

Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e

Informática

Unidade Acadêmica de Sistemas e

Computação

CC - Ciências da Computação - Laboratório de Programação II
Período 2018.1

Apresentação:

O desenvolvimento do projeto consta a partir da ideia de uma Lista de compras. Onde tem-se o intuito de como o próprio nome sugere, a criação de listas de compras, onde nelas, são anexadas o lugar, o preço e quantidade dos itens. Finalizando a ideia da lista, a mesma ainda indica qual local será mais favorável para realizar tais compras.

Tal projeto foi realizado/idealizado com a separação de afazeres, onde cada membro tornou-se responsável por tais classes e seus testes. O projeto foi desenvolvido usando o melhor Design que foi-se pensado.

Sprints:

O projeto foi dividido em 3 sprints, onde em cada um deles, conseguimos manter a pontualidade e funcionalidade(seguinto o uso do EasyAccept), atrelado à entrega.

Casos de uso:

Caso_1: O primeiro caso de uso foi para fazer “a base”, ou seja a criação, leitura, atualização e remoção, de itens(*Class Item*) em uma lista, onde a *Class Item*, é uma classe abstrata e alguns de seus métodos são sobrescritos, por tais classes que representam os itens quando se relacionam a Quilo, Unidade e Quantidade. Nesse caso, foi utilizado o padrão de heranças (nos tipos de itens) e composição (Item em ItemController). Aplicamos o que a especificação desejava, logo, não se é possível adicionar itens que sejam iguais no sistema.

Caso_2: No segundo caso, foi implementado classes para ordenação utilizando a coleção Comparator. Foram adicionados métodos nas classe ItemController e Facade.

Caso_3: No terceiro caso foi proposto a criação de listas(*Class Lista*) onde sua principal característica é um descritor e compras(*Class Compra*). Nesse caso utilizou-se um LinkedHashMap para guardar as compras. As listas quando geradas, insere automaticamente a data/hora da compra realizada. Foi utilizado composição (Lista em ListaController; Compra em Lista; Item em Compra).

Caso_4: No quarto caso foi desenvolvido para a abordagem das listas serem pesquisadas através de seus respectivos descritores. Ou através da data inserida automaticamente, logo métodos foram adicionados à *Classe Lista* e ListaController. Nesse caso, foram utilizados alguns Comparators para ordenação de listas.

Caso_5: No caso de uso 5, foi desenvolvido os métodos para criação de listas de compras automáticas, inseridos na classe ListaController, criando, também, métodos auxiliares na classe Lista.

Caso_6: No caso de uso 6, o desenvolvimento foi voltado à verificar qual o melhor estabelecimento para realizar as compras da lista de compras. Foi desenvolvido a classe Supermercado, fazendo uso de composição de Compra em Supermercado. Foi utilizado ArrayList para armazenar as compras em Supermercado.

Caso_7: No sétimo caso, utilizamos dos recursos de Java, para a leitura e escrita de arquivos, assim salvando todas as operações, funcionando como um “log” do sistema.

Responsáveis:

- **Euclides Ramos:**

Facade (caso01 ao caso06), ItemController, ListaController (caso03, caso06), OrdenaçãoDefault (caso04), Validador, Categoria, Supermercado, Relatório.

- **Edson Wesley:**

Comparators, ListaController (caso06), Lista, Relatório.

- **Eduardo Pereira:**

Item, itemQuilo, itemQuantidadeFixa, itemUnidade, Compra, ListaController (caso05), Facade (caso06, caso07).

- **Joao Antonio:**

OrdenaAlfabetica, OrdenaPreço, OrdenaListaData, ListaController(caso04).

Repositório: <https://github.com/EuclidesRamos/ProjetoLP2>