Projeto de Laboratório - Programação II

Na **US1** implementamos a classe abstrata Item, que representa a base para os tipos de produtos compatíveis com a especificação. Posteriormente criamos três subclasses de Item: ProdutoNaoIndustrializadoPorQuilo, ProdutoPorUnidade e ProdutoQuantidadeFixa. Estas são responsáveis por atender a necessidade da geração dos diversos tipos de representação textual dos itens compráveis, além de possuir diferentes comportamentos de acordo com o seu tipo.

Criamos também as classes SistemaController, Facade e ListaPraMim responsáveis por: criar, pesquisar, atualizar e deletar itens, fazer a conexão entre o backend e a interface com usuário e oferecer um contrato de tudo que o sistema deve oferecer, respectivamente. Como cada Item tem um identificador único, foi natural optar por usar Map – mais especificamente HashMap - para armazenar os itens cadastrados no sistema. O hashMap de Itens associa para cada Item(valor) ao seu identificador único(chave).

Na **US2** implementamos diferentes funcionalidades de listagem de itens. Essas listagens podem seguir diferentes tipos de ordenação: por cadastro, ordem alfabética, menor preço e por nome. Para fazer a ordenação foi necessário criar alguns comparadores: *ComparaCategoria* que compara dois itens baseando-se em suas categorias, *ComparaNome* que compara dois itens pelo nome, *ComparaValor* que compara dois itens de acordo com o seu menor preço disponível em locais de compra.

Na **US3** criamos as classes Compra e ListaDeCompras, a primeira é responsável por representar uma compra, possuindo um Item associado e uma determinada quantidade, a segunda por manipular as compras em um determinado estabelecimento. Para armazenar as Compra(s) em ListaDeCompras utilizamos HashMap, pelo mesmo motivo do uso na US2, que associa cada Compra(valor) ao id do item associado(chave). Além disso, a classe Compra tem uma própria representação textual que é necessária para listagem de compras e também possui um método próprio de comparação. Decidimos armazenar as listas de compras em um HashMap dentro de SistemaController, que associa cada ListaDeCompras(valor) a seu descritor(chave).

Por fim, para ordenar as listas em 2 níveis, por categoria e por nome, implementamos os métodos getCategoria() e compareTo() em Compras, responsáveis pela lógica de ordenação.

Na **US4** foram implementados diferentes métodos para pesquisar em lista de compras. Para que atendesse as diferentes tipos de ordenações.

Na **US5** foram criado métodos para geração automática de lista de compras. Essas gerações podem seguir três estratégias: criando uma cópia da mais recente, a última lista que contem uma compra e uma lista que contem os itens mais comprados.

Na **US6** implementamos métodos que sugerem um melhor estabelecimento para as compras de um determinado item e métodos para a listagem dessas listas. Além disso, foi criado a classe *Estabelecimento* para armazenar todos as compras que possuem em um local de compra. Nessa Us foi adicionada uma funcionalidade para ordenação desses estabelecimentos. Logo, foi necessário a criação de um método *CompareTo para comparar o estabelecimento* de acordo com o valor final resultante de todas as compras que ele possui.

Na **US7** implementamos métodos responsáveis pelo armazenamento e carregamento dos dados do sistema.

Em todas as US's foram criados testes de unidade pegando os casos limites de suas especificações.