

Gerenciador de Tarefas

1.0

Gerado por Doxygen 1.9.1

1 Gerenciador de Tarefas	1
1.0.1 Descrição do Projeto	1
1.0.2 Tarefa.h	1
1.0.3 Tarefa.cpp	1
1.0.4 GerenciadorTarefas.cpp	2
1.0.5 Conclusão	2
1.0.6 Compilação	2
1.0.7 Menu	2
2 Índice Hierárquico	3
2.1 Hierarquia de Classes	3
3 Índice dos Componentes	5
3.1 Lista de Classes	5
4 Índice dos Arquivos	7
4.1 Lista de Arquivos	7
5 Classes	9
5.1 Referência da Classe Tarefa	9
5.1.1 Descrição detalhada	10
5.1.2 Construtores e Destrutores	10
5.1.2.1 Tarefa()	10
5.1.3 Funções membros	10
5.1.3.1 atualizarTotalTarefas()	11
5.1.3.2 estaConcluida()	11
5.1.3.3 getDescricao()	11
5.1.3.4 getId()	11
5.1.3.5 getTotalTarefas()	12
5.1.4 Amigas e Funções Relacionadas	12
5.1.4.1 operator<<	12
5.1.5 Atributos	12
5.1.5.1 concluida	12
5.1.5.2 descricao	12
5.1.5.3 id	13
5.1.5.4 totalTarefas	13
5.2 Referência da Classe TarefaAmanha	13
5.2.1 Descrição detalhada	14
5.2.2 Construtores e Destrutores	14
5.2.2.1 TarefaAmanha()	14
5.2.3 Funções membros	14
5.2.3.1 atualizarTotalTarefasAmanha()	15
5.2.3.2 getTotalTarefasAmanha()	15
5.2.4 Atributos	15

5.2.4.1 totalTarefasAmanha	15
6 Arquivos	17
6.1 Referência do Arquivo src/GerenciadorTarefas.cpp	17
6.1.1 Descrição detalhada	18
6.1.2 Funções	18
6.1.2.1 adicionarTarefa()	18
6.1.2.2 adicionarTarefaAmanha()	18
6.1.2.3 concluirTarefa()	19
6.1.2.4 exibirTarefas()	19
6.1.2.5 listarTarefasAmanha()	19
6.1.2.6 main()	20
6.1.2.7 salvarTarefas()	20
6.2 Referência do Arquivo src/Tarefa.cpp	20
6.2.1 Descrição detalhada	21
6.2.2 Funções	21
6.2.2.1 operator<<()	21
6.3 Referência do Arquivo src/Tarefa.h	21
6.3.1 Descrição detalhada	22
Índice Remissivo	23

Capítulo 1

Gerenciador de Tarefas

1.0.1 Descrição do Projeto

O projeto consiste em um gerenciador de tarefas implementado em C++, dividido em três arquivos principais: `Tarefa.h`, `Tarefa.cpp` e `GerenciadorTarefas.cpp`. O código é estruturado de forma a demonstrar conceitos importantes de programação, como alocação de memória dinâmica, manipulação de arquivos, construtores, destrutores, uso de memória estática, sobrecarga de função e herança.

1.0.2 Tarefa.h

Este arquivo contém a definição da classe `Tarefa` e a classe derivada `TarefaAmanha`. A classe `Tarefa` possui membros privados, como o identificador (`id`), descrição (`descricao`) e estado de conclusão (`concluida`). Há também um membro estático (`totalTarefas`) que mantém o total de tarefas criadas.

A classe `Tarefa` possui métodos para obter o ID, a descrição, verificar se a tarefa está concluída, concluir a tarefa, exibir informações e métodos estáticos para obter o total de tarefas e atualizar esse total. A sobrecarga do operador de saída (`<<`) é utilizada para facilitar a exibição das informações da tarefa.

A classe derivada `TarefaAmanha` herda da classe `Tarefa` e adiciona seu próprio membro estático (`totalTarefasAmanha`). Além disso, ela sobreescreve o método `exibir` para incluir informações específicas sobre tarefas programadas para o dia seguinte.

1.0.3 Tarefa.cpp

Neste arquivo, são implementados os métodos declarados na classe `Tarefa` e `TarefaAmanha`. A alocação dinâmica de memória é utilizada para criar novas instâncias de tarefas. O destrutor é responsável por liberar a memória alocada para cada tarefa.

Os métodos de manipulação de tarefas, como adicionar, concluir e exibir, são implementados para facilitar a interação com o gerenciador. O arquivo também contém a implementação de funções estáticas para atualizar o total de tarefas.

1.0.4 GerenciadorTarefas.cpp

O arquivo principal do programa implementa as funções de interação com o usuário, utilizando um menu simples. As funções incluem adição de tarefas, listagem, conclusão de tarefas, adição de tarefas para o dia seguinte, listagem de tarefas para o dia seguinte, salvamento das tarefas em um arquivo e desalocação de memória.

É importante observar que a desalocação de memória é realizada no final do programa para evitar vazamento de memória. Além disso, as funções de salvar tarefas em um arquivo e listar tarefas para o dia seguinte demonstram o uso de manipulação de arquivos.

1.0.5 Conclusão

O projeto fornece uma implementação prática de diversos conceitos de programação em C++, demonstrando a aplicação de boas práticas, herança, manipulação de memória, interação com o usuário e persistência de dados. O código está estruturado de forma modular, facilitando a compreensão e manutenção do sistema.

1.0.6 Compilação

Um método mais prático seria fazendo uso do CMake e na pasta raiz do projeto executar os comandos abaixo:

```
mkdir build
cd build
cmake ..
```

Uma vez compilado:

```
make
./GerenciadorTarefas
```

1.0.7 Menu

Capítulo 2

Índice Hierárquico

2.1 Hierarquia de Classes

Esta lista de hierarquias está parcialmente ordenada (ordem alfabética):

Tarefa	9
TarefaAmanha	13

Capítulo 3

Índice dos Componentes

3.1 Lista de Classes

Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:

Tarefa	Classe que representa uma tarefa	9
TarefaAmanha	Classe que representa uma tarefa agendada para amanhã	13

Capítulo 4

Índice dos Arquivos

4.1 Lista de Arquivos

Esta é a lista de todos os arquivos documentados e suas respectivas descrições:

src/ GerenciadorTarefas.cpp	
Implementação das funções do Gerenciador de Tarefas	17
src/ Tarefa.cpp	
Implementação das funções relacionadas às classes Tarefa e TarefaAmanha	20
src/ Tarefa.h	
Declaração das classes Tarefa e TarefaAmanha	21

Capítulo 5

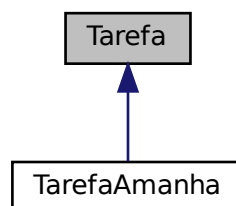
Classes

5.1 Referência da Classe Tarefa

Classe que representa uma tarefa.

```
#include <Tarefa.h>
```

Diagrama de hierarquia para Tarefa:



Membros Públicos

- **Tarefa** (const std::string &descricao)
Construtor da classe Tarefa.
- virtual ~Tarefa ()
Destrutor da classe Tarefa.
- int getId () const
Obtém o ID da tarefa.
- std::string getDescricao () const
Obtém a descrição da tarefa.
- bool estaConcluida () const
Verifica se a tarefa está concluída.
- void concluir ()
Marca a tarefa como concluída.
- virtual void exibir () const
Exibe informações sobre a tarefa.

Membros Públicos Estáticos

- static int `getTotalTarefas` ()
Obtém o total de tarefas criadas.
- static void `atualizarTotalTarefas` (int novoTotal)
Atualiza o total de tarefas criadas.

Atributos Privados

- int `id`
- std::string `descricao`
- bool `concluida`

Atributos Privados Estáticos

- static int `totalTarefas` = 0

Amigas

- std::ostream & `operator<<` (std::ostream &os, const `Tarefa` &tarefa)
Sobrecarga do operador de saída para facilitar a exibição de uma tarefa.

5.1.1 Descrição detalhada

Classe que representa uma tarefa.

5.1.2 Construtores e Destrutores

5.1.2.1 `Tarefa()`

```
Tarefa::Tarefa (  
    const std::string & descricao )
```

Construtor da classe `Tarefa`.

Parâmetros

<code>descricao</code>	Descrição da tarefa.
------------------------	----------------------

5.1.3 Funções membros

5.1.3.1 atualizarTotalTarefas()

```
void Tarefa::atualizarTotalTarefas (
    int novoTotal ) [static]
```

Atualiza o total de tarefas criadas.

Parâmetros

<i>novoTotal</i>	Novo total de tarefas.
------------------	------------------------

5.1.3.2 estaConcluida()

```
bool Tarefa::estaConcluida ( ) const
```

Verifica se a tarefa está concluída.

Retorna

true se a tarefa estiver concluída, false caso contrário.

5.1.3.3 getDescricao()

```
std::string Tarefa::getDescricao ( ) const
```

Obtém a descrição da tarefa.

Retorna

Descrição da tarefa.

5.1.3.4 getId()

```
int Tarefa::getId ( ) const
```

Obtém o ID da tarefa.

Retorna

ID da tarefa.

5.1.3.5 getTotalTarefas()

```
int Tarefa::getTotalTarefas ( ) [static]
```

Obtém o total de tarefas criadas.

Retorna

Total de tarefas.

5.1.4 Amigas e Funções Relacionadas

5.1.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Tarefa & tarefa ) [friend]
```

Sobrecarga do operador de saída para facilitar a exibição de uma tarefa.

Parâmetros

<i>os</i>	Stream de saída.
<i>tarefa</i>	Tarefa a ser exibida.

Retorna

Stream de saída atualizada.

5.1.5 Atributos

5.1.5.1 concluida

```
bool Tarefa::concluida [private]
```

Indica se a tarefa está concluída.

5.1.5.2 descricao

```
std::string Tarefa::descricao [private]
```

Descrição da tarefa.

5.1.5.3 id

```
int Tarefa::id [private]
```

Identificador único da tarefa.

5.1.5.4 totalTarefas

```
int Tarefa::totalTarefas = 0 [static], [private]
```

Total de tarefas criadas.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- [src/Tarefa.h](#)
- [src/Tarefa.cpp](#)

5.2 Referência da Classe TarefaAmanha

Classe que representa uma tarefa agendada para amanhã.

```
#include <Tarefa.h>
```

Diagrama de hierarquia para TarefaAmanha:

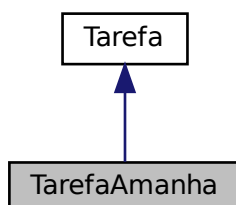
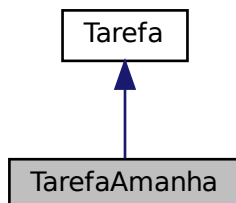


Diagrama de colaboração para TarefaAmanha:



Membros Públicos

- `TarefaAmanha` (const std::string &descricao)
Construtor da classe `TarefaAmanha`.
- virtual `~TarefaAmanha` ()
Destrutor da classe `TarefaAmanha`.
- virtual void `exibir` () const override
Sobrescreve a função `exibir` para exibir informações específicas de tarefas agendadas para amanhã.

Membros Públicos Estáticos

- static int `getTotalTarefasAmanha` ()
Obtém o total de tarefas agendadas para amanhã.
- static void `atualizarTotalTarefasAmanha` (int novoTotal)
Atualiza o total de tarefas agendadas para amanhã.

Atributos Privados Estáticos

- static int `totalTarefasAmanha` = 0

5.2.1 Descrição detalhada

Classe que representa uma tarefa agendada para amanhã.

5.2.2 Construtores e Destrutores

5.2.2.1 `TarefaAmanha()`

```
TarefaAmanha::TarefaAmanha (
    const std::string & descricao )
```

Construtor da classe `TarefaAmanha`.

Parâmetros

<code>descricao</code>	Descrição da tarefa agendada para amanhã.
------------------------	---

5.2.3 Funções membros

5.2.3.1 atualizarTotalTarefasAmanha()

```
void TarefaAmanha::atualizarTotalTarefasAmanha (
    int novoTotal ) [static]
```

Atualiza o total de tarefas agendadas para amanhã.

Parâmetros

<i>novoTotal</i>	Novo total de tarefas agendadas para amanhã.
------------------	--

5.2.3.2 getTotalTarefasAmanha()

```
int TarefaAmanha::getTotalTarefasAmanha ( ) [static]
```

Obtém o total de tarefas agendadas para amanhã.

Retorna

Total de tarefas agendadas para amanhã.

5.2.4 Atributos

5.2.4.1 totalTarefasAmanha

```
int TarefaAmanha::totalTarefasAmanha = 0 [static], [private]
```

Total de tarefas agendadas para amanhã.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- [src/Tarefa.h](#)
- [src/Tarefa.cpp](#)

Capítulo 6

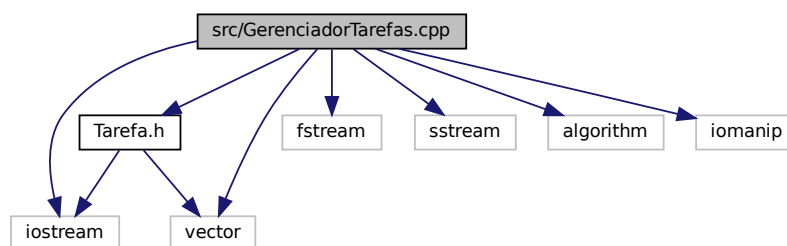
Arquivos

6.1 Referência do Arquivo src/GerenciadorTarefas.cpp

Implementação das funções do Gerenciador de Tarefas.

```
#include "Tarefa.h"
#include <vector>
#include <fstream>
#include <sstream>
#include <iostream>
#include <algorithm>
#include <iomanip>
```

Gráfico de dependência de inclusões para GerenciadorTarefas.cpp:



Funções

- void `exibirMenu` ()
Exibe o menu de opções do programa.
- void `exibirTarefas` (const std::vector< `Tarefa` * > &tarefas)
Exibe a lista de tarefas.
- void `adicionarTarefa` (std::vector< `Tarefa` * > &tarefas)
Adiciona uma nova tarefa à lista.
- void `concluirTarefa` (std::vector< `Tarefa` * > &tarefas)
Conclui e remove uma tarefa da lista.

- void `adicionarTarefaAmanha` (`std::vector< Tarefa * > &tarefas`)
Adiciona uma nova tarefa agendada para amanhã à lista.
- void `listarTarefasAmanha` (`const std::vector< Tarefa * > &tarefas`)
Lista as tarefas agendadas para amanhã.
- void `salvarTarefas` (`const std::vector< Tarefa * > &tarefas`)
Salva as tarefas no arquivo "tarefas.txt".
- int `main` ()
Função principal do programa.

6.1.1 Descrição detalhada

Implementação das funções do Gerenciador de Tarefas.

Este arquivo contém a implementação das funções responsáveis por gerenciar tarefas, como adicionar, exibir, concluir e agendar tarefas para amanhã.

Autor

Francisco Willem Romão Moreira

Data

2023-12-18

6.1.2 Funções

6.1.2.1 adicionarTarefa()

```
void adicionarTarefa (  
    std::vector< Tarefa * > & tarefas )
```

Adiciona uma nova tarefa à lista.

Parâmetros

<code>tarefas</code>	Vetor de ponteiros para as tarefas.
----------------------	-------------------------------------

6.1.2.2 adicionarTarefaAmanha()

```
void adicionarTarefaAmanha (  
    std::vector< Tarefa * > & tarefas )
```

Adiciona uma nova tarefa agendada para amanhã à lista.

Parâmetros

<i>tarefas</i>	Vetor de ponteiros para as tarefas.
----------------	-------------------------------------

6.1.2.3 concluirTarefa()

```
void concluirTarefa (
    std::vector< Tarefa * > & tarefas )
```

Conclui e remove uma tarefa da lista.

Parâmetros

<i>tarefas</i>	Vetor de ponteiros para as tarefas.
----------------	-------------------------------------

6.1.2.4 exibirTarefas()

```
void exibirTarefas (
    const std::vector< Tarefa * > & tarefas )
```

Exibe a lista de tarefas.

Parâmetros

<i>tarefas</i>	Vetor de ponteiros para as tarefas.
----------------	-------------------------------------

6.1.2.5 listarTarefasAmanha()

```
void listarTarefasAmanha (
    const std::vector< Tarefa * > & tarefas )
```

Lista as tarefas agendadas para amanhã.

Parâmetros

<i>tarefas</i>	Vetor de ponteiros para as tarefas.
----------------	-------------------------------------

6.1.2.6 main()

```
int main ( )
```

Função principal do programa.

Retorna

0 ao finalizar o programa.

6.1.2.7 salvarTarefas()

```
void salvarTarefas (
    const std::vector< Tarefa * > & tarefas )
```

Salva as tarefas no arquivo "tarefas.txt".

Parâmetros

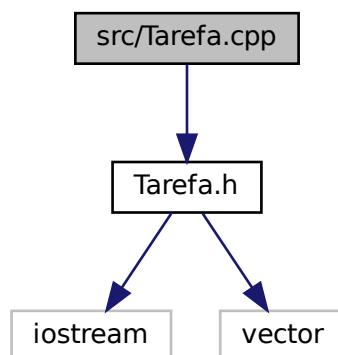
<i>tarefas</i>	Vetor de ponteiros para as tarefas.
----------------	-------------------------------------

6.2 Referência do Arquivo src/Tarefa.cpp

Implementação das funções relacionadas às classes [Tarefa](#) e [TarefaAmanha](#).

```
#include "Tarefa.h"
```

Gráfico de dependência de inclusões para Tarefa.cpp:



Funções

- `std::ostream & operator<< (std::ostream &os, const Tarefa &tarefa)`

Sobrecarga do operador de saída para facilitar a exibição de uma tarefa.

6.2.1 Descrição detalhada

Implementação das funções relacionadas às classes [Tarefa](#) e [TarefaAmanha](#).

Este arquivo contém a implementação das funções membros das classes [Tarefa](#) e [TarefaAmanha](#), incluindo construtores, destrutores e outras funcionalidades relacionadas.

Autor

Francisco Willem Romão Moreira

Data

2023-12-18

6.2.2 Funções

6.2.2.1 operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const Tarefa & tarefa )
```

Sobrecarga do operador de saída para facilitar a exibição de uma tarefa.

Parâmetros

<i>os</i>	Stream de saída.
<i>tarefa</i>	Tarefa a ser exibida.

Retorna

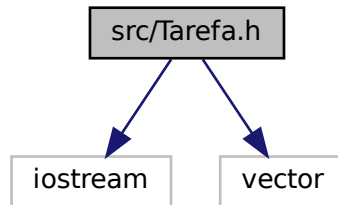
Stream de saída atualizada.

6.3 Referência do Arquivo src/Tarefa.h

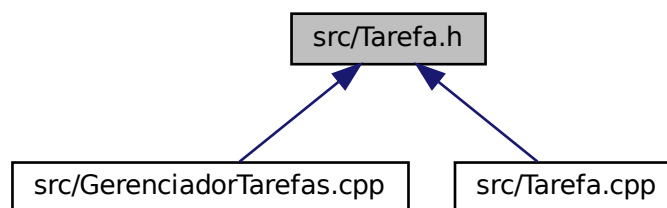
Declaração das classes [Tarefa](#) e [TarefaAmanha](#).

```
#include <iostream>
#include <vector>
```

Gráfico de dependência de inclusões para Tarefa.h:



Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com esse arquivo:



Componentes

- class [Tarefa](#)
Classe que representa uma tarefa.
- class [TarefaAmanha](#)
Classe que representa uma tarefa agendada para amanhã.

6.3.1 Descrição detalhada

Declaração das classes [Tarefa](#) e [TarefaAmanha](#).

Autor

Francisco Willem Romão Moreira

Data

2023-12-18

Índice Remissivo

- adicionarTarefa
 - GerenciadorTarefas.cpp, 18
- adicionarTarefaAmanha
 - GerenciadorTarefas.cpp, 18
- atualizarTotalTarefas
 - Tarefa, 10
- atualizarTotalTarefasAmanha
 - TarefaAmanha, 14
- concluida
 - Tarefa, 12
- concluirTarefa
 - GerenciadorTarefas.cpp, 19
- descricao
 - Tarefa, 12
- estaConcluida
 - Tarefa, 11
- exibirTarefas
 - GerenciadorTarefas.cpp, 19
- GerenciadorTarefas.cpp
 - adicionarTarefa, 18
 - adicionarTarefaAmanha, 18
 - concluirTarefa, 19
 - exibirTarefas, 19
 - listarTarefasAmanha, 19
 - main, 19
 - salvarTarefas, 20
- getDescricao
 - Tarefa, 11
- getId
 - Tarefa, 11
- getTotalTarefas
 - Tarefa, 11
- getTotalTarefasAmanha
 - TarefaAmanha, 15
- id
 - Tarefa, 12
- listarTarefasAmanha
 - GerenciadorTarefas.cpp, 19
- main
 - GerenciadorTarefas.cpp, 19
- operator<<
 - Tarefa, 12
 - Tarefa.cpp, 21
- salvarTarefas
 - GerenciadorTarefas.cpp, 20
 - src/GerenciadorTarefas.cpp, 17
 - src/Tarefa.cpp, 20
 - src/Tarefa.h, 21
- Tarefa, 9
 - atualizarTotalTarefas, 10
 - concluida, 12
 - descricao, 12
 - estaConcluida, 11
 - getDescricao, 11
 - getId, 11
 - getTotalTarefas, 11
 - id, 12
 - operator<<, 12
 - Tarefa, 10
 - totalTarefas, 13
- Tarefa.cpp
 - operator<<, 21
- TarefaAmanha, 13
 - atualizarTotalTarefasAmanha, 14
 - getTotalTarefasAmanha, 15
 - TarefaAmanha, 14
 - totalTarefasAmanha, 15
- totalTarefas
 - Tarefa, 13
- totalTarefasAmanha
 - TarefaAmanha, 15