



## Exercícios

### Exercício 1: Registro de Alunos

**Descrição:** Crie uma classe template chamada `Registro` que armazena informações sobre alunos, como nome (tipo `std::string`) e nota (tipo `T`, que pode ser `int`, `float` ou `double`). Implemente métodos para adicionar alunos, calcular a média das notas e exibir a lista de alunos.

**Objetivo:** Testar a classe com diferentes tipos de nota e criar uma interface simples para interagir com os dados.

### Exercício 2: Fila Genérica

**Descrição:** Implemente uma classe template chamada `Fila` (Queue) que permite adicionar e remover elementos de forma FIFO (First In, First Out). A classe deve ter métodos para enfileirar (`enqueue`), desenfileirar (`dequeue`) e verificar se a fila está vazia.

**Objetivo:** Testar a fila com diferentes tipos de dados (por exemplo, `int`, `std::string`, e `float`) e simular operações de enfileiramento e desenfileiramento.

### Exercício 3: Sistema de Comparação de Itens

**Descrição:** Crie uma classe template chamada `Item` que possui um nome (`std::string`) e um valor (`T`). Implemente um método que compare dois itens com base em seus valores. Use `std::less` para determinar qual item é "menor".

**Objetivo:** Criar uma lista de itens com diferentes tipos de valor (por exemplo, `int`, `double`) e demonstrar a comparação utilizando a classe `Item`.

### Exercício 4: Mapa Genérico

**Descrição:** Implemente uma classe template chamada `Mapa` que simula um dicionário simples. A classe deve permitir armazenar pares de chave-valor, onde a chave é do tipo `K` e o valor é do tipo `V`. Implemente métodos para adicionar pares, buscar valores pela chave e listar todas as chaves armazenadas.

**Objetivo:** Testar o mapa com diferentes combinações de tipos de chave e valor (por exemplo, `std::string` para a chave e `int` para o valor).