

PROJETO 1 / PARTE 2 – RELACIONAMENTO 1:N (3 pontos)

Implementar um relacionamento 1:N. Isso exigirá a criação do CRUD de uma segunda entidade. Nesta tarefa, vocês devem implementar um CRUD de **categorias** de produtos (Eletrônicos, Casa, Brinquedos, Escritório, Vestuário, etc.). Essa entidade precisará apenas de um atributo que contenha o nome da categoria.

Em seguida, vocês devem alterar a entidade de produtos para que contenha mais um atributo que contenha o ID da categoria a que cada produto pertence. A consistência deve ser mantida em todas as operações de inclusão, alteração e exclusão de produtos.

Também deve ser possível emitir a lista de produtos de uma determinada categoria.

ROTEIRO PARA IMPLEMENTAÇÃO

1. Acrescentem, à entidade Produto, um atributo `idCategoria`.
2. Criem um CRUD para as categorias. É um CRUD simples e bem semelhante ao CRUD anterior.
3. Atualizem o menu principal para que seja possível acessar todas as operações dos CRUDs de produtos e categorias.

Neste momento, vocês têm dois CRUDs que atendem às operações necessárias, mas ainda não é possível garantir a consistência entre eles, isto é, que um produto só seja cadastrado em categorias válidas. Para isso, precisaremos, a cada vez que um `idCategoria` for digitado em um produto, ver se é um ID válido. Da mesma forma, na hora de apresentar um produto, precisaremos apresentar também o nome da categoria a que ele pertence.

Essas alterações não precisam ser feitas na classe `Arquivo` nem nas classes `Produto` ou `Categoria`, mas no método principal, que contém as operações de entrada e saída de dados.

4. Nos métodos de inclusão e de alteração de produtos, após o usuário digitar o ID da categoria, usem o método de busca do arquivo de categorias e chequem se alguma categoria válida é retornada. Se houver uma categoria válida, apresentem o nome dessa categoria. Caso contrário, informem que a categoria não existe e solicitem a digitação de um novo ID de categoria.
5. Nos métodos de busca e de listagem de produtos, a apresentação dos produtos não poderá mais ser feita por meio de um método `toString()` da classe `Produto`, pois será também necessário usar dados da categoria. Assim, a apresentação dos dados do produto deverá ser feita localmente, atributo por atributo. Nesse processo, ao escrever o `idCategoria`, façam a mesma busca do passo anterior e informem o nome da categoria do produto.

Ainda precisamos garantir que uma categoria não será excluída se o seu ID estiver sendo usado por algum produto. Apesar de ainda não termos estudado uma estrutura de dados eficiente para isso, vamos fazer o processo da forma mais simples (e lenta).

6. Criem uma cópia do método `listar()` da classe `ArquivoIndexado` (ou de outra classe que vocês tenham feito) que receba um ID de categoria como parâmetro e que use esse ID para filtrar os produtos listados, isto é, esse novo método retornará uma lista dos produtos que pertencem a uma determinada categoria. Assim, a categoria só poderá ser excluída se a lista retornada estiver vazia.

PROJETO 1 / PARTE 3 – RELACIONAMENTO N:N (4 pontos)

Nesta tarefa, vocês devem criar um CRUD para a entidade `Cliente`, que terá pelo menos três atributos (ID, nome e email do cliente). Vocês deverão, por meio das estruturas auxiliares (outros arquivos), registrar as compras de produtos por esses clientes.

Para isso, precisaremos de mais uma entidade: `Compra`. Essa entidade terá um relacionamento 1:N com a entidade `Cliente`, pois um cliente poderá comprar várias vezes. Seus atributos deverão conter, pelo menos, os seguintes atributos: ID da compra, ID do cliente, data da compra e valor total da compra.

Cada `Compra` poderá conter vários produtos, mas pelo menos 1. Não deve haver limite superior. De forma análoga, um produto poderá aparecer em várias compras. É por isso que chamamos o relacionamento de N:N, pois N produtos podem aparecer em N compras.

Enfim, deve haver alguma nova entidade (ex.: `ItemComprado`) que relacione cada produto comprado a uma compra específica. Assim, essa entidade deverá conter os seguintes atributos: ID, ID da compra, ID do produto, quantidade do produto e preço unitário. Observem duas questões importantes aqui: (1) é necessário registrar o preço unitário, porque o preço do produto pode alterar depois da compra e, nesta compra, o preço é aquele pago à época, e (2) cada instância dessa entidade deve ter seu próprio ID, para que possamos fazer alterações e exclusões posteriores.

Novamente, a consistência dos dados deve ser preservada em todas as operações.

OBS.: Uma penalidade de 0,5 pontos será aplicada a cada dia corrido de atraso na entrega.

ROTEIRO PARA IMPLEMENTAÇÃO

1. Criem o CRUD de cliente.
2. Criem o CRUD de compra, estabelecendo um relacionamento 1:N com o cliente.
3. Alterem o CRUD de compra para que ele permita a gestão de produtos comprados (na entidade `ItemComprado`). Se acharem mais fácil, podem fazer um CRUD independente (apesar de não ser a forma como vemos nos sistemas reais). Em outras palavras, a gestão de produtos em uma compra usualmente é feita durante a inclusão ou manutenção de compras.
4. Para vocês saberem quais produtos pertencem a uma compra ou quais compras foram feitas de um produto, vocês precisarão usar duas árvores B+ de chave composta. Essas árvores, da mesma forma que os demais arquivos, deverão ser abertas no método principal. A primeira árvore será usada para armazenar os pares de chaves (`idCompra`, `idProduto`). A segunda árvore será usada para armazenar os pares de chaves (`idProduto`, `idCompra`). Como o relacionamento entre os IDs é unidirecional, precisamos de 2 árvores mesmo.

Assim, a cada inclusão de um registro de item comprado, vocês devem inserir o par de chaves (`idCompra`, `idProduto`) na primeira árvore B+ e o par de chaves (`idProduto`, `idCompra`) na segunda árvore. Da mesma forma, a cada exclusão de registro de item comprado, vocês também devem fazer as exclusões nessas árvores.

5. Agora, façam o ajuste nos métodos de exclusão de compras e de produtos, de tal forma que não seja possível excluir compras ou produtos que estiverem registrados no arquivo de itens comprados. Para isso, basta usar novamente o método `lista()` da Árvore B+. Se esse método retornar algo diferente de uma lista vazia, então a compra ou o produto está sendo usado em algum registro de item comprado.
6. Da mesma forma, faça o ajuste no método de exclusão de clientes para que não seja possível excluir clientes que tenham compras registradas no arquivo.

PROJETO 1 / PARTE 4 – Relatórios (3 pontos)

Para concluir o primeiro projeto, vocês devem criar os relatórios das compras (listagem em tela). Esses relatórios devem apresentar os produtos comprados por um cliente e os clientes que compraram um produto. Vocês também devem criar um terceiro relatório entre as seguintes opções:

- a. Um relatório que mostre os N produtos mais vendidos (e a quantidade de cada um)
- b. Um relatório que mostre os N melhores clientes (que mais gastaram em compras)
- c. Um relatório que mostre a quantidade de vendas (valor total) por mês
- d. Um relatório que mostre os valores vendidos por categoria de produto

Vocês ainda podem propor outros relatórios que não estejam nessa lista (desde que a proposta seja aprovada previamente pelo professor).