

Estoque com Produtos Perecíveis

Agora você terá que criar uma nova versão para nosso sistema de Controle de Estoque.

Inicialmente pedimos que você crie um pacote com o nome “estoqueComProdutoPerecivel” para implementar esta versão do trabalho.

Reforçamos também que você deverá submeter a pasta com o seu nome, compactada, dentro da pasta src do projeto. Exemplo: Poo20212/src/pedroSantosNeto essa pasta como nome “pedroSantosNeto” é que eu deveria compactar e enviar como trabalho. Ela vai incluir diversas outras pastas, incluindo estoque e estoqueComProdutoPerecivel.

Neste trabalho a grande diferença é a presença de um novo tipo de produto, denominado ProdutoPerecivel. Esse produto tem uma característica especial: tais produtos devem possuir lotes internamente e cada lote deve possuir a indicação da quantidade de produtos no lote e sua data de validade.

Um ProdutoPerecivel tem tudo que um produto tem. Mas a compra de produtos deve ser feita incluindo lotes internamente ao produto. O método compra da classe Estoque também foi alterado. Agora ele exige que seja sempre passado a data de validade do produto. Caso a data seja nula, indica que o produto não tem validade.

A compra deve continuar atualizando o preço de compra, bem como o preço de venda (via acréscimo do lucro).

A venda também terá uma mudança importante: no caso de venda de produtos perecíveis devemos sempre começar vendendo o mais próximo do vencimento. Se um lote com 10 produtos for o mais próximo do vencimento e tiver sido solicitado a venda de 15 produtos, devemos vender esse lote inteiro e depois procurar o próximo lote disponível para realizar a venda do restante dos produtos.

A venda só pode ser feita para produtos dentro da validade. Não é permitida a venda de produtos fora da validade. Também não devemos vender produtos perecíveis se não houve a quantidade solicitada de itens válidos.

É possível saber a quantidade de itens vencidos de um produto. Deve ser criado um método específico para isso, listado a seguir.

Não deve ser possível cadastrar um produto com dados inválidos. Isso significa que os campos numéricos devem ser maiores ou igual a zero, os campos de texto devem ser preenchidos.

Métodos Classe Estoque Atualizado

```
public interface InterfaceEstoque {
```

```
    // Retorna false se houver algum campo numérico invalido ou texto vazio ou null ou produto já cadastrado.
```

```
    public boolean incluir(Produto p;
```

```

        // Retorna false se o produto não existir, se houver valores numéricos inválidos ou se enviar data
        e o produto comprado não for perecível
        public boolean comprar(int cod, int quant, double preco, Date val);

        // Retorna -1 se o produto não existir, se for vencido ou se houver um dado inválido em números.
        public double vender(int cod, int quant);

        // Retorna null se o produto não existir
        public Produto pesquisar (int cod);

        // Retorna lista de produtos abaixo do mínimo. Lista vazia se não tiver nenhum.
        public ArrayList<Produto> estoqueAbaixoDoMinimo();

        // Retorna lista de produtos vencidos. Lista vazia se não tiver nenhum.
        public ArrayList<Produto> estoqueVencido();

        // Retorna 0 se não houver nenhum produto vencido ou a quantidade de produtos vencidos de
        um determinado código.
        public int quantidadeVencidos(int cod) throws ProdutoInexistente;
    }

```

Construtores:

```

public Produto(int cod, String desc, int min, double lucro, Fornecedor forn);
public ProdutoPerecivel(int cod, String desc, int min, double lucro, Fornecedor forn);
public Lote(int quant, Date validade);

```