UNIVERSIDADE FEDERAL DE OURO PRETO
*CAMPUS* MORRO DO CRUZEIRO – OURO PRETO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Eduardo Gordiano Passos Silva
Guilherme Silva
Niege Marryany dos Reis da Silva

# TRABALHO PRÁTICO I
# CORRETORA DE IMÓVEIS

OURO PRETO
2023

Eduardo Gordiano Passos Silva
Guilherme Silva
Niege Marryany dos Reis da Silva

# CORRETORA DE IMÓVEIS

OURO PRETO
2023

SUMÁRIO

# 1. INTRODUÇÃO

A implementação de uma corretora de imóveis é um projeto que exige a aplicação de conceitos avançados de programação, especialmente no contexto da Standard Template Library (STL) em C++. Neste trabalho, a ênfase será dada não apenas ao armazenamento eficiente dos dados sobre os imóveis, mas também à utilização inteligente das funcionalidades oferecidas pelos contêineres da STL.

Ao empregar os contêineres da STL de forma criteriosa, é possível otimizar o gerenciamento dos imóveis e aproveitar algoritmos pré-implementados, evitando a necessidade de implementar funcionalidades já existentes. A leitura inicial do arquivo, contendo informações cruciais sobre os imóveis, será fundamental para preencher adequadamente a coleção e permitir operações eficientes de busca e manipulação de dados.

A padronização do arquivo, com informações específicas para diferentes tipos de imóveis (apartamento, casa e chácara), adiciona uma camada de complexidade que será abordada com a utilização inteligente dos recursos disponíveis na STL. A estruturação do código para lidar com essas informações de maneira modular e eficaz será essencial para o sucesso da aplicação.

## 1.1. Objetivo Geral

O objetivo geral deste trabalho é desenvolver uma aplicação para uma corretora de imóveis, implementando um sistema de gerenciamento de imóveis baseado na Standard Templates Library (STL) em C++. O sistema deve ser capaz de ler um arquivo contendo informações sobre diferentes tipos de imóveis (casa, apartamento e chácara), armazenar esses dados em uma coleção polimórfica e oferecer funcionalidades como exibição estruturada, busca por proprietário, filtragem por valor e número de quartos, ordenação por tipo e valor, busca por cidade, localização de imóveis por proprietário, e geração de relatórios detalhados, utilizando operadores sobrecarregados e algoritmos da STL sempre que possível. O objetivo é criar um sistema eficiente e modular, explorando os recursos da linguagem e da STL para a manipulação eficaz e organizada dos dados relacionados a imóveis.

## 1.2. Objetivos Específicos

Ao implementar a aplicação, temos como objetivo.

- Implementar as classes imóvel, casa, chácara e apartamento.
- Nas classes implementadas, sobrescrevem os operadores << e implementar os métodos acessores.
- Ler um arquivo de texto contendo as informações de cada imóvel e os instanciar com base nessas linhas.
- Implementar uma função para verificar se um proprietário possui algum imóvel cadastrado.
- Implementar uma função que filtre imóveis pelo preço.
- Implementar uma função que filtre imóveis pela quantidade de quartos.
- Implementar uma função que retorne os imóveis ordenados pelo preço e suas características específicas.
- Implementar uma função que filtre os imóveis pela cidade, ordenados pelo preço.
- Implementar uma função que retorne os imóveis separados pelos tipos.
- Implementar uma função que dado a opção, salva os dados dos imóveis em um  arquivo ou imprime na tela todos os imóveis.

## 1.3. Justificativa

Este projeto visa consolidar conhecimentos em C++, explorando herança, polimorfismo e encapsulamento através da implementação de uma corretora de imóveis. Priorizando eficiência com a STL e boas práticas, o trabalho abrange manipulação de dados, como leitura de arquivos e algoritmos de filtragem e ordenação. A abordagem modular, baseada em herança e polimorfismo, proporciona flexibilidade, enquanto o encapsulamento garante organização coesa. O foco em funcionalidades comuns em corretoras culmina na implementação de uma função versátil para exibição ou salvamento de todos os tipos de imóveis, demonstrando habilidades avançadas com tipos polimórficos em C++.

## 2. DESENVOLVIMENTO

### Desenvolvimento Inicial das Classes

A etapa inicial do projeto foi dedicada à elaboração das classes que representam os diversos tipos de imóveis: casa, apartamento e chácara. Cada classe foi cuidadosamente construída com atributos específicos, refletindo as características únicas de cada categoria. A utilização do diagrama UML proporcionou uma visualização clara da hierarquia e das inter-relações entre as classes, estabelecendo uma base sólida para a implementação.

### Implementação da Leitura do Arquivo

Posteriormente, concentramos-nos na implementação da leitura do arquivo contendo as informações sobre os imóveis disponíveis. Cada linha do arquivo foi processada e convertida em instâncias das respectivas classes, garantindo uma representação fiel de cada propriedade. Esse processo foi crucial para a criação de uma coleção inicial que servirá como base para as operações subsequentes.

### Implementação das Funções de Filtro e Busca

Com a coleção devidamente preenchida, o foco então se direcionou para a implementação das funções essenciais de filtragem e busca. Desenvolvemos métodos que permitem a seleção de imóveis com base em critérios como valor, número de quartos e localização. Essas operações de filtragem foram projetadas para serem eficientes, aproveitando os recursos da Standard Template Library (STL) para otimização e clareza de código.

### Melhoria no código e refatoração

Na fase final do projeto, concentramos nossos esforços na refatoração do código, buscando aprimorar a sua legibilidade, eficiência e estrutura geral. Essa etapa foi crucial para garantir uma base sólida e sustentável, facilitando a manutenção futura e permitindo a expansão do sistema com maior facilidade.

## 2.1. Especificação de Requisitos

### 1. Introdução

#### 1.1 Objetivo do Documento

Este documento tem como objetivo especificar os requisitos para a implementação de uma corretora de imóveis, com foco na gestão de imóveis baseada na Standard Templates Library (STL).

### 2. Objetivo do Sistema

#### 2.1 Escopo do Projeto

O sistema consistirá em uma corretora de imóveis com a capacidade de gerenciar imóveis do tipo apartamento, casa e chácara. A implementação utilizará a STL para manipulação eficiente dos dados.

### 3. Definições, Acrônimos e Abreviações

STL - Standard Templates Library

### 4. Descrição Geral do Sistema

#### 4.1 Detalhes do Sistema

O sistema terá três tipos de imóveis cadastrados: casa, apartamento e chácara. Os atributos e métodos desses tipos de imóveis estão definidos conforme o diagrama UML apresentado na Figura 1.

### 5. Requisitos Funcionais

#### 5.1 Leitura do Arquivo de Imóveis

O sistema deverá ler o arquivo de imóveis disponível no link fornecido, preenchendo a coleção de imóveis no início da execução.

#### 5.2 Sobrecarga do Operador « (saída)

Todas as classes devem sobrecarregar o operador « para permitir a exibição estruturada das informações dos imóveis, separando-os por uma linha tracejada.

#### 5.3 Verificação de Proprietário

Uma função deve ser implementada para verificar se há algum imóvel associado a um determinado proprietário.

5.4 Filtragem por Valor

Uma função deve retornar uma coleção de imóveis com valor igual ou abaixo de um valor especificado.

5.5 Filtragem por Número de Quartos

Uma função deve retornar uma coleção de imóveis com número de quartos igual ou acima do especificado.

5.6 Ordenação por Tipo e Valor

Uma função deve retornar uma coleção de imóveis de um tipo específico, ordenados pelo valor.

5.7 Filtragem por Cidade e Ordenação por Valor

Uma função deve retornar uma coleção de imóveis em uma cidade específica, ordenados por valor.

5.8 Busca por Proprietário

Uma função deve retornar um iterador para cada tipo de imóvel apontando para o elemento encontrado associado a um determinado proprietário.

5.9 Impressão ou Salvamento em Arquivo

Uma função deve imprimir ou salvar em um arquivo (saida.txt) todos os tipos de imóveis de uma coleção, exibindo os dados comuns e específicos do imóvel.

6. Requisitos Não Funcionais

6.1 Eficiência na Manipulação de Dados

O sistema deve priorizar o uso eficiente dos contêineres da STL para manipulação de dados, dando preferência aos algoritmos já implementados.

7. Matriz de Rastreabilidade

7.1 Vínculos entre Requisitos e Operações

Estabelecer vínculos entre os requisitos e as operações correspondentes no código.

## 2.2. Análise dos containers do STL

Vamos analisar os containers da STL para casos de uso específicos, tais como acessar uma posição aleatória, inserir no início ou no final, acessar uma chave específica, remover ou buscar elementos.

### 2.2.1. Acessar uma posição qualquer

Ao acessar uma posição o ideal a utilização dos vector's já que eles são uma implementação de lista sequencial, e com isso conseguimos fazer aritmética de ponteiros, para acessar uma posição, não sendo necessário percorrer toda a lista até encontrar o elemento.

### 2.2.2. Adicionar um elemento e manter somente elementos únicos

Ao adicionar um elemento e manter somente elementos únicos, o ideal é o set, já que ele já insere ordenado para facilitar a busca, necessária ao inserir o novo elemento, para verificar se o elemento já existe na estrutura.

### 2.2.3. Inserir elementos no final

Ao inserir elementos no final, os vector's e ainda as deque's são ideias, já que a complexidade é constante por serem uma implementação de lista sequencial e lista sequencial com duplo início/fim respectivamente.

### 2.2.4. Remover elementos no final

Ao remover no final pelo mesmo motivo da inserção temos que os vector's e os deque's são ideais.

### 2.2.5. Retornar um valor baseado na chave (não necessariamente inteiro)

Ao retornar um valor baseado na chave é o ideal usar a tabela hash, tendo em vista que o valor não necessariamente é inteiro dificultando a busca, e já que a tabela hash é a implementação de um vetor sequencial, ou seja, acesso a qualquer elemento custo O(1) e o seu índice é um valor inteiro gerado a partir de uma função hash.

#### 2.2.6. Inserir um elemento no início

Ao inserir um elemento no início o ideal é usar os deque's já que eles possuem uma terminação dupla, e a inserção e remoção nas pontas é constante.

#### 2.2.7. Remover um elemento no início

Ao remover um elemento no início, pelo mesmo motivo de inserir um elemento no início, o ideal é usar uma deque.

#### 2.2.8. Buscar por um elemento

Ao buscar por um elemento o ideal é termos os vectores que com eles, conseguimos ordenar os elementos com O(log n) para podermos aplicar os algoritmos de busca, como a busca binária que também é O(log n), tendo assim o custo de O(log n)

#### 2.2.9. Container com comportamento FILO (first in, last out)

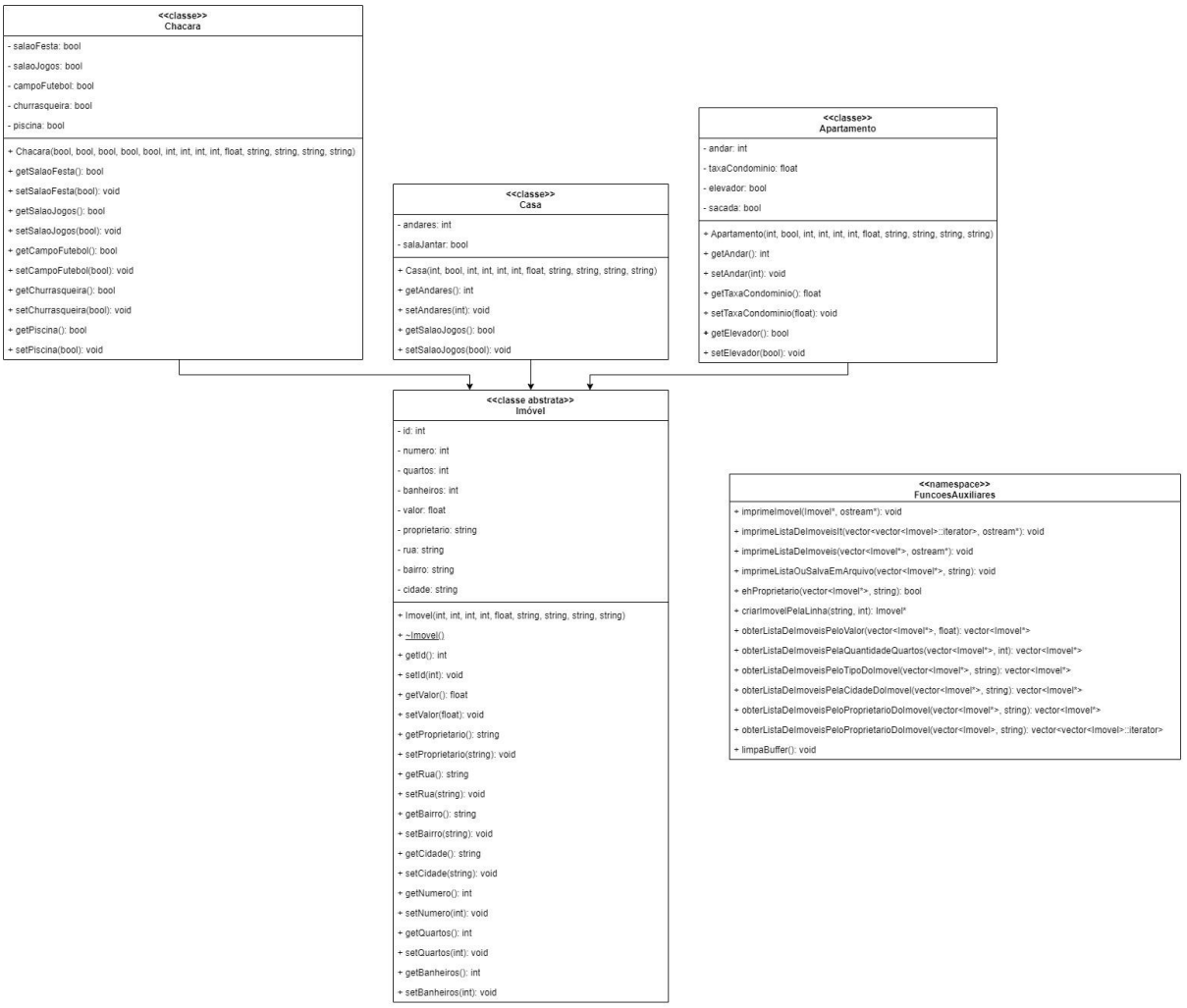
Para estas estruturas temos as implementações de Stack (pilha), a qual implementa a estrutura primeiro a entrar é o último a sair.

#### 2.2.10. Container com comportamento FIFO (first in, first out)

Para estas estruturas temos as implementações de Queue (filas), a qual implementa a estrutura a primeira a entrar e primeira a sair.

## 2.3. Diagrama de Classes

Figura 1 – Diagrama de Classes




A Figura 1 apresenta as classes Chácara, Casa, Apartamento, a classe base Imóvel e o namespace Funções Auxiliares, o qual utilizamos para implementarmos as funções auxiliares.

## 3. CONCLUSÃO

O desenvolvimento da corretora de imóveis baseada na Standard Templates Library (STL) representa um desafio empolgante e recompensador. Este trabalho teve como objetivo principal criar uma solução eficiente e modular para a gestão de imóveis, explorando as funcionalidades da STL de maneira criteriosa.

Ao longo do projeto, foi evidenciado o papel crucial dos contêineres da STL na organização e manipulação dos dados. A escolha cuidadosa dos algoritmos disponíveis na STL, em vez de recriá-los, destaca a importância de uma abordagem pragmática para garantir a eficiência do sistema.

O diagrama UML proporcionou uma representação visual clara da estrutura do sistema, facilitando o entendimento e a implementação dos diferentes tipos de imóveis. A sobrecarga do operador « (saída) e a definição de funções específicas, como busca, filtragem e ordenação, contribuíram para a construção de um sistema versátil e de fácil utilização.

A conclusão bem-sucedida da leitura do arquivo de imóveis e o preenchimento da coleção no início da execução são marcos importantes, preparando o sistema para execução de diversas operações conforme os requisitos estabelecidos.

A implementação dessas operações, como a verificação de proprietário, filtragem por valor e número de quartos, e a ordenação por tipo e valor, demonstra a capacidade do sistema de atender às demandas específicas de uma corretora de imóveis.

A busca por constantes melhorias, otimizações e aderência aos princípios da programação modular reforça o comprometimento com a qualidade e a eficiência do código. A aplicação dessas práticas não apenas torna o sistema mais robusto, mas também facilita futuras expansões e manutenções.

Em resumo, a implementação bem-sucedida da corretora de imóveis baseada na STL representa não apenas a concretização de requisitos técnicos, mas também uma prova da capacidade de enfrentar desafios complexos de programação de maneira estruturada e eficaz. Este projeto não apenas entrega uma solução funcional, mas também destaca a importância de abordagens práticas e estratégicas no desenvolvimento de software.