Classes

Existem as classes Vendedor, Produto e Oferta.

A classe Vendedor tem como atributos:

* Nome
* Nome de Usuário
* Data de Nascimento
* Endereço físico completo (rua, número, complemento, bairro, cidade, estado)
* Endereço eletrônico
* Telefone (poderia ser mais de um número de telefone)
* Produtos (lista de produtos à venda)
* XP (que seria a "vida" do vendedor)

Nota (média das notas atribuídas pelos compradores. Só há um porém, como controlaremos quem poderá votar? Já que não há cadastro de compradores.)

OBS. : dependendo da forma que iremos implementar, podemos fazer que o array de Produtos seja de todos os produtos (à venda ou não).

OBS. 2: Como temos em 'estado' um número finito e reduzido, poderíamos implementá-los na forma de lista na GUI.

A classe *Produto* tem como atributos:

* Nome
* Descrição
* Categoria (apenas uma por produto)\*
* Preço
* Quantidade\*\*
* Tempo de venda (apenas como informação extra, pelo menos até agora. Por comodidade seria usado a API Time)
* Estado (indicaria se o produto foi vendido ou ainda está à venda. Seria interessante se usássemos o array para todos os produtos. Sendo assim, seriam apenas removidos os produtos, pelo Vendedor, que por algum motivo não se queira mais vender. Seria booleano, true para vendido, false para à venda)

\* A categoria de um produto deve ser selecionada a partir de uma lista previamente definida.

\*\* Seria necessário para evitar a burla da venda. O Vendedor assim, por restrições do sistema, não poderia cadastrar um mesmo produto com mesmo nome. Pois poderia se criar alguns produtos iguais, "vender" um produto, aumentar seus XPs e permanecer com os outros na loja e serem facilmente encontrados. Caso o Vendedor tenha mais de um produto daquele tipo, deveria ser indicado pela quantidade de itens daquele produto. Veja mais em "Mecanismo para evitar burla ao sistema" na categoria XP.

OBS. : a quantidade pode ser "livre" ou restrita (por exemplo, até 2 a 5 itens do mesmo produto). Caso restrita, é mais um mecanismo para evitar a burla, pois o propósito da aplicação é que o Vendedor ponha aquilo que já não mais lhe interessa, como o OLX, não o Mercado Livre, na qual há lojas que possuem estoque que pode ser renovado. A questão de não se criar dois produtos com nomes iguais reside no fato que isso dificultaria a burla do sistema, pois dificultaria para o comprador encontrar o produto pelo seu nome.

OBS. 2: a cada venda de um produto a quantidade decrementaria. Seria permitido a mudança da quantidade do produto até antes da primeira venda.

OBS. 3: Poderíamos adicionar tb. como atributo uma imagem do produto. Caso não adicionado pelo Vendedor, se usaria uma imagem padrão.

A classe *Oferta* tem como atributos:

Herda de Produto. Além da adição de ‘porcentagem’ (desconto em relação ao valor original), tempo (de permanência da oferta).

**XP**

Tem de ficar bem definido o que seria os XPs e para que serviriam. Quais são os métodos de ganhá-los.

*Ganharia XP por:*

* Cadastrar-se como vendedor (5 a 30, a depender da "cotação" do XP. Se pra conseguir algo precisa de uma quantidade enorme de XPs, ex.: 250, ou nem tanto, ex.: 75.)
* Incluir algum novo produto (10 a 20) \*\*\*
* Venda (tem que criar um método para que se evite burla) \*\*\*\*

\*\*\* a retirada de um produto não vendido acarreta na perda dos pontos recebidos. Assim como a retirada de produtos vendidos, pois poderia ser uma formar de burlar o sistema pois poderia usar o mesmo nome. Assim, pode-se retirar, mas perderá pontos de cadastro e venda.

\*\*\*\* a depender do valor da venda, os XPs recebidos seriam diferentes. Quanto mais caro o produto, mais XPs se ganha.

*Se faria com o XPs:*

Uma melhor classificação no sistema de vendas.

*Mecanismos para evitar burla ao sistema XPs:*

Uso de uma fórmula (a ser pensada) para evitar a burla ao sistema. Para isso deveria seguir as seguintes regras:

* Considere o tempo. O motivo de considerar o tempo é para evitar que após colocado à venda, o produto seja "vendido" de forma rápida. Não seria negado XPs, apenas receberia menos. Poderia se pôr tb. um limite máximo de XPs para que não permanecesse por muito tempo na loja.
* Quantidade de itens de um produto. Apesar de ser contraditório para um sistema de vendas, quanto mais itens de um mesmo produto de um Vendedor, ele receberá menos por cada uma das vendas. Isso é para evitar que sempre se coloque uma quantidade maior de itens de um produto do que se tem. Por exemplo, tenho 2 DVDs que quero pôr à venda, mas ponho 3, "vendendo" um, consequentemente ganhando XPs e ficando com os 2 para venda, mas vou sair perdendo um pouco pois não terá o valor máximo que quando apenas vende-se um item de cada produto.
* Quando há mais de um item para venda de um produto, o Vendedor terá de esperar a venda de todos os itens para receber todos os XPs da venda.
* Se após a venda de um item de um produto, outro item desse mesmo produto for vendido em menos de X horas, o valor final do XP será diminuído. Poderia prejudicar alguns, mas no geral evitaria a burla ao sistema.
* Após trocado o estado de um item para "vendido", não poderá mais voltar para "à disposição". Para evitar que possa "vender" várias vezes o mesmo item, ganhando XPs.
* Após uma quantidade Y de tempo (30 dias por exemplo), poderia se retirar uma porcentagem (10%) ou número absoluto (200) dos XPs de um Vendedor. Pois, pelo sistema, para ter uma ótima venda ele deveria vender algo em torno do 5° dia e após a venda do primeiro esperar no mínimo algumas horas para vender o outro. Logo, demandaria tempo, mas como perderia XPs após um mês, pensaria duas vezes, pois poderia ser inútil, principalmente, se usarmos porcentagem.
* "Instruiríamos" os compradores após a compra de um produto, pedir a alteração de seu estado para vendido, para que não permanecesse por mais tempo do que se demorou para a venda.

OBS. : Todos esses mecanismos devem ser debatidos.

**Filtros**

Os resultados da busca dos filtros apenas retornam aqueles produtos que estejam com estado "à venda" (ou seja, os "vendidos", logicamente, não seriam retornados). Nós podemos usar os seguintes filtros:

* Por nome do produto
* Por categoria
* Por faixa Repositório de preço
* Por local (cidade do Vendedor)\*\*\*\*\*
* Por vendedor\*\*\*\*\* [decidido que não colocaria no projeto, pois ninguém que procure um produto x mais barato procurará apenas por um vendedor]

OBS. : os filtros poderiam ser combinados para um melhor resultado.

OBS. 2: poderíamos fazer que filtros fossem uma classe, que pesquisasse nos repositórios de Vendedor e Produto (a depender do filtro selecionado). Dentro dessa classe existiria os tipos de filtro que definirmos.

OBS. 3: os filtros devem levar em consideração os XPs dos vendadores.

\*\*\*\*\* Temos de definir se são importantes para a aplicação.

**Repositório**

Seria um repositório para cada classe básica, exceto, caso os filtros sejam uma classe, pois a meu ver não há necessidade para tal, e Oferta, caso seja apenas uma por período (dia, horas).

**Exceções**

* ProdutoJaCadastradoException
* CpfJaCadastradoException (não se pode cadastrar dois vendedores com o mesmo CPF)
* NaoEncontradoProdutosException
* NaoEncontradoVendedorException
* NomeUsuarioJaCadastradoException