serão considerados todos os *commits/pushes* realizados ao longo das semanas, não somente o último com a resposta final do exercício.