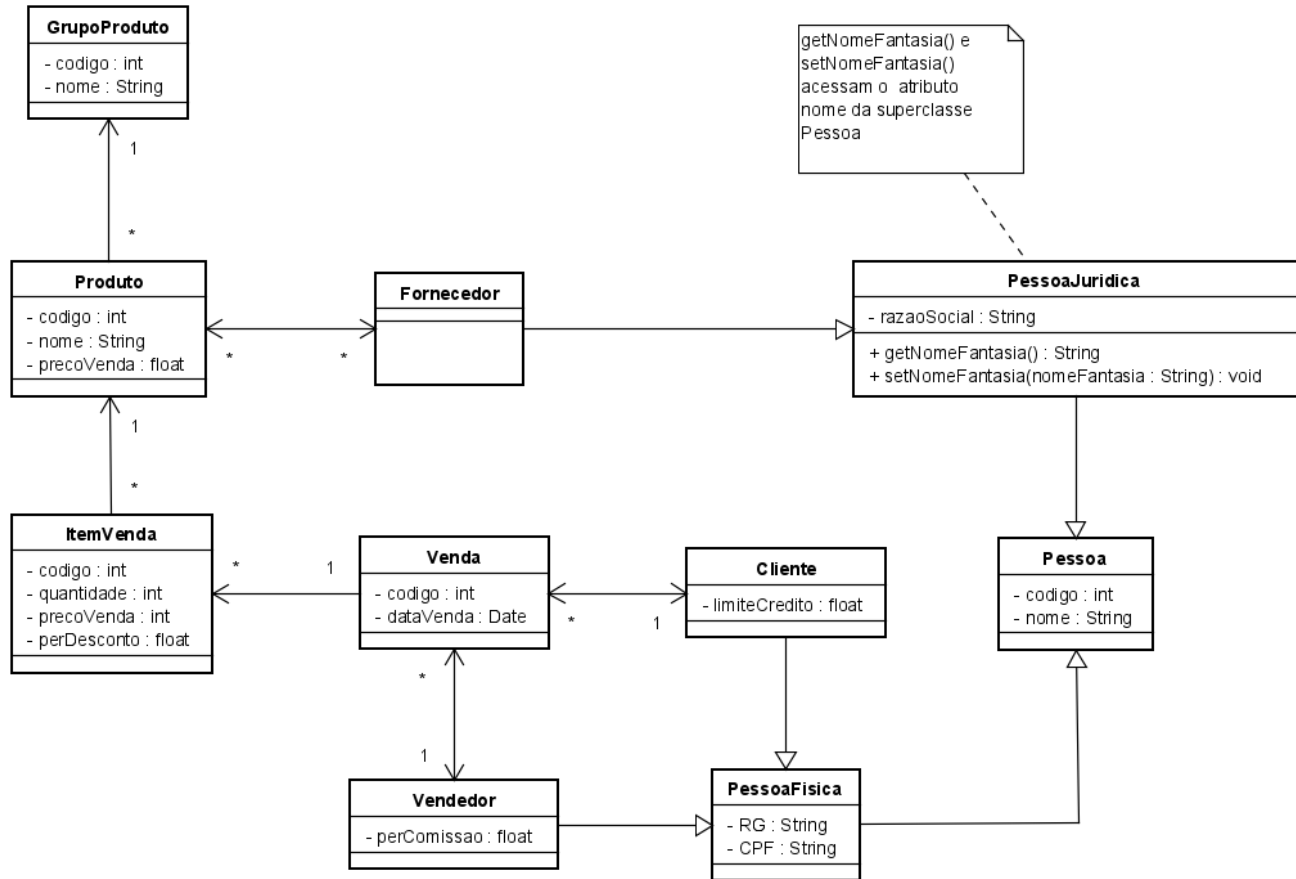


Exercício Mapeamento de Herança com JPA

Considere o diagrama de classes a seguir.



1. Crie um projeto Java Application com o nome **ProjetoMapeamentoRelacoes02**
2. Configure uma unidade de persistência JPA para manipular uma base de dados em um SGBD PostgreSQL com o nome: **vendas2**
3. Implemente em um pacote **vo** cada uma das classes do diagrama como um VO
4. Faça as anotações JPA conforme os relacionamentos e cardinalidades
5. Implemente em um pacote **apps** classes que façam a persistência de instâncias das classes VOs, sendo necessário a inclusão de, pelo menos, três instâncias de: **GrupoProduto**, **Produto**, **Fornecedor**, **Cliente** e **Vendedor**. Para **Venda**, deve persistir pelo menos cinco instâncias.
6. Implemente em um pacote **consulta**, uma classe para cada uma das seguintes consultas:
 - a. Listar todos os fornecedores cadastrados, ordenados por razão social (código e nome) e os produtos que ele fornece (código, nome do produto);
 - b. Listar todos os produtos cadastrados, ordenados por nome (código, nome, preço de venda, nome do grupo de produto e seus fornecedores (código e razão social));

Exercício Mapeamento de Herança com JPA

- c. Listar todos os clientes cadastrados, ordenados por nome (código, nome, quantidade de vendas que já foram realizadas para ele).
- d. Listar todas as vendas cadastradas, ordenadas por data da venda (código, data da venda, valor total da venda) e seus itens (código, quantidade, preço venda, percentual de desconto)
- e. Listar todos os vendedores cadastrados, ordenados por nome (código, nome, valor de comissão de cada venda que ele realizou (código da venda, valor total da venda, valor comissão na venda))

Obs1: além dos atributos sugeridos e métodos getters e setters, podem ser criados outros métodos, se necessário.

Obs2: os dados a serem inseridos poder ser estabelecidos de forma literal no código;