UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA - CEEI
DISCIPLINA: TÉCNICAS DE PROGRAMAÇÃO
PROFESSOR: DR. MARCUS SALERNO DE AQUINO

DÉBORA DA SILVA COSTA

NOÊMIA CÍNTIA SALES SANTOS DA SILVA

RAFAEL DOS SANTOS LIMA

SABRINA ARAÚJO CARDOSO

PROJETO: CONTROLE DE ESTOQUE DE UMA CERVEJARIA ARTESANAL: BEER'S ++

Maio/2021

Campina Grande - PB

1. Introdução

1.1. Objetivo

Implementar um projeto que faça o controle de estoque e vendas de uma Cervejaria Artesanal usando a linguagem de programação C++.

1.2. Resumo do Projeto

Inicialmente optamos por trabalhar com dois tipos de vendas: por unidade e por litro. Dessa forma, assim que o programa inicia pergunta-se ao usuário que tipo de produto ele deseja armazenar no estoque e/ou fazer a venda. Os dois tipos de produto funcionam de maneira igual e sendo assim a explicação de um é equivalente para o outro.

Además, cada integrante do grupo implementou uma das classes e a main, assim:

- **Debora da Silva Costa**: Implementou a classe “Cervejaria” além de fazer as sobrecargas dos operadores para as classes que precisavam.
- **Noêmia Cíntia Sales Santos da Silva**: Implementou a classe “CervejaLitro” além de implementar em todos os arquivos tudo que envolvia esta classe desde manipulação de arquivos até a passagem de parâmetros pros métodos.
- **Rafael dos Santos Lima**: Implementou a classe “CervejaUnd” além de implementar em todos os arquivos tudo que envolvia esta classe desde manipulação de arquivos até a passagem de parâmetros pros métodos.
- **Sabrina Araújo Cardoso**: Implementou a classe “Estoque” e a main principal, e também deixou o programa identado e visualmente legal na hora de compilar.

Na parte final do trabalho decidimos nos reunirmos para poder resolvermos alguns bugs na manipulação de arquivos que o nosso programa tinha e também revisar a parte visual - as saídas - dele.

1.3. Material Utilizado

- Todo o projeto foi versionado utilizando o *Git/GitHub*, possibilitando rastrear e organizar cada etapa de progresso;
- O projeto foi desenvolvido utilizando a linguagem de programação C++, como fruto da conclusão da disciplina Técnicas de Programação, na Universidade Federal de Campina Grande,do curso de Engenharia Elétrica;
- O projeto foi desenvolvido utilizando o conhecimento de orientação a objetos, através da criação de classes e classes com herança, seguido do posterior desenvolvimento dos seus métodos. Além disso, foi definido um fluxo de comunicação entre as classes;
- Utilização de manipulação de arquivos e alocação dinâmica (possibilitado com a implementação de um *Vector*);
- E a especificação do Diagrama UML e do Diagrama de Casos.

2. Classes e Métodos Utilizados

2.1. Sobrecarga.h

É responsável por realizar sobrecarga em operadores específicos, de modo que seja possível fazer toda a parte de manipulação com os arquivos nos construtores e destrutores, garantido que as listas de objetos sejam lidas logo no início da execução e sobrescritas ao final da execução. Com essas implementações, é possível realizar a manipulação de arquivos em listas de *Vector* de forma rápida e prática nas seguintes classes:

- Classe Cervejaria;
- Classe Estoque;
- Classe CervejaUnd;
- Classe CervejaLitro;

2.2. Cervejaria.cpp

A classe Cervejaria é responsável por armazenar diversos atributos de um objeto. Envolve informações sobre preço de compra, nome e código de barras do produto. Os métodos são referentes a configuração e impressão dos atributos.

2.3. CervejaUnd.cpp

A classe CervejaUnd é responsável por armazenar diversos atributos de um objeto. Sendo uma classe que tem herança com a classe Cervejaria, ela vai englobar todos os métodos envolvidos nessa classe, com o adicional de informações da quantidade de unidades do produto (seria a quantidade inserida pelo usuário para armazenar ou retirar da quantidade total), a quantidade no estoque (que seria a quantidade total existente no estoque) além das informações de validade do produto.

2.4. CervejaLitro.cpp

Análoga a CervejaUnd, a classe CervejaLitro é responsável por armazenar diversos atributos de um objeto. Sendo uma classe que tem herança com a classe Cervejaria, ela vai englobar todos os métodos envolvidos nessa classe, com o adicional de informações da quantidade em litros do produto (seria a quantidade inserida pelo usuário para armazenar ou retirar da quantidade total), a quantidade no estoque (que seria a quantidade total existente no estoque) além das informações de validade do produto.

2.5. Estoque.cpp

A classe Estoque é responsável por realizar o gerenciamento dos produtos disponíveis para venda na cervejaria. Os seus atributos são capazes de armazenar todos os objetos pertencentes a classe CervejaUnd e CervejaLitro em um *Vector.* Desse modo, foi implementado os seguintes métodos:

- Métodos que retornam um objeto armazenado na lista de *Vector* através do nome;
- Métodos que retornam um objeto armazenado na lista de *Vector* através do codigo;
- Métodos para salvar as listas de CervejaUnd e CervejaLitro em um arquivo;
- Métodos que alteram a quantidade de cada um dos produtos no estoque;
- Métodos para inserir os CervejaUnd e CervejaLitro nas listas de *Vector*;
- Métodos para imprimir os CervejaUnd e CervejaLitro nas listas de *Vector*;
- Métodos para remover os CervejaUnd e CervejaLitro nas listas de *Vector;*
- Métodos para realizar a ordenação nos dados através dos atributos associados às classes de objetos;
- Métodos que alteram a quantidade de cada um dos produtos no estoque;
- Métodos para retornar as regras de manipulação e transporte dos produtos;
- E métodos para efetuar venda dos produtos.

2.6. Main (Programa Principal)

Sendo o arquivo executável do programa, ele é responsável pela interação com o usuário uma vez que, ele permite a manipulação de produtos no estoque e também alguns métodos de retorno em que é possível imprimir o estoque ordenado bem como efetuar a venda. A opções que o usuário pode escolher são:

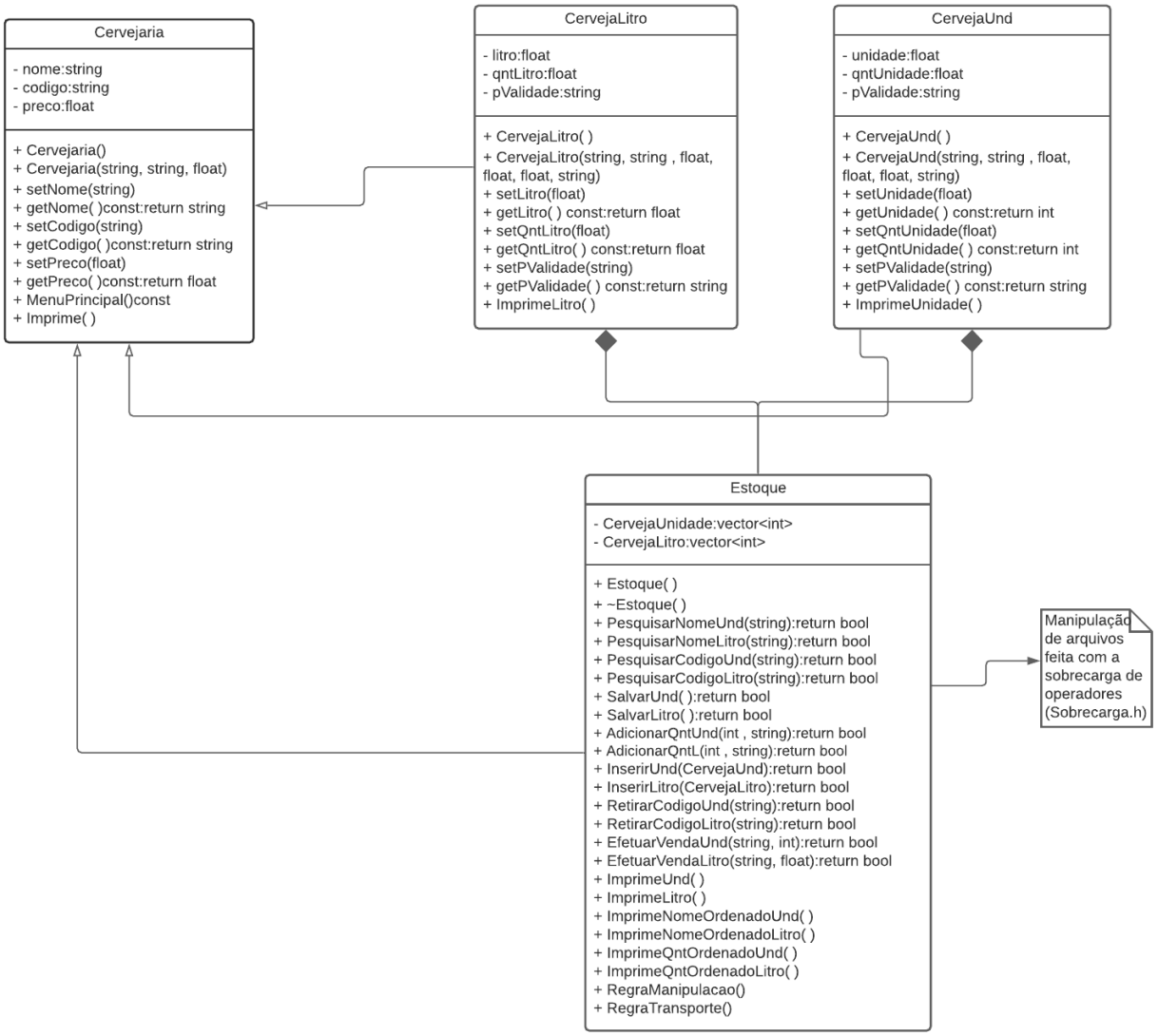
1. Para inserir uma nova cerveja no estoque.
2. Para adicionar uma determinada quantidade a um produto existente.
3. Para pesquisar uma cerveja por nome.
4. Para pesquisar cerveja por código.
5. Para efetuar venda.
6. Para remover cerveja por código.
7. Para imprimir estoque.
8. Para imprimir estoque ordenado por nome.
9. Para imprimir estoque ordenado por quantidade.
10. Para salvar o estoque em um arquivo .txt
11. Para encerrar o programa.

A cada escolha do usuário, é chamado algum método das classes Cervejaria.h, CervejaUnd.h, CervejaLitro.h e Estoque.h fazendo assim o programa ser executado.

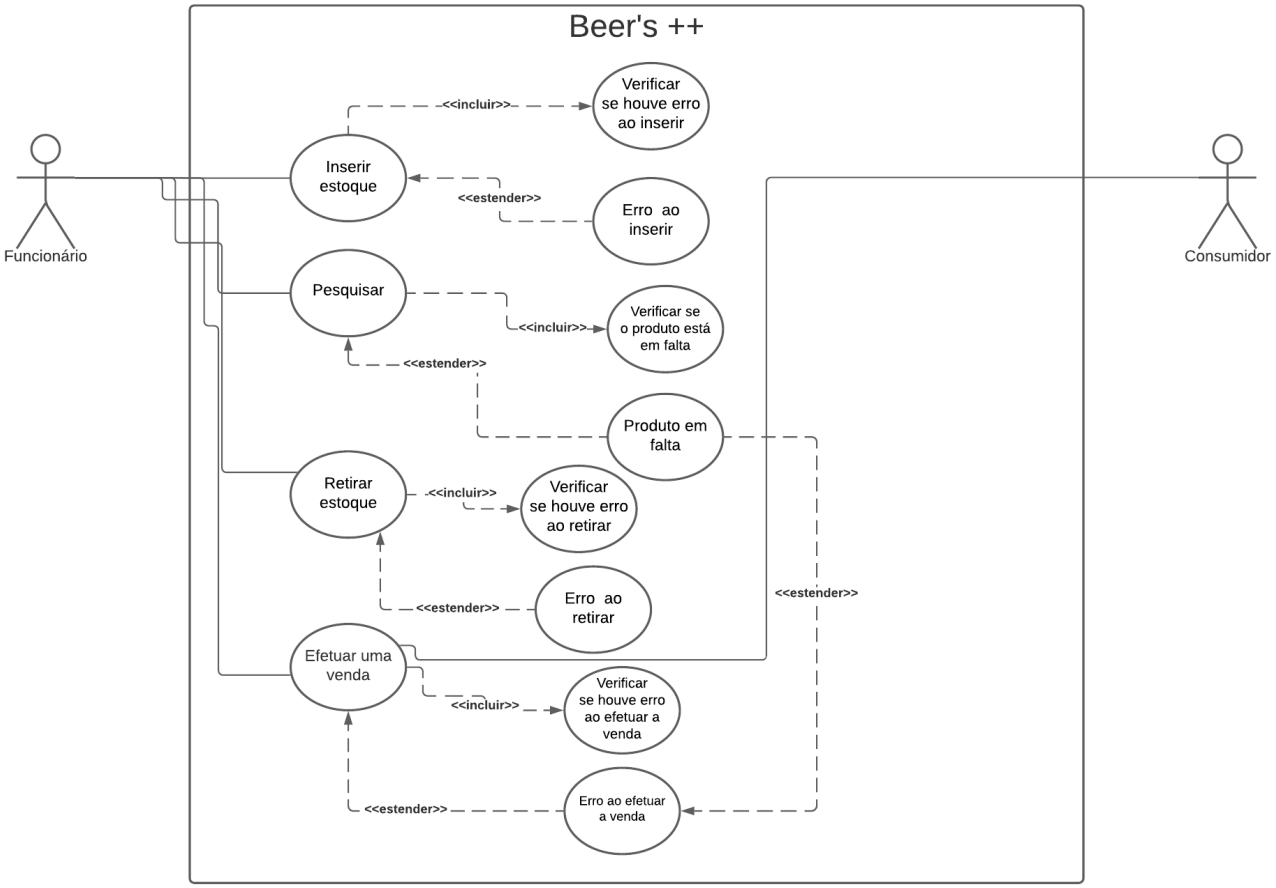
Além disso, são criados arquivos de texto para salvar as listas dos objetos CervejaUnd e CervejaLitro na máquina. Esses arquivos são EstoqueUnidade.txt e EstoqueLitro.txt.

3. Representação UML e Casos de Uso

3.1. Representação UML



3.2. Casos de Uso



4. Conclusão

Com este projeto conseguimos desenvolver muito mais nossa lógica de programação, já que utilizamos a manipulação de diversos objetos em classes distintas aumentando assim nosso repertório para também fazermos o uso de alocação dinâmica, através do Vector, e a manipulação de arquivos para otimizar nosso trabalho.

Além disso, o trabalho em grupo foi fundamental para esse desenvolvimento visto que pudemos aprender e ensinar um ao outro não só

acerca da linguagem abordada mas também o uso de outras ferramentas que otimizaram nosso trabalho, a exemplo do Git/GitHub.

Durante o processo de construção, encontramos inúmeras dificuldades como a manipulação dos arquivos com a sobrecarga, pois não estávamos conseguindo sobrecarregar um atributo de outra classe já que a sobrecarga não estava sendo feita naquela classe. Somando-se a isso tivemos uma dificuldade em perceber como iríamos calcular o estoque dos nossos produtos já que estávamos tentando fazer dentro do método de inserir. O monitor nos orientou a criar um método para calcular esse estoque e isso fez com que nosso programa tivesse melhoras significativas. Ademais, reconhecemos que existem melhorias a serem feitas em nosso programa, no entanto, não tivemos tempo para tal.

Visto tudo isso, podemos concluir que atingimos o objetivo proposto na disciplina de Técnicas de Programação.