

Sobre a lib <ctime>

A biblioteca <ctime> em C++ fornece funcionalidades para manipulação de data e hora. Esta biblioteca é uma interface para a biblioteca de tempo em C padrão e inclui várias funções e estruturas úteis para trabalhar com datas e horas.

A estrutura `std::tm` é usada para representar datas e horas em um formato detalhado. Aqui está a definição da estrutura `std::tm`:

```
struct tm {  
    int tm_sec; // segundos após o minuto [0, 60]  
    int tm_min; // minutos após a hora [0, 59]  
    int tm_hour; // horas após a meia-noite [0, 23]  
    int tm_mday; // dia do mês [1, 31]  
    int tm_mon; // meses desde janeiro [0, 11]  
    int tm_year; // anos desde 1900  
    int tm_wday; // dias desde domingo [0, 6]  
    int tm_yday; // dias desde 1º de janeiro [0, 365]  
    int tm_isdst; // flag de horário de verão  
};
```

Para converter uma data no formato de string (como "YYYY-MM-DD" no arquivo csv) para uma estrutura `std::tm`, usamos a função `strptime` fornecida pela biblioteca <ctime>. Aqui está um exemplo de como fazer isso:

```
#include <iostream>  
#include <iomanip>  
#include <ctime>  
#include <string>  
  
std::tm stringToDate(const std::string& date) {  
    std::tm tm = {};  
    strptime(date.c_str(), "%Y-%m-%d", &tm);  
    return tm;  
}
```

Para converter uma estrutura `std::tm` de volta para uma string formatada, usamos a função `strftime` ou o manipulador `std::put_time`:

```
#include <iostream>
#include <iomanip>
#include <ctime>

void printDate(const std::tm& tm) {
    std::cout << std::put_time(&tm, "%Y-%m-%d") << std::endl;
}

int main() {
    std::tm tm = {};
    tm.tm_year = 90; // 1990
    tm.tm_mon = 0; // Janeiro
    tm.tm_mday = 1; // 1º dia

    printDate(tm);
    return 0;
}
```

Para comparar duas datas representadas por `std::tm`, convertê-las para `time_t` pode ser útil, pois `time_t` representa o tempo em segundos desde a "época" (geralmente 1º de janeiro de 1970). Aqui está um exemplo de comparação:

```
#include <iostream>
#include <ctime>

bool compareDates(const std::tm& tm1, const std::tm& tm2) {
    std::time_t time1 = std::mktime(const_cast<std::tm*>(&tm1));
    std::time_t time2 = std::mktime(const_cast<std::tm*>(&tm2));
    return std::difftime(time1, time2) < 0;
}

int main() {
    std::tm tm1 = stringToDate("1990-01-01");
    std::tm tm2 = stringToDate("1985-05-15");
}
```

```
    if (compareDates(tm1, tm2)) {  
        std::cout << "Data 1 é anterior à Data 2" << std::endl;  
    } else {  
        std::cout << "Data 1 é posterior ou igual à Data 2" << std::endl;  
    }  
  
    return 0;  
}
```

Informações adicionais podem ser encontradas nessas fontes:

<https://cplusplus.com/reference/ctime/>

https://www.tutorialspoint.com/cplusplus/cpp_date_time.htm