

# TRABALHO PRÁTICO PARTE 1

**Grupo:** Daniel Muller Rezende, Délio Pereira Santos Barroso Amaro, Pedro Luiz Bonorino Braga e Thomás Sousa Causin Alves

**Link do repositório no GitHub:** <https://github.com/MrMilhas/Estrutura-de-Dados-II>

## Relatório parcial

### Detalhamento das Atividades

A divisão de tarefas relacionadas a produção do trabalho foi feita da seguinte forma: como o grupo foi formado por 4 pessoas, cada duas pessoas ficaram com uma etapa do trabalho, Daniel e Delio ficaram com a etapa 1, de processamentos de dados, e Thomas e Pedro ficaram com a etapa 2, de importação de registros e módulo de teste, como foi pedido no trabalho o relatório foi feito por todos os membros do grupo. Ficou combinado de cada dupla se reunir e no final realizar os commits. Além disso, fora criado um servidor na plataforma “Discord” e um grupo no “WhatsApp” para que tornasse mais simples e eficaz a comunicação da equipe, proporcionando assim uma maior integração de conhecimento entre todas as partes da equipe.

### Decisões de Implementação

O código é dividido em algumas partes, sendo elas “headers”, “src” (essas duas sendo pastas), “.gitignore” e “makefile”. Ao entrar em “headers” é possível encontrar 2 arquivos, “review.h” e “track.h”. No review.h é onde ocorre a criação da classe Review e dentro dessa classe, existem funções importantes para o código, como por exemplo “void processaReview();”, “void verificaDir();”, “void importaRegistros();” que desempenham papéis fundamentais no código, como o de verificar a existência do arquivo binário no diretório, importar registros, verificar o diretório além de funções auxiliares para o programa funcionar em harmonia. Mas vale lembrar que não é aqui que ocorre o desenvolvimento das mesmas, apenas sua implementação. Entrando na Track.h entendemos que a mesma lógica se aplica aqui. Nessa parte são criadas strings para armazenar algumas informações a respeito do código. Ao entrar na pasta “src”, vimos 3 arquivos, “main.cpp”, “review.cpp” e “track.cpp”. Nesse momento ocorre o desenvolvimento das funções implementadas nos arquivos com terminação “.h”, sendo assim, ao entrar na review.cpp ou na track.cpp nos deparamos com aquelas mesmas funções citadas anteriormente (como exemplo), entre também muitas outras, só que agora com uma sequência lógica de código no corpo delas, que permite através disso, auxiliar a resolução da questão do trabalho, já que cada função desenvolve uma “ação” (vale lembrar que todo o código da equipe apresenta comentários ao longo dele, explicando mais a fundo o que cada função faz).

Saindo desses dois arquivos e acessando o arquivo main.cpp é o lugar onde o código se une. Na main.cpp devemos incluir algumas bibliotecas para que o código tenha melhor funcionalidade, logo abaixo dessa inclusão temos um “if”, que serve de condição de comparação para assim, desempenhar a nossa ideia desenvolvida ao longo do código. Abaixo, é criado também uma variável do tipo Review com o nome de “r” e ela aponta para a função “verificaDir();” que inicia o processo lógico desenvolvido nela e assim, completando o algoritmo.

### Referencias Utilizadas

[https://en.cppreference.com/w/cpp/language/reinterpret\\_cast](https://en.cppreference.com/w/cpp/language/reinterpret_cast)

[https://www.geeksforgeeks.org/reinterpret\\_cast-in-c-type-casting-operators/](https://www.geeksforgeeks.org/reinterpret_cast-in-c-type-casting-operators/)

<https://docs.microsoft.com/pt-br/cpp/cpp/reinterpret-cast-operator?view=msvc-170>

<https://www.inf.pucrs.br/~pinho/Laprol/Arquivos/ArquivosBinarios.htm>

<https://dcm.ffclrp.usp.br/~augusto/teaching/ici/Arquivos-Binarios-CPP.pdf>

[https://moodle.ufsc.br/pluginfile.php/2377820/mod\\_resource/content/0/exercicios%20arquivos.pdf](https://moodle.ufsc.br/pluginfile.php/2377820/mod_resource/content/0/exercicios%20arquivos.pdf)

<https://docs.microsoft.com/pt-br/cpp/c-language/sizeof-operator-c?view=msvc-170>

<https://docs.microsoft.com/pt-br/cpp/c-runtime-library/reference/strcpy-wscpy-mbscpy?view=msvc-170>

<https://en.cppreference.com/w/cpp/string/byte/strcpy>

