Bacharelado em Ciência da Computação - Organização e Sistemas de Arquivos
IFNMG - Campus Montes Claros

Trabalho Prático 2: Framework Genérico para Manipulação de Arquivos

Prof. Wagner Ferreira de Barros

Objetivo

O objetivo deste trabalho é consolidar os conceitos iniciais de Representação de Registros em Arquivos em um *framework* genérico, simples e robusto em C++. Utilizando polimorfismo estático (*templates*), a classe `Arquivo` se tornará um componente reutilizável, capaz de gerenciar qualquer tipo de registro que siga um contrato predefinido, independente de seus campos internos ou formato de serialização.

Arquitetura Proposta

A arquitetura final é composta por três componentes principais que promovem baixo acoplamento e alta coesão.

- **Classe `Registro` (Abstrata):** Define a interface (contrato) que toda classe de registro (ex: `RegistroAluno`, `RegistroProduto`) deve implementar. A responsabilidade de serializar e desserializar os dados em múltiplos formatos é totalmente encapsulada aqui.
- **Classe `Buffer`:** Uma classe de baixo nível que serve como ferramenta para as classes de registro, fornecendo métodos para manipulação de bytes e campos individuais.
- **Classe `Arquivo<T>` (Template):** Um gerenciador de arquivos genérico e de alto nível. Ele é parametrizado com o tipo de registro que irá conter (ex: `Arquivo<RegistroAluno>`), garantindo segurança de tipo e flexibilidade.

Estrutura de Arquivos do Projeto

```
/projeto
|-- Registro.h
|-- RegistroAluno.h, RegistroAluno.cpp
|-- Buffer.h, Buffer.cpp
|-- Arquivo.h
|-- main.cpp
|-- dados.csv
|-- makefile
```

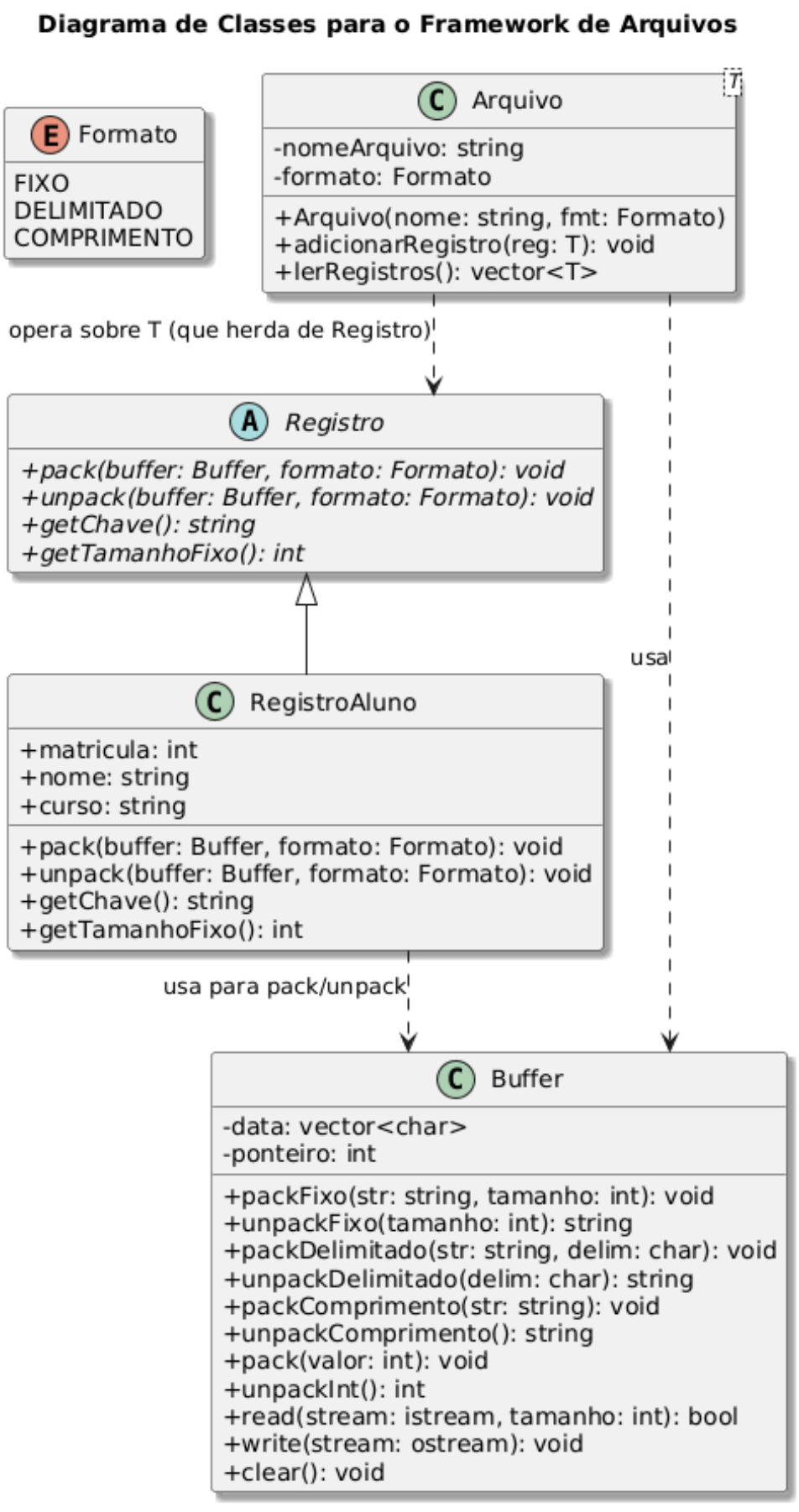

Figura 1: Diagrama de Classes UML.

Etapas do Desenvolvimento

As etapas de desenvolvimento aqui propostas são uma base para a implementação seguindo as instruções dadas em aula e neste documento. Vocês podem alterar a estrutura e a forma de implementar, mas neste projeto, o que se espera é que o programador possa reutilizá-lo para qualquer tipo de Registro de dados, apenas criando uma classe `RegistroQualquer` (que herda da classe abstrata Registro). Posteriormente, basta informar à classe Arquivo que você irá manipular aquele tipo de dado, e o formato que você deseja que ele seja representado (fixo, com delimitador ou com descritor de tamanho). A entrada dos dados continuará sendo com arquivos no formato CSV e um arquivo será disponibilizado junto à descrição da atividade. Uma proposta de desenvolvimento é dada a seguir:

1. **Definir a Classe Abstrata `Registro`:** Crie o arquivo `Registro.h` contendo um `enum` para os formatos (`FIXO, DELIMITADO, COMPRIMENTO`) e a interface da classe abstrata com os métodos virtuais puros:

```
virtual void pack(Buffer& buffer, Formato formato) const = 0;
virtual void unpack(Buffer& buffer, Formato formato) = 0;
```

2. **Implementar a Classe `Buffer`:** Desenvolva a classe `Buffer` com os métodos de empacotamento descritivos:
   - `packFixo`
   - `unpackFixo`
   - `packDelimitado`
   - `unpackDelimitado`
   - `packComprimento`
   - `unpackComprimento`.
3. **Implementar a Classe Concreta `RegistroAluno`:** Esta é a parte central do trabalho. A classe `RegistroAluno` deve herdar de `Registro` e implementar os métodos `pack` e `unpack`. Dentro desses métodos, utilize uma estrutura `switch-case` baseada no parâmetro `formato` para executar a lógica de serialização correta.
4. **Implementar a Classe Template `Arquivo<T>`:** A classe `Arquivo` deve ser um template. Ela recebe em seu construtor o formato que irá gerenciar e utiliza o tipo `T` para criar instâncias de registros ao ler do arquivo, garantindo segurança de tipo.
5. **Criar o Programa Principal (`main`):** O programa deve ler os dados de `dados.csv` e, para cada um dos três formatos, instanciar um objeto `Arquivo<RegistroAluno>` diferente, escrever os registros e depois lê-los de volta para verificação.

Entrega

- A entrega deverá ser feita em arquivo único compactado contendo todos os arquivos necessários para compilar e testar seu projeto.
- A atividade poderá ser feita em duplas.
- O prazo e a pontuação serão colocados na descrição da atividade no Google Class.