



**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



<b>1. Introdução</b>	<b>2</b>
1.1. Objetivo	2
1.2. Resumo do Projeto	2
1.3. Material Utilizado	3
<b>2. Classes e Métodos Utilizados</b>	<b>4</b>
2.1. Sobrecarga.h	4
2.2. Cervejaria.cpp	4
2.3. CervejaUnd.cpp	4
2.4. CervejaLitro.cpp	5
2.5. Estoque.cpp	5
2.6. Main (Programa Principal)	6
<b>3. Representação UML e Casos de Uso</b>	<b>8</b>
3.1. Representação UML	8
3.2. Caso de Uso	9
<b>4. Conclusão</b>	<b>9</b>



# 1. Introdução

## 1.1. Objetivo

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

## 1.2. Resumo do Projeto

Inicialmente, optamos por trabalhar com dois tipos de estoques e vendas: por unidade e por litro. Ao iniciar, o Menu Principal do programa direciona o usuário para que este faça a escolha do estoque que deseja verificar ou inserir: 1-Unidade; 2-Litro.

A organização dos Estoques dos Produtos funcionam de maneira semelhantes; assim as explicações e direcionamentos de um são equivalentes para o outro.

Para a construção desse programa 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 essa classe desde a manipulação de arquivos até a passagem de parâmetros para os métodos.

*Rafael dos Santos Lima* implementou a classe “CervejaUnd” além de implementar em todos os arquivos tudo que envolvia essa classe desde a manipulação de arquivos até a passagem de parâmetros para os 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 nos reunimos para 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

- 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 do curso de Engenharia Elétrica da Universidade Federal de Campina Grande;
- 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;
- 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
- 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 herdeira da classe Cervejaria, ela engloba todos os métodos envolvidos nessa classe com o adicional de informações sobre a 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 engloba 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; 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 código;
- 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, a Main é responsável pela interação com o usuário uma vez que 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. As opções que o usuário pode escolher são:

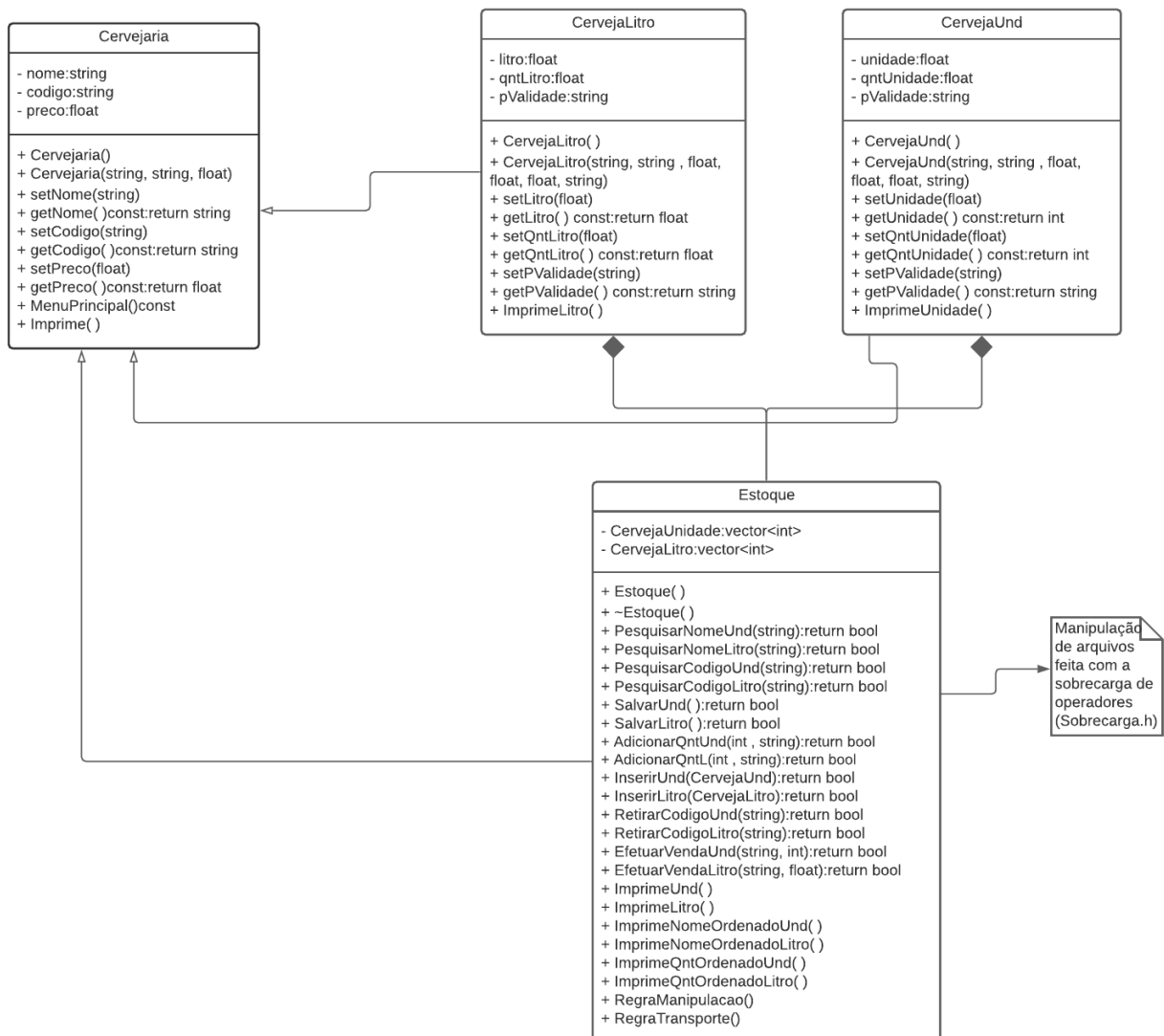
1. Para inserir uma nova cerveja no estoque.
2. Para adicionar uma determinada quantidade a um produto existente.
3. Para pesquisar uma cerveja por nome.
4. Para pesquisar cerveja por código.
5. Para efetuar venda.
6. Para remover cerveja por código.
7. Para imprimir estoque.
8. Para imprimir estoque 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

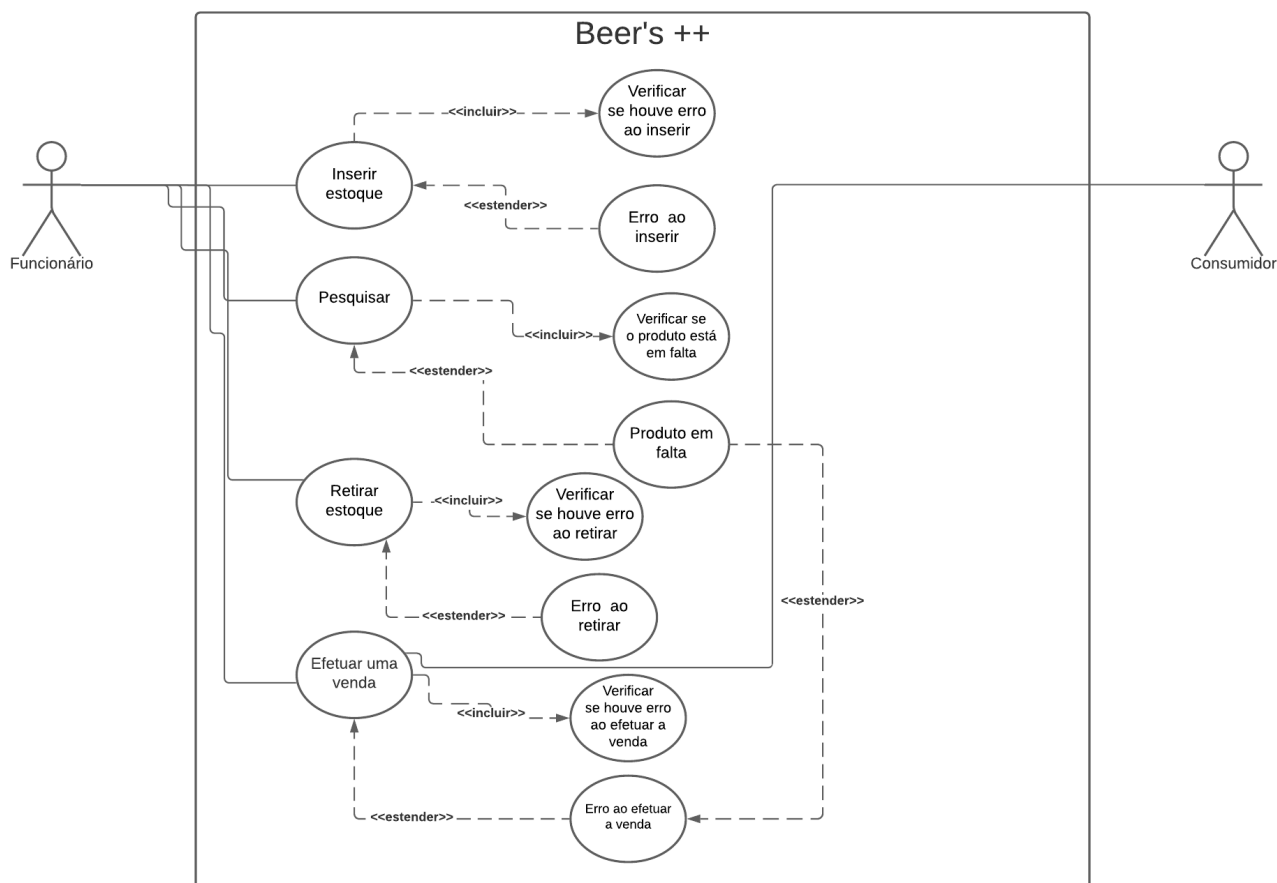
#### 3.1. Representação UML







### 3.2. Caso de Uso



## 4. Conclusão

Com este projeto conseguimos aprimorar o desenvolvimento de nossa lógica de programação, uma vez que utilizamos a manipulação de diversos objetos em classes distintas; ampliando assim nosso repertório para também fazermos o uso de alocação dinâmica, através do Vector, e a manipulação de arquivos para otimizar nosso trabalho.

O trabalho em grupo foi fundamental para esse desenvolvimento tendo em vista podermos, por meio da interação, aprender e ensinar um ao outro



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

Durante o processo de construção de nosso programa, encontramos algumas dificuldades, como por exemplo, 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; além disso, tivemos dificuldades em perceber como calcular o estoque dos nossos produtos pois estávamos tentando fazer isso dentro do método de inserir. Como parte da solução, nosso monitor nos orientou a criar um método para calcular esse estoque, com isto nosso programa avançou significativamente.

Apesar de todo o avanço e da relevância de nosso programa de controle de estoque, reconhecemos que ainda existem melhorias a serem feitas no mesmo e que isto pode, em outro momento, ser realizado por nós mesmos ou por outro grupo de graduandos.

Nessa perspectiva, conclui-se que atingimos o que nos foi proposto na disciplina de Técnicas de Programação, um projeto que teve como objetivo principal implementar um programa capaz de fazer o controle de estoque e de vendas de uma Cervejaria Artesanal usando a linguagem de programação C++.