

Trabalho de Grafos

1

Gerado por Doxygen 1.8.10

Sumário

1	Índice das Estruturas de Dados	1
1.1	Estruturas de Dados	1
2	Índice dos Arquivos	3
2.1	Lista de Arquivos	3
3	Estruturas	5
3.1	Referência da Estrutura Celula_priv	5
3.1.1	Descrição Detalhada	5
3.1.2	Campos	5
3.1.2.1	duracao	5
3.1.2.2	executada	5
3.1.2.3	id_externo	5
3.1.2.4	ini_min	6
3.1.2.5	nome	6
3.1.2.6	pre_req	6
3.1.2.7	reqs	6
3.2	Referência da Estrutura grafo_priv	6
3.2.1	Descrição Detalhada	7
3.2.2	Campos	7
3.2.2.1	origem	7
3.2.2.2	tabela	7
3.2.2.3	vert	7
3.3	Referência da Estrutura lista_aresta	7
3.3.1	Descrição Detalhada	8
3.3.2	Campos	8
3.3.2.1	destino	8
3.3.2.2	next	8
3.3.2.3	peso	8
3.4	Referência da Estrutura lista_origem	9
3.4.1	Descrição Detalhada	9
3.4.2	Campos	9

3.4.2.1	destino	9
3.4.2.2	next	9
3.5	Referência da Estrutura lista_vert	9
3.5.1	Descrição Detalhada	10
3.5.2	Campos	10
3.5.2.1	antecessores	10
3.5.2.2	id	10
3.5.2.3	id_externo	10
3.5.2.4	next	11
3.5.2.5	sucessores	11
3.6	Referência da Estrutura lista_vert_codigo	11
3.6.1	Descrição Detalhada	11
3.6.2	Campos	11
3.6.2.1	dado	11
3.6.2.2	id	12
3.6.2.3	next	12
3.7	Referência da Estrutura Modulo	12
3.7.1	Descrição Detalhada	12
4	Arquivos	13
4.1	Referência do Arquivo /home/rafael/ProjetoFinalMP/Fase2Interface/include/grafos.h	13
4.1.1	Descrição Detalhada	14
4.1.2	Definições dos tipos	14
4.1.2.1	Celula_priv_t	14
4.1.2.2	grafo_priv_t	14
4.1.2.3	resposta	14
4.1.3	Enumerações	14
4.1.3.1	resp	14
4.1.4	Funções	15
4.1.4.1	achar_celula(const grafo_priv_t *meu_grafo, int id_externo)	15
4.1.4.2	achar_id(const grafo_priv_t *meu_grafo, int id_externo)	15
4.1.4.3	cria_celula(int id_externo, int executada, int duracao, int ini_min, int pre_req, int *reqs, const char *nome)	16
4.1.4.4	criaGrafoArg(char *nomeArg)	16
4.1.4.5	criar_grafo(void)	16
4.1.5	Descrição	16
4.1.6	Parâmetros	17
4.1.7	Assertiva de saída	17
4.1.7.1	deletar_grafo(grafo_priv_t *meu_grafo)	17
4.1.8	Descrição	17

4.1.9	Assertiva de saída	18
4.1.9.1	editar_celula(grafo_priv_t *meu_grafo, int ID)	18
4.1.9.2	eh_conexo(const grafo_priv_t *meu_grafo)	19
4.1.9.3	existe_aresta(const grafo_priv_t *meu_grafo, int id_externo1, int id_externo2)	19
4.1.9.4	existe_vert(const grafo_priv_t *meu_grafo, int id_externo)	20
4.1.10	Descrição	20
4.1.11	Assertiva de entrada	20
4.1.11.1	Grava_Arq(grafo_priv_t *meu_grafo, char *NomeArq)	21
4.1.11.2	Imprime_Tarefas(const grafo_priv_t *meu_grafo, int linha, int coluna)	21
4.1.11.3	inserir_aresta(grafo_priv_t *meu_grafo, int id_externo1, Celula_priv_t *celula2, int peso)	22
4.1.11.4	inserir_vert(grafo_priv_t *meu_grafo, Celula_priv_t *celula)	22
4.1.11.5	jafeito(const grafo_priv_t *meu_grafo, int d)	23
4.1.11.6	Ler_Tarefas(grafo_priv_t *meu_grafo, Celula_priv_t *celula, const char *NomeArq)	24
4.1.11.7	maior_id(const grafo_priv_t *meu_grafo)	24
4.1.11.8	menor_caminho(const grafo_priv_t *meu_grafo, int **dist)	24
4.1.11.9	num_arestas(const grafo_priv_t *meu_grafo)	25
4.1.11.10	num_vert(const grafo_priv_t *meu_grafo)	25
4.1.11.11	remover_aresta(grafo_priv_t *meu_grafo, int id_externo1, int id_externo2)	26
4.1.11.12	remover_vert(grafo_priv_t *meu_grafo, int id_externo)	26
4.1.11.13	tempo_minimo(const grafo_priv_t *meu_grafo, int id_fim)	27
4.2	Referência do Arquivo /home/rafael/ProjetoFinalMP/Fase2Interface/include/grafa_priv.h	28
4.2.1	Definições dos tipos	29
4.2.1.1	Celula_priv_t	29
4.2.1.2	grafa_priv_t	29
4.2.1.3	lista_aresta_t	29
4.2.1.4	lista_origem_t	29
4.2.1.5	lista_vert_codigo_t	30
4.2.1.6	lista_vert_t	30
4.2.2	Funções	30
4.2.2.1	existe_origem(const grafo_priv_t *meu_grafo, int id_externo)	30
4.2.2.2	inserir_origem(grafo_priv_t *meu_grafo, Celula_priv_t *celula)	30
4.2.2.3	remover_origem(grafo_priv_t *meu_grafo, int id_externo)	31
4.3	Referência do Arquivo grafo.cpp	32
4.3.1	Definições e macros	33
4.3.1.1	DEBUG	33
4.3.2	Funções	33
4.3.2.1	achar_celula(const grafo_priv_t *meu_grafo, int id_externo)	33
4.3.2.2	achar_id(const grafo_priv_t *meu_grafo, int id_externo)	34

4.3.2.3	<code>cria_celula(int id_externo, int executada, int duracao, int ini_min, int pre_req, int *reqs, const char *nome)</code>	34
4.3.2.4	<code>criaGrafoArq(char *nomeArq)</code>	34
4.3.2.5	<code>criar_grafo(void)</code>	35
4.3.3	Descrição	35
4.3.4	Parâmetros	35
4.3.5	Assertiva de saída	35
4.3.5.1	<code>deletar_grafo(grafo_priv_t *meu_grafo)</code>	36
4.3.6	Descrição	36
4.3.7	Assertiva de saída	36
4.3.7.1	<code>dfs(const grafo_priv_t *meu_grafo, lista_vert_t *atual, int *marc)</code>	37
4.3.7.2	<code>editar_celula(grafo_priv_t *meu_grafo, int ID)</code>	37
4.3.7.3	<code>eh_conexo(const grafo_priv_t *meu_grafo)</code>	38
4.3.7.4	<code>existe_aresta(const grafo_priv_t *meu_grafo, int id_externo1, int id_externo2)</code>	38
4.3.7.5	<code>existe_origem(const grafo_priv_t *meu_grafo, int id_externo)</code>	39
4.3.7.6	<code>existe_vert(const grafo_priv_t *meu_grafo, int id_externo)</code>	39
4.3.8	Descrição	39
4.3.9	Assertiva de entrada	40
4.3.9.1	<code>Grava_Arq(grafo_priv_t *meu_grafo, char *NomeArq)</code>	40
4.3.9.2	<code>Imprime_Tarefas(const grafo_priv_t *meu_grafo, int linha, int coluna)</code>	41
4.3.9.3	<code>inserir_aresta(grafo_priv_t *meu_grafo, int id_externo1, Celula_priv_t *celula2, int peso)</code>	41
4.3.9.4	<code>inserir_origem(grafo_priv_t *meu_grafo, Celula_priv_t *celula)</code>	42
4.3.9.5	<code>inserir_vert(grafo_priv_t *meu_grafo, Celula_priv_t *celula)</code>	43
4.3.9.6	<code>ja_feito(const grafo_priv_t *meu_grafo, int d)</code>	43
4.3.9.7	<code>maior_id(const grafo_priv_t *meu_grafo)</code>	44
4.3.9.8	<code>menor_caminho(const grafo_priv_t *meu_grafo, int **dist)</code>	45
4.3.9.9	<code>num_arestas(const grafo_priv_t *meu_grafo)</code>	45
4.3.9.10	<code>num_vert(const grafo_priv_t *meu_grafo)</code>	45
4.3.9.11	<code>remover_aresta(grafo_priv_t *meu_grafo, int id_externo1, int id_externo2)</code>	46
4.3.9.12	<code>remover_origem(grafo_priv_t *meu_grafo, int id_externo)</code>	46
4.3.9.13	<code>remover_vert(grafo_priv_t *meu_grafo, int id_externo)</code>	47
4.3.9.14	<code>tempo_minimo(const grafo_priv_t *meu_grafo, int id_fim)</code>	48
4.4	Referência do Arquivo <code>interface.cpp</code>	49
4.4.1	Definições e macros	49
4.4.1.1	<code>NCURSES_CONST</code>	49
4.4.2	Funções	49
4.4.2.1	<code>interface_caminho_completo(const grafo_priv_t *meu_grafo)</code>	49
4.4.2.2	<code>interface_caminho_parcial(grafo_priv_t *meu_grafo)</code>	50
4.4.2.3	<code>interface_editar_tarefa(grafo_priv_t *meu_grafo)</code>	51

4.4.2.4	interface_inserir_tarefa(grafo_priv_t *meu_grafo)	51
4.4.2.5	interface_remover_tarefa(grafo_priv_t *meu_grafo)	52
4.4.2.6	interface_vizualizar_determinada_tarefa(const grafo_priv_t *meu_grafo)	53
4.4.2.7	interface_vizualizar_tarefas(const grafo_priv_t *meu_grafo)	53
4.4.2.8	main()	54
Índice		55

Capítulo 1

Índice das Estruturas de Dados

1.1 Estruturas de Dados

Aqui estão as estruturas de dados, uniões e suas respectivas descrições:

Celula_priv	5
grafo_priv	6
lista_aresta	7
lista_origem	9
lista_vert	9
lista_vert_codigo	11
Modulo	12

Capítulo 2

Índice dos Arquivos

2.1 Lista de Arquivos

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

/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafo.h	
Define funções usadas pelo usuario	13
/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafo_priv.h	28
grafo.cpp	32
interface.cpp	49

Capítulo 3

Estruturas

3.1 Referência da Estrutura Celula_priv

```
#include <grafo_priv.h>
```

Campos de Dados

- int `id_externo`
- int `executada`
- int `duracao`
- int `ini_min`
- int `pre_req`
- int * `reqs`
- char `nome` [200]

3.1.1 Descrição Detalhada

Definição na linha 6 do arquivo `grafo_priv.h`.

3.1.2 Campos

3.1.2.1 int Celula_priv::duracao

Definição na linha 7 do arquivo `grafo_priv.h`.

Referenciado por `cria_celula()`, `criaGrafoArq()`, `editar_celula()`, `Grava_Arq()`, `Imprime_Tarefas()`, `interface_inserir_tarefa()` e `interface_vizualizar_determinada_tarefa()`.

3.1.2.2 int Celula_priv::executada

Definição na linha 7 do arquivo `grafo_priv.h`.

Referenciado por `cria_celula()`, `criaGrafoArq()`, `criar_grafo()`, `editar_celula()`, `Grava_Arq()`, `Imprime_Tarefas()`, `interface_inserir_tarefa()` e `interface_vizualizar_determinada_tarefa()`.

3.1.2.3 int Celula_priv::id_externo

Definição na linha 7 do arquivo `grafo_priv.h`.

Referenciado por `achar_celula()`, `achar_id()`, `cria_celula()`, `criaGrafoArq()`, `criar_grafo()`, `deletar_grafo()`, `editar_celula()`, `existe_vert()`, `Grava_Arq()`, `Imprime_Tarefas()`, `inserir_aresta()`, `inserir_origem()`, `inserir_vert()`, `interface_caminho_completo()`, `interface_inserir_tarefa()` e `interface_vizualizar_determinada_tarefa()`.

3.1.2.4 `int Celula_priv::ini_min`

Definição na linha 8 do arquivo `grafo_priv.h`.

Referenciado por `cria_celula()`, `criaGrafoArq()`, `criar_grafo()`, `editar_celula()`, `Grava_Arq()`, `Imprime_Tarefas()`, `inserir_vert()`, `interface_inserir_tarefa()`, `interface_vizualizar_determinada_tarefa()` e `menor_caminho()`.

3.1.2.5 `char Celula_priv::nome[200]`

Definição na linha 10 do arquivo `grafo_priv.h`.

Referenciado por `cria_celula()`, `criaGrafoArq()`, `criar_grafo()`, `editar_celula()`, `Grava_Arq()`, `Imprime_Tarefas()`, `inserir_origem()`, `inserir_vert()`, `interface_inserir_tarefa()` e `interface_vizualizar_determinada_tarefa()`.

3.1.2.6 `int Celula_priv::pre_req`

Definição na linha 8 do arquivo `grafo_priv.h`.

Referenciado por `cria_celula()`, `criaGrafoArq()`, `criar_grafo()`, `editar_celula()`, `Grava_Arq()`, `Imprime_Tarefas()`, `inserir_vert()`, `interface_inserir_tarefa()` e `interface_vizualizar_determinada_tarefa()`.

3.1.2.7 `int* Celula_priv::reqs`

Definição na linha 9 do arquivo `grafo_priv.h`.

Referenciado por `cria_celula()`, `criaGrafoArq()`, `editar_celula()`, `Grava_Arq()`, `Imprime_Tarefas()`, `interface_inserir_tarefa()` e `interface_vizualizar_determinada_tarefa()`.

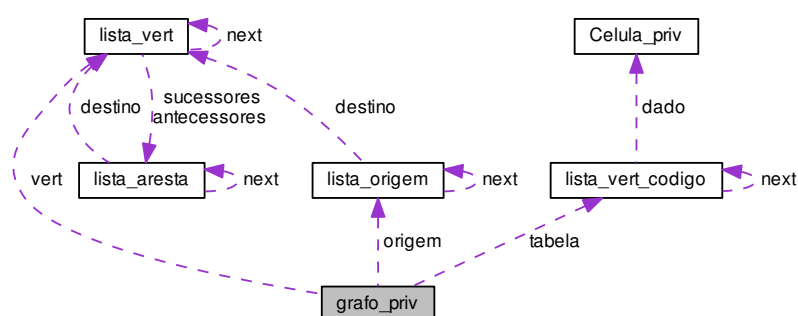
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `/home/rafael/ProjetoFinalIMP/Fase2Interface/include/grafo_priv.h`

3.2 Referência da Estrutura `grafo_priv`

```
#include <grafo_priv.h>
```

Diagrama de colaboração para `grafo_priv`:



Campos de Dados

- [lista_vert_codigo_t * tabela](#)
- [lista_vert_t * vert](#)
- [lista_origem_t * origem](#)

3.2.1 Descrição Detalhada

Definição na linha 41 do arquivo grafo_priv.h.

3.2.2 Campos

3.2.2.1 lista_origem_t* grafo_priv::origem

Definição na linha 44 do arquivo grafo_priv.h.

Referenciado por `criar_grafo()`, `eh_conexo()`, `existe_origem()`, `inserir_origem()` e `remover_origem()`.

3.2.2.2 lista_vert_codigo_t* grafo_priv::tabela

Definição na linha 42 do arquivo grafo_priv.h.

Referenciado por `achar_celula()`, `achar_id()`, `criar_grafo()`, `deletar_grafo()`, `existe_vert()`, `Grava_Arq()`, `Imprime_↔` `Tarefas()`, `inserir_vert()`, `interface_caminho_completo()`, `ja_feito()`, `maior_id()` e `remover_vert()`.

3.2.2.3 lista_vert_t* grafo_priv::vert

Definição na linha 43 do arquivo grafo_priv.h.

Referenciado por `criar_grafo()`, `existe_aresta()`, `inserir_aresta()`, `inserir_origem()`, `inserir_vert()`, `menor_caminho()`, `num_arestas()`, `num_vert()`, `remover_aresta()` e `remover_vert()`.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafos_priv.h](#)

3.3 Referência da Estrutura lista_aresta

```
#include <grafo_priv.h>
```

Diagrama de colaboração para lista_aresta:



Campos de Dados

- struct `lista_vert` * `destino`
- double `peso`
- struct `lista_aresta` * `next`

3.3.1 Descrição Detalhada

Definição na linha 22 do arquivo `grafo_priv.h`.

3.3.2 Campos

3.3.2.1 struct `lista_vert`* `lista_aresta::destino`

Definição na linha 23 do arquivo `grafo_priv.h`.

Referenciado por `dfs()`, `existe_aresta()`, `inserir_aresta()`, `menor_caminho()` e `remover_vert()`.

3.3.2.2 struct `lista_aresta`* `lista_aresta::next`

Definição na linha 25 do arquivo `grafo_priv.h`.

Referenciado por `dfs()`, `existe_aresta()`, `inserir_aresta()`, `menor_caminho()`, `num_arestas()` e `remover_aresta()`.

3.3.2.3 double `lista_aresta::peso`

Definição na linha 24 do arquivo `grafo_priv.h`.

Referenciado por `inserir_aresta()` e `menor_caminho()`.

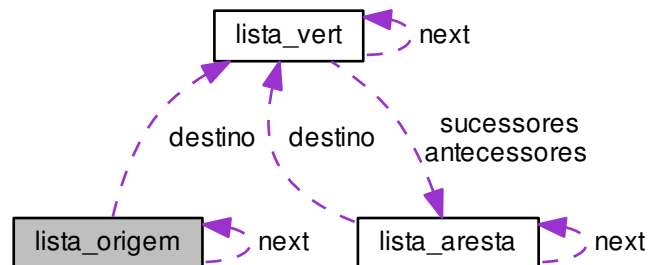
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafo_priv.h`

3.4 Referência da Estrutura lista_origem

```
#include <grafo_priv.h>
```

Diagrama de colaboração para lista_origem:



Campos de Dados

- `lista_vert_t * destino`
- `struct lista_origem * next`

3.4.1 Descrição Detalhada

Definição na linha 36 do arquivo grafo_priv.h.

3.4.2 Campos

3.4.2.1 `lista_vert_t * lista_origem::destino`

Definição na linha 37 do arquivo grafo_priv.h.

Referenciado por `eh_conexo()`, `existe_origem()`, `inserir_origem()` e `remover_origem()`.

3.4.2.2 `struct lista_origem * lista_origem::next`

Definição na linha 38 do arquivo grafo_priv.h.

Referenciado por `eh_conexo()`, `existe_origem()`, `inserir_origem()` e `remover_origem()`.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafos_priv.h`

3.5 Referência da Estrutura lista_vert

```
#include <grafo_priv.h>
```

Diagrama de colaboração para lista_vert:



Campos de Dados

- int `id`
- int `id_externo`
- `lista_aresta_t` * `antecessores`
- `lista_aresta_t` * `sucessores`
- struct `lista_vert` * `next`

3.5.1 Descrição Detalhada

Definição na linha 28 do arquivo `grafo_priv.h`.

3.5.2 Campos

3.5.2.1 `lista_aresta_t`* `lista_vert::antecessores`

Definição na linha 31 do arquivo `grafo_priv.h`.

Referenciado por `inserir_vert()` e `remover_aresta()`.

3.5.2.2 `int lista_vert::id`

Definição na linha 29 do arquivo `grafo_priv.h`.

Referenciado por `dfs()`, `eh_conexo()`, `inserir_origem()`, `inserir_vert()`, `menor_caminho()`, `remover_aresta()`, `remover_origem()` e `remover_vert()`.

3.5.2.3 `int lista_vert::id_externo`

Definição na linha 30 do arquivo `grafo_priv.h`.

Referenciado por `existe_aresta()`, `existe_origem()`, `inserir_vert()` e `remover_vert()`.

3.5.2.4 struct lista_vert* lista_vert::next

Definição na linha 33 do arquivo grafo_priv.h.

Referenciado por existe_aresta(), inserir_aresta(), inserir_origem(), inserir_vert(), menor_caminho(), num_arestas(), num_vert(), remover_aresta() e remover_vert().

3.5.2.5 lista_aresta_t* lista_vert::sucessores

Definição na linha 32 do arquivo grafo_priv.h.

Referenciado por dfs(), existe_aresta(), inserir_aresta(), inserir_vert(), menor_caminho(), num_arestas() e remover_aresta().

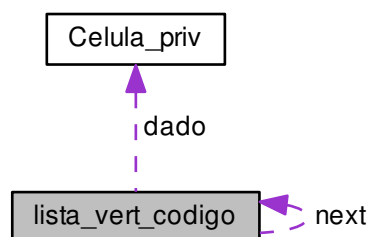
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafo_priv.h](#)

3.6 Referência da Estrutura lista_vert_codigo

```
#include <grafo_priv.h>
```

Diagrama de colaboração para lista_vert_codigo:



Campos de Dados

- [Celula_priv_t](#) dado
- int id
- struct [lista_vert_codigo](#) * next

3.6.1 Descrição Detalhada

Definição na linha 16 do arquivo grafo_priv.h.

3.6.2 Campos

3.6.2.1 Celula_priv_t lista_vert_codigo::dado

Definição na linha 17 do arquivo grafo_priv.h.

Referenciado por `achar_celula()`, `achar_id()`, `deletar_grafo()`, `existe_vert()`, `Grava_Arq()`, `Imprime_Tarefas()` e `interface_caminho_completo()`.

3.6.2.2 `int lista_vert_codigo::id`

Definição na linha 18 do arquivo `grafo_priv.h`.

Referenciado por `achar_id()`, `inserir_vert()`, `ja_feito()`, `maior_id()` e `remover_vert()`.

3.6.2.3 `struct lista_vert_codigo* lista_vert_codigo::next`

Definição na linha 19 do arquivo `grafo_priv.h`.

Referenciado por `achar_celula()`, `achar_id()`, `existe_vert()`, `Grava_Arq()`, `Imprime_Tarefas()`, `inserir_vert()`, `interface_caminho_completo()`, `ja_feito()`, `maior_id()` e `remover_vert()`.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafo_priv.h](#)

3.7 Referência da Estrutura Modulo

3.7.1 Descrição Detalhada

Este modulo contem a interface grafica que sera mostrada para o usuario do Gerenciador de Tarefas.

Cada funcao foi feita para apresentar uma determinada janela e executar uma diferente operacao sobre o gerenciador.

As funcoes possuem uma padronizacao em sua denominacao, sendo esse padrao dado por "interface_o_que_a↵_funcao_faz"

Todas as funcoes recebem como parametro um ponteiro para o grafo que contem o gerenciador de tarefas.

Estas funcoes podem ser acessadas por meio de um menu inicial apresentado na tela quando se inicia o programa.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [interface.cpp](#)

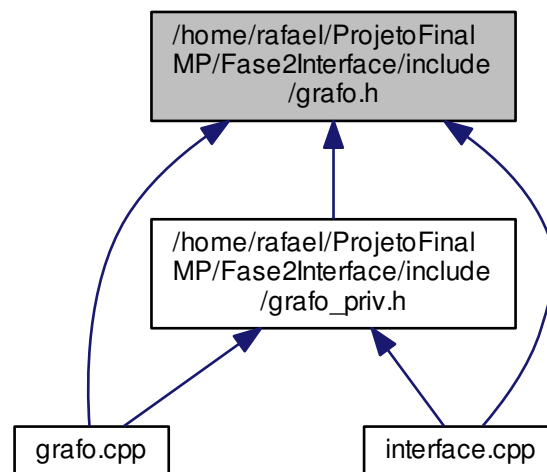
Capítulo 4

Arquivos

4.1 Referência do Arquivo /home/rafael/ProjetoFinalMP/Fase2Interface/include/graf.h

Define funções usadas pelo usuario.

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



Definições de Tipos

- typedef enum `resp` `resposta`
- typedef struct `grafo_priv` `grafo_priv_t`
- typedef struct `Celula_priv` `Celula_priv_t`

Enumerações

- enum `resp` { `FALSE_T`, `TRUE_T` }

Funções

- `grafo_priv_t * criar_grafo` (void)
Cria grafo.
- `grafo_priv_t * deletar_grafo` (grafo_priv_t *meu_grafo)
Deleta grafo.
- `resposta existe_vert` (const grafo_priv_t *meu_grafo, int id_externo)
Existe vértice nome?
- `Celula_priv_t * cria_celula` (int id_externo, int executada, int duracao, int ini_min, int pre_req, int *reqs, const char *nome)
- `resposta existe_aresta` (const grafo_priv_t *meu_grafo, int id_externo1, int id_externo2)
- `int achar_id` (const grafo_priv_t *meu_grafo, int id_externo)
- `Celula_priv_t * achar_celula` (const grafo_priv_t *meu_grafo, int id_externo)
- `void inserir_vert` (grafo_priv_t *meu_grafo, Celula_priv_t *celula)
- `void inserir_aresta` (grafo_priv_t *meu_grafo, int id_externo1, Celula_priv_t *celula2, int peso)
- `void remover_vert` (grafo_priv_t *meu_grafo, int id_externo)
- `void remover_aresta` (grafo_priv_t *meu_grafo, int id_externo1, int id_externo2)
- `int maior_id` (const grafo_priv_t *meu_grafo)
- `int num_vert` (const grafo_priv_t *meu_grafo)
- `int num_arestas` (const grafo_priv_t *meu_grafo)
- `int menor_caminho` (const grafo_priv_t *meu_grafo, int **dist)
- `resposta eh_conexo` (const grafo_priv_t *meu_grafo)
- `int tempo_minimo` (const grafo_priv_t *meu_grafo, int id_fim)
- `void ja_feito` (const grafo_priv_t *meu_grafo, int d)
- `void Ler_Tarefas` (grafo_priv_t *meu_grafo, Celula_priv_t *celula, const char *NomeArq)
- `void editar_celula` (grafo_priv_t *meu_grafo, int ID)
- `void Imprime_Tarefas` (const grafo_priv_t *meu_grafo, int linha, int coluna)
- `void Grava_Arq` (grafo_priv_t *meu_grafo, char *NomeArq)
- `grafo_priv_t * criaGrafoArq` (char *nomeArq)

4.1.1 Descrição Detalhada

Define funções usadas pelo usuário.

4.1.2 Definições dos tipos

4.1.2.1 typedef struct Celula_priv Celula_priv_t

Definição na linha 19 do arquivo grafo.h.

4.1.2.2 typedef struct grafo_priv grafo_priv_t

Definição na linha 17 do arquivo grafo.h.

4.1.2.3 typedef enum resp resposta

4.1.3 Enumerações

4.1.3.1 enum resp

Valores de enumerações

FALSE_T

TRUE_T

Definição na linha 9 do arquivo grafos.h.

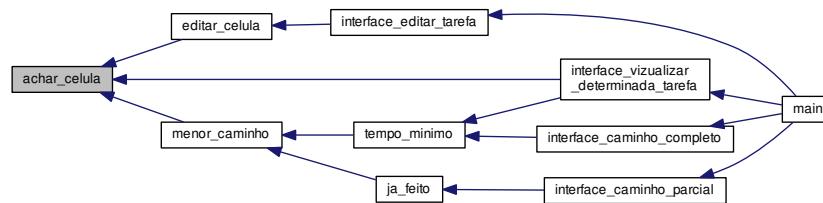
4.1.4 Funções**4.1.4.1 Celula_priv_t* achar_celula (const grafos_priv_t * meu_grafos, int id_externo)**

Definição na linha 117 do arquivo grafos.cpp.

Referências lista_vert_codigo::dado, Celula_priv::id_externo, lista_vert_codigo::next e grafos_priv::tabela.

Referenciado por editar_celula(), interface_vizualizar_determinada_tarefa() e menor_caminho().

Este é o diagrama das funções que utilizam esta função:

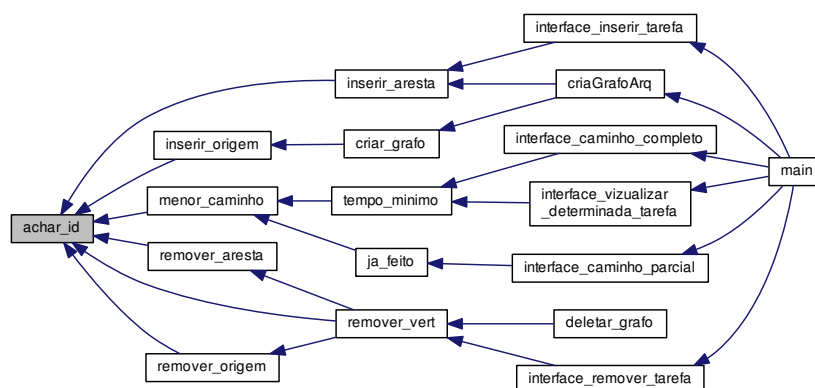
**4.1.4.2 int achar_id (const grafos_priv_t * meu_grafos, int id_externo)**

Definição na linha 107 do arquivo grafos.cpp.

Referências lista_vert_codigo::dado, lista_vert_codigo::id, Celula_priv::id_externo, lista_vert_codigo::next e grafos_priv::tabela.

Referenciado por inserir_aresta(), inserir_origem(), menor_caminho(), remover_aresta(), remover_origem() e remover_vert().

Este é o diagrama das funções que utilizam esta função:



4.1.4.3 `Celula_priv_t* cria_celula (int id_externo, int executada, int duracao, int ini_min, int pre_req, int * reqs, const char * nome)`

Definição na linha 43 do arquivo grafo.cpp.

Referências `Celula_priv::duracao`, `Celula_priv::executada`, `Celula_priv::id_externo`, `Celula_priv::ini_min`, `Celula_priv::nome`, `Celula_priv::pre_req` e `Celula_priv::reqs`.

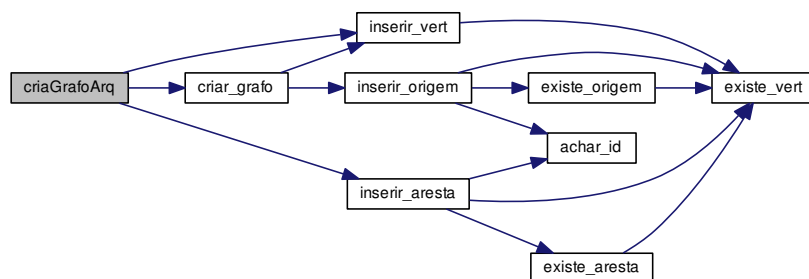
4.1.4.4 `grafo_priv_t* criaGrafoArq (char * nomeArq)`

Definição na linha 767 do arquivo grafo.cpp.

Referências `criar_grafo()`, `Celula_priv::duracao`, `Celula_priv::executada`, `Celula_priv::id_externo`, `Celula_priv::ini_min`, `inserir_aresta()`, `inserir_vert()`, `Celula_priv::nome`, `Celula_priv::pre_req` e `Celula_priv::reqs`.

Referenciado por `main()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.1.4.5 `grafo_priv_t* criar_grafo (void)`

Cria grafo.

4.1.5 Descrição

Aloca a memória necessária e inicializa um grafo

4.1.6 Parâmetros

Não há parâmetros, a alocação e inicialização não dependem de nenhum parâmetro do usuário

Retorna

Se retorna um ponteiro para o grafo criado

4.1.7 Assertiva de saída

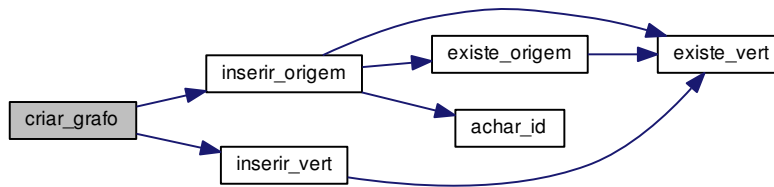
O grafo gerado é consistente e não possui nenhum vértice, origem ou aresta.

Definição na linha 12 do arquivo grafos.cpp.

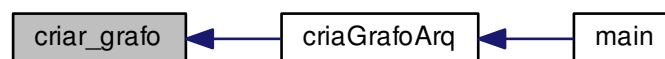
Referências `Celula_priv::executada`, `Celula_priv::id_externo`, `Celula_priv::ini_min`, `inserir_origem()`, `inserir_vert()`, `Celula_priv::nome`, `grafos_priv::origem`, `Celula_priv::pre_req`, `grafos_priv::tabela` e `grafos_priv::vert`.

Referenciado por `criaGrafoArq()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.1.7.1 `grafos_priv_t* deletar_grafo (grafos_priv_t * meu_grafo)`

Deleta grafo.

4.1.8 Descrição

Desaloca toda a memória utilizada pelo grafo

Parâmetros

<i>meu_grafo</i>	- Deve ser passado um ponteiro para um grafo inicializado
------------------	---

Retorna

Retorna um ponteiro para o grafo, que será NULL. O valor de retorno é muito importante, uma vez que se ele não for utilizado o grafo do usuário apontará para um endereço não alocado e qualquer tentativa de utilizá-lo poderá gerar erros no sistema. Caso o grafo passado não tenha sido inicializado, o programa poderá parar a execução

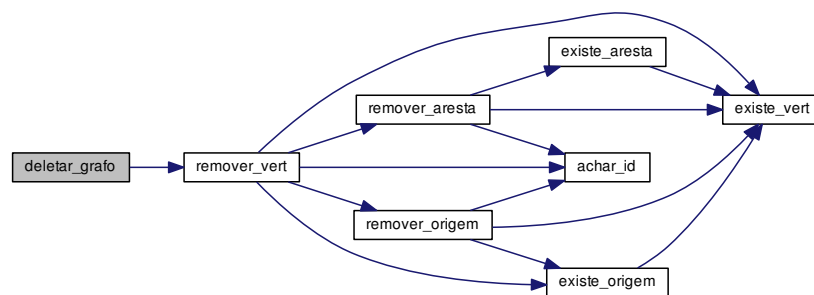
4.1.9 Assertiva de saída

O grafo já deve ter sido inicializado por [criar_grafo\(\)](#) Assertiva de saída Se retornará um ponteiro para o mesmo grafo passado, após a deleção que será NULL.

Definição na linha 32 do arquivo grafo.cpp.

Referências `lista_vert_codigo::dado`, `Celula_priv::id_externo`, `remover_vert()` e `grafo_priv::tabela`.

Este é o diagrama das funções utilizadas por esta função:

**4.1.9.1 void editar_celula (grafo_priv_t * meu_grafo, int ID)**

Definição na linha 664 do arquivo grafo.cpp.

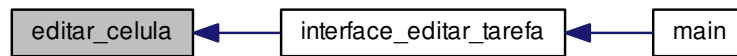
Referências `achar_celula()`, `Celula_priv::duracao`, `Celula_priv::executada`, `Celula_priv::id_externo`, `Celula_priv::ini_min`, `Celula_priv::nome`, `Celula_priv::pre_req` e `Celula_priv::reqs`.

Referenciado por `interface_editar_tarefa()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:

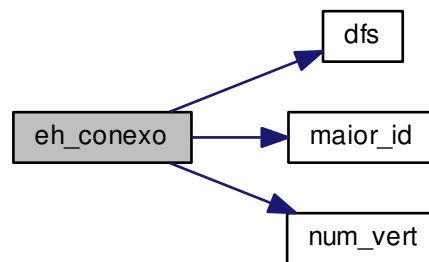


4.1.9.2 resposta eh_conexo (const grafo_priv_t * meu_grafo)

Definição na linha 581 do arquivo grafo.cpp.

Referências lista_origem::destino, dfs(), FALSE_T, lista_vert::id, maior_id(), lista_origem::next, num_vert(), grafo->_priv::origem e TRUE_T.

Este é o diagrama das funções utilizadas por esta função:



4.1.9.3 resposta existe_aresta (const grafo_priv_t * meu_grafo, int id_externo1, int id_externo2)

Definição na linha 87 do arquivo grafo.cpp.

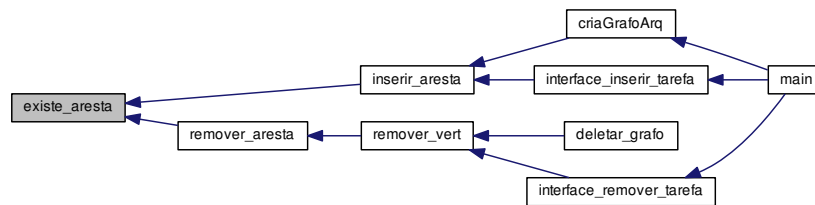
Referências lista_aresta::destino, existe_vert(), FALSE_T, lista_vert::id_externo, lista_aresta::next, lista_vert::next, lista_vert::sucessores, TRUE_T e grafo_priv::vert.

Referenciado por inserir_aresta() e remover_aresta().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.1.9.4 resposta existe_vert (const grafo_priv_t * meu_grafo, int id_externo)

Existe vértice nome?

4.1.10 Descrição

Verfica a existência de um vértice dado

Parâmetros

<i>meu_grafo</i>	- Deve ser passado um ponteiro para um grafo inicializado
<i>nome</i>	- Deve ser passado uma string de até 100 caracteres, mais o caracter zero terminal. Se a string tiver mais que 100 caracteres, o vértice não será encontrado, uma vez que 100 é o maior número de caracteres que um vértice pode ter.

Retorna

Retorna uma resposta, que será TRUE caso exista o vértice, e FALSE caso não exista.

4.1.11 Assertiva de entrada

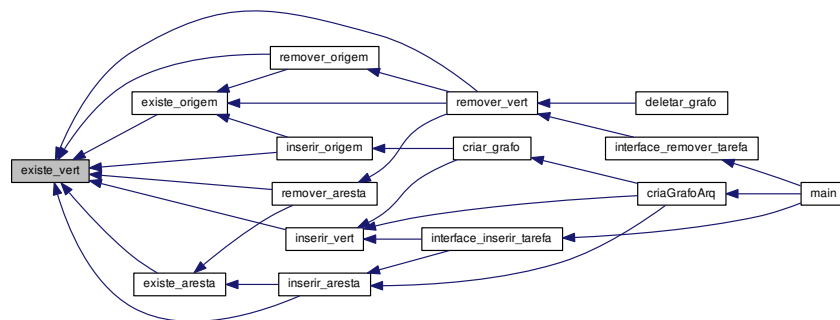
O grafo já deve ter sido inicializado por [criar_grafo\(\)](#), se não for o programa pode ser interrompido. `strlen(nome) <= 100` Assertivas de saída O retorno será TRUE ou FALSE O conteúdo do grafo e da string passados não serão modificados

Definição na linha 64 do arquivo grafo.cpp.

Referências `lista_vert_codigo::dado`, `FALSE_T`, `Celula_priv::id_externo`, `lista_vert_codigo::next`, `grafo_priv::tabela` e `TRUE_T`.

Referenciado por `existe_aresta()`, `existe_origem()`, `inserir_aresta()`, `inserir_origem()`, `inserir_vert()`, `remover_aresta()`, `remover_origem()` e `remover_vert()`.

Este é o diagrama das funções que utilizam esta função:



4.1.11.1 void Grava_Arq (grafo_priv_t * meu_grafo, char * NomeArq)

Definição na linha 637 do arquivo grafos.cpp.

Referências lista_vert_codigo::dado, Celula_priv::duracao, Celula_priv::executada, Celula_priv::id_externo, Celula_priv::ini_min, lista_vert_codigo::next, Celula_priv::nome, Celula_priv::pre_req, Celula_priv::reqs e grafo_priv::tabela.

Referenciado por main().

Este é o diagrama das funções que utilizam esta função:



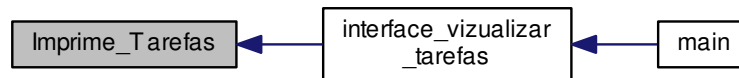
4.1.11.2 void Imprime_Tarefas (const grafo_priv_t * meu_grafo, int linha, int coluna)

Definição na linha 616 do arquivo grafos.cpp.

Referências lista_vert_codigo::dado, Celula_priv::duracao, Celula_priv::executada, Celula_priv::id_externo, Celula_priv::ini_min, lista_vert_codigo::next, Celula_priv::nome, Celula_priv::pre_req, Celula_priv::reqs e grafo_priv::tabela.

Referenciado por interface_vizualizar_tarefas().

Este é o diagrama das funções que utilizam esta função:



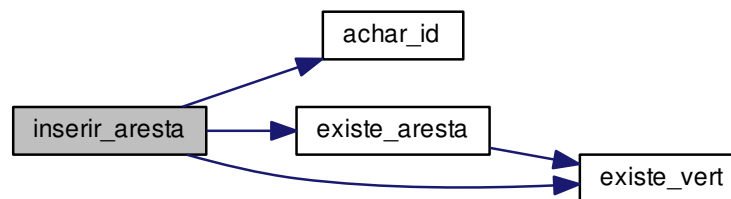
4.1.11.3 void inserir_aresta (grafo_priv_t * meu_grafo, int id_externo1, Celula_priv_t * celula2, int peso)

Definição na linha 250 do arquivo grafo.cpp.

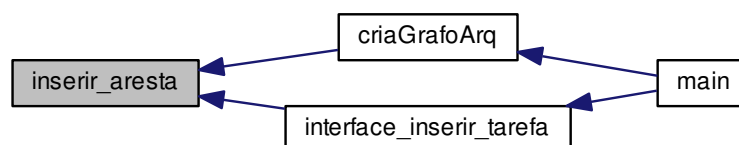
Referências `achar_id()`, `lista_aresta::destino`, `existe_aresta()`, `existe_vert()`, `FALSE_T`, `Celula_priv::id_externo`, `lista_aresta::next`, `lista_vert::next`, `lista_aresta::peso`, `lista_vert::sucessores`, `TRUE_T` e `grafo_priv::vert`.

Referenciado por `criaGrafoArq()` e `interface_inserir_tarefa()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.1.11.4 void inserir_vert (grafo_priv_t * meu_grafo, Celula_priv_t * celula)

Definição na linha 128 do arquivo grafo.cpp.

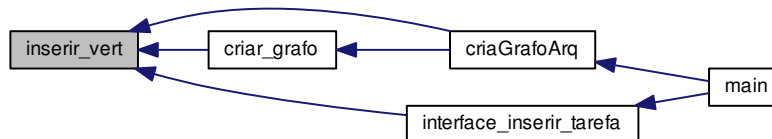
Referências lista_vert::antecessores, existe_vert(), FALSE_T, lista_vert_codigo::id, lista_vert::id, Celula_priv::id↔_externo, lista_vert::id_externo, Celula_priv::ini_min, lista_vert_codigo::next, lista_vert::next, Celula_priv::nome, Celula_priv::pre_req, lista_vert::sucessores, grafo_priv::tabela e grafo_priv::vert.

Referenciado por criaGrafoArq(), criar_grafo() e interface_inserir_tarefa().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



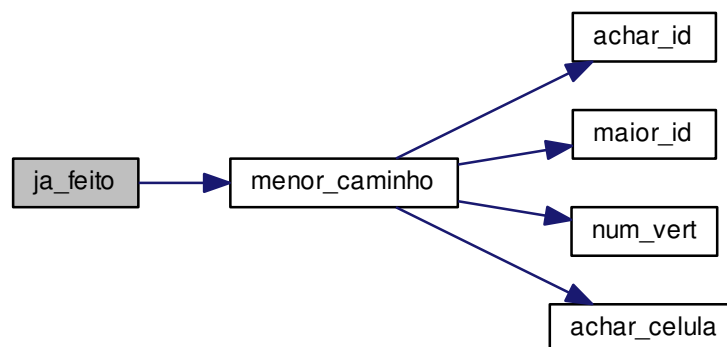
4.1.11.5 void jafeito (const grafo_priv_t * meu_grafo, int d)

Definição na linha 735 do arquivo grafo.cpp.

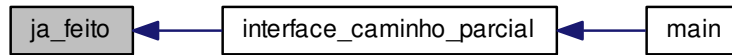
Referências lista_vert_codigo::id, menor_caminho(), lista_vert_codigo::next e grafo_priv::tabela.

Referenciado por interface_caminho_parcial().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.1.11.6 void Ler_Tarefas (grafo_priv_t * meu_grafo, Celula_priv_t * celula, const char * NomeArq)

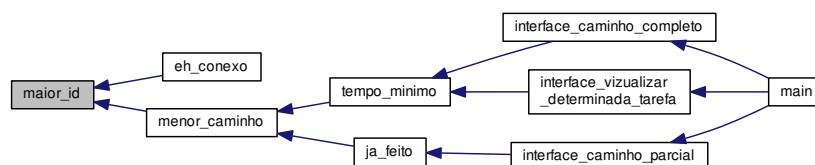
4.1.11.7 int maior_id (const grafo_priv_t * meu_grafo)

Definição na linha 490 do arquivo grafo.cpp.

Referências lista_vert_codigo::id, lista_vert_codigo::next e grafo_priv::tabela.

Referenciado por eh_conexo() e menor_caminho().

Este é o diagrama das funções que utilizam esta função:



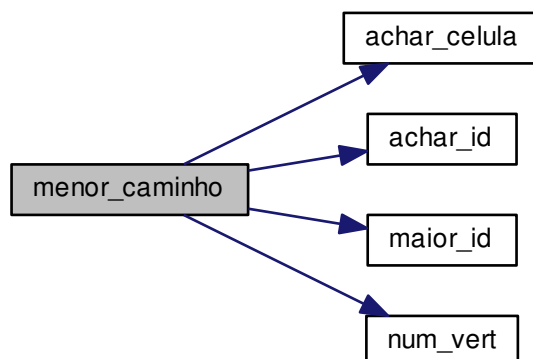
4.1.11.8 int menor_caminho (const grafo_priv_t * meu_grafo, int ** dist)

Definição na linha 524 do arquivo grafo.cpp.

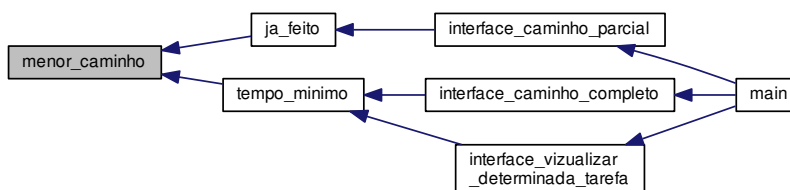
Referências achar_celula(), achar_id(), lista_aresta::destino, lista_vert::id, Celula_priv::ini_min, maior_id(), lista_aresta::next, lista_vert::next, num_vert(), lista_aresta::peso, lista_vert::sucessores e grafo_priv::vert.

Referenciado por ja_feito() e tempo_minimo().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.1.11.9 `int num_arestas (const grafo_priv_t * meu_grafo)`

Definição na linha 511 do arquivo `grafo.cpp`.

Referências `lista_aresta::next`, `lista_vert::next`, `lista_vert::sucessores` e `grafo_priv::vert`.

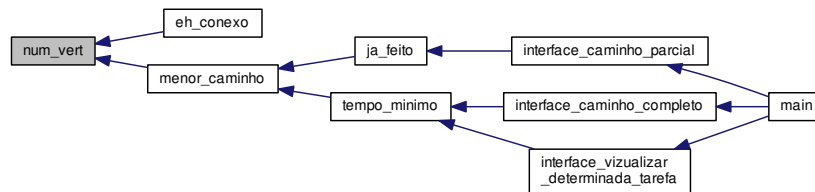
4.1.11.10 `int num_vert (const grafo_priv_t * meu_grafo)`

Definição na linha 501 do arquivo `grafo.cpp`.

Referências `lista_vert::next` e `grafo_priv::vert`.

Referenciado por `eh_conexo()` e `menor_caminho()`.

Este é o diagrama das funções que utilizam esta função:



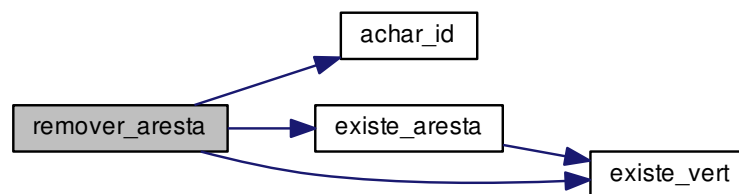
4.1.11.11 void remover_aresta (grafo_priv_t * meu_grafo, int id_externo1, int id_externo2)

Definição na linha 440 do arquivo grafo.cpp.

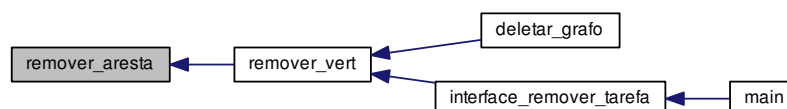
Referências achar_id(), lista_vert::antecessores, existe_aresta(), existe_vert(), lista_vert::id, lista_aresta::next, lista_vert::next, lista_vert::sucessores, TRUE_T e grafo_priv::vert.

Referenciado por remover_vert().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



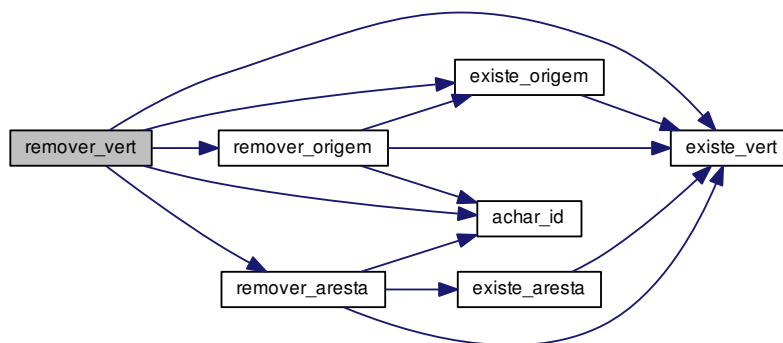
4.1.11.12 void remover_vert (grafo_priv_t * meu_grafo, int id_externo)

Definição na linha 340 do arquivo grafo.cpp.

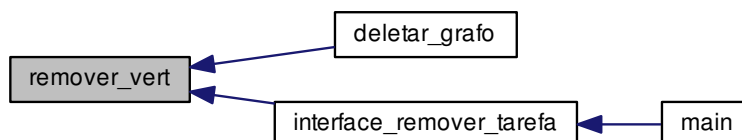
Referências achar_id(), lista_aresta::destino, existe_origem(), existe_vert(), lista_vert_codigo::id, lista_vert::id, lista_vert::id_externo, lista_vert_codigo::next, lista_vert::next, remover_aresta(), remover_origem(), grafo_priv::tabela, TRUE_T e grafo_priv::vert.

Referenciado por `deletar_grafo()` e `interface_remove_tarefa()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



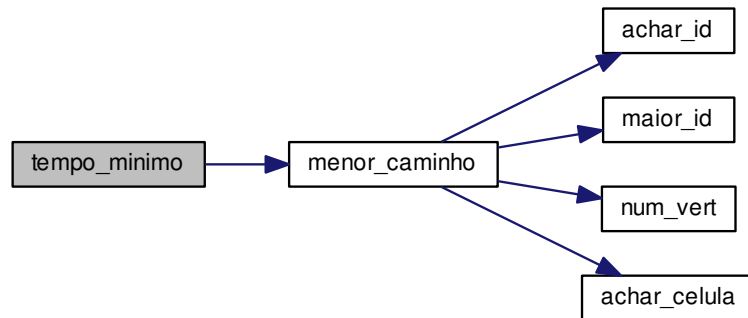
4.1.11.13 `int tempo_minimo (const grafo_priv_t * meu_grafo, int id_fim)`

Definição na linha 720 do arquivo `grafos.cpp`.

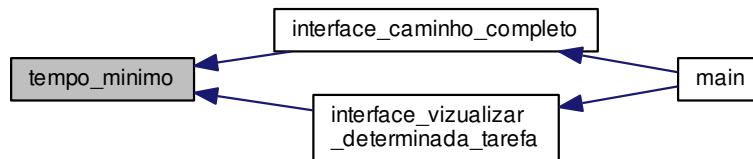
Referências `menor_caminho()`.

Referenciado por `interface_caminho_completo()` e `interface_vizualizar_determinada_tarefa()`.

Este é o diagrama das funções utilizadas por esta função:



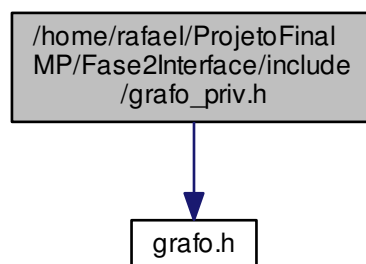
Este é o diagrama das funções que utilizam esta função:



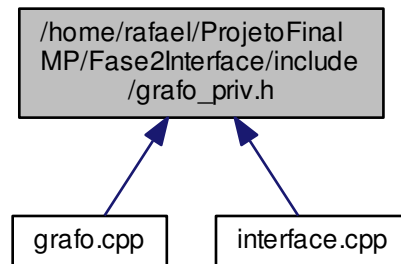
4.2 Referência do Arquivo /home/rafael/ProjetoFinalMP/Fase2Interface/include/grafopriv.h

```
#include "grafo.h"
```

Gráfico de dependência de inclusões para grafopriv.h:



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



Estruturas de Dados

- struct `Celula_priv`
- struct `lista_vert_codigo`
- struct `lista_aresta`
- struct `lista_vert`
- struct `lista_origem`
- struct `grafos_priv`

Definições de Tipos

- typedef struct `Celula_priv` `Celula_priv_t`
- typedef struct `lista_vert_codigo` `lista_vert_codigo_t`
- typedef struct `lista_aresta` `lista_aresta_t`
- typedef struct `lista_vert` `lista_vert_t`
- typedef struct `lista_origem` `lista_origem_t`
- typedef struct `grafos_priv` `grafos_priv_t`

Funções

- `resposta existe_origem` (const `grafos_priv_t` *meu_grafos, int id_externo)
- void `inserir_origem` (`grafos_priv_t` *meu_grafos, `Celula_priv_t` *celula)
- void `remover_origem` (`grafos_priv_t` *meu_grafos, int id_externo)

4.2.1 Definições dos tipos

4.2.1.1 typedef struct `Celula_priv` `Celula_priv_t`

4.2.1.2 typedef struct `grafos_priv` `grafos_priv_t`

4.2.1.3 typedef struct `lista_aresta` `lista_aresta_t`

4.2.1.4 typedef struct `lista_origem` `lista_origem_t`

4.2.1.5 `typedef struct lista_vert_codigo lista_vert_codigo_t`

4.2.1.6 `typedef struct lista_vert lista_vert_t`

4.2.2 Funções

4.2.2.1 `resposta existe_origem (const grafo_priv_t * meu_grafo, int id_externo)`

Definição na linha 74 do arquivo grafo.cpp.

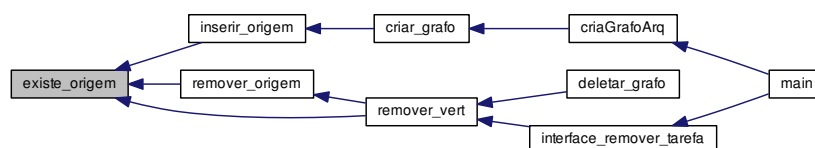
Referências `lista_origem::destino`, `existe_vert()`, `FALSE_T`, `lista_vert::id_externo`, `lista_origem::next`, `grafo_priv_t::origem` e `TRUE_T`.

Referenciado por `inserir_origem()`, `remover_origem()` e `remover_vert()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



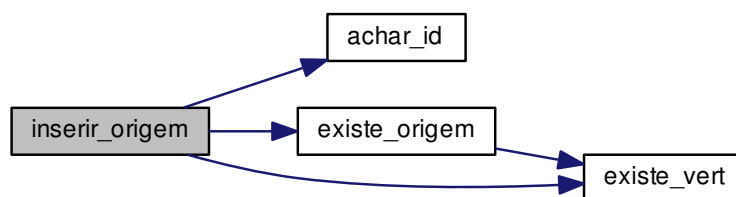
4.2.2.2 `void inserir_origem (grafo_priv_t * meu_grafo, Celula_priv_t * celula)`

Definição na linha 202 do arquivo grafo.cpp.

Referências `achar_id()`, `lista_origem::destino`, `existe_origem()`, `existe_vert()`, `FALSE_T`, `lista_vert::id`, `Celula_priv_t::id_externo`, `lista_vert::next`, `lista_origem::next`, `Celula_priv_t::nome`, `grafo_priv_t::origem`, `TRUE_T` e `grafo_priv_t::vert`.

Referenciado por `criar_grafo()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



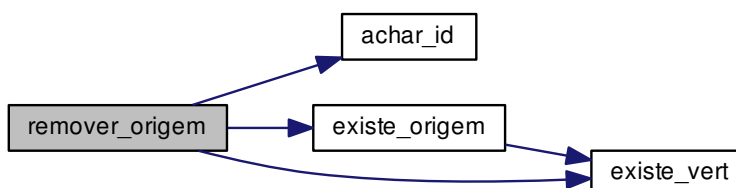
4.2.2.3 void remover_origem (grafo_priv_t * meu_grafo, int id_externo)

Definição na linha 413 do arquivo `grafo.cpp`.

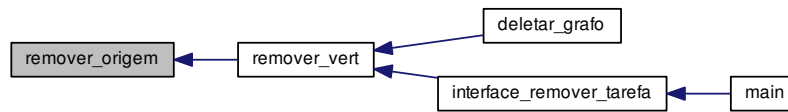
Referências `achar_id()`, `lista_origem::destino`, `existe_origem()`, `existe_vert()`, `lista_vert::id`, `lista_origem::next`, `grafo_priv::origem` e `TRUE_T`.

Referenciado por `remover_vert()`.

Este é o diagrama das funções utilizadas por esta função:



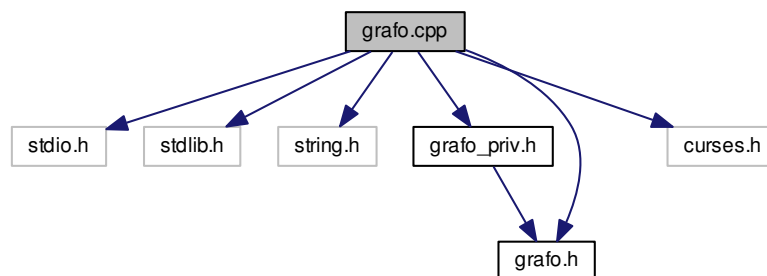
Este é o diagrama das funções que utilizam esta função:



4.3 Referência do Arquivo grafo.cpp

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "grafo_priv.h"
#include "grafo.h"
#include <curses.h>
```

Gráfico de dependência de inclusões para grafo.cpp:



Definições e Macros

- `#define DEBUG`

Funções

- `grafo_priv_t * criar_grafo (void)`
Cria grafo.
- `grafo_priv_t * deletar_grafo (grafo_priv_t *meu_grafo)`
Deleta grafo.
- `Celula_priv_t * cria_celula (int id_externo, int executada, int duracao, int ini_min, int pre_req, int *reqs, const char *nome)`
- `resposta existe_vert (const grafo_priv_t *meu_grafo, int id_externo)`
Existe vértice nome?
- `resposta existe_origem (const grafo_priv_t *meu_grafo, int id_externo)`
- `resposta existe_aresta (const grafo_priv_t *meu_grafo, int id_externo1, int id_externo2)`

- int `achar_id` (const `grafo_priv_t` *meu_grafo, int id_externo)
- `Celula_priv_t` * `achar_celula` (const `grafo_priv_t` *meu_grafo, int id_externo)
- void `inserir_vert` (`grafo_priv_t` *meu_grafo, `Celula_priv_t` *celula)
- void `inserir_origem` (`grafo_priv_t` *meu_grafo, `Celula_priv_t` *celula)
- void `inserir_aresta` (`grafo_priv_t` *meu_grafo, int id_externo1, `Celula_priv_t` *celula2, int peso)
- void `remover_vert` (`grafo_priv_t` *meu_grafo, int id_externo)
- void `remover_origem` (`grafo_priv_t` *meu_grafo, int id_externo)
- void `remover_aresta` (`grafo_priv_t` *meu_grafo, int id_externo1, int id_externo2)
- int `maior_id` (const `grafo_priv_t` *meu_grafo)
- int `num_vert` (const `grafo_priv_t` *meu_grafo)
- int `num_arestas` (const `grafo_priv_t` *meu_grafo)
- int `menor_caminho` (const `grafo_priv_t` *meu_grafo, int **dist)
- int `dfs` (const `grafo_priv_t` *meu_grafo, `lista_vert_t` *atual, int *marc)
- `resposta eh_conexo` (const `grafo_priv_t` *meu_grafo)
- void `Imprime_Tarefas` (const `grafo_priv_t` *meu_grafo, int linha, int coluna)
- void `Grava_Arq` (`grafo_priv_t` *meu_grafo, char *NomeArq)
- void `editar_celula` (`grafo_priv_t` *meu_grafo, int ID)
- int `tempo_minimo` (const `grafo_priv_t` *meu_grafo, int id_fim)
- void `ja_feito` (const `grafo_priv_t` *meu_grafo, int d)
- `grafo_priv_t` * `criaGrafoArq` (char *nomeArq)

4.3.1 Definições e macros

4.3.1.1 #define DEBUG

Definição na linha 10 do arquivo grafo.cpp.

4.3.2 Funções

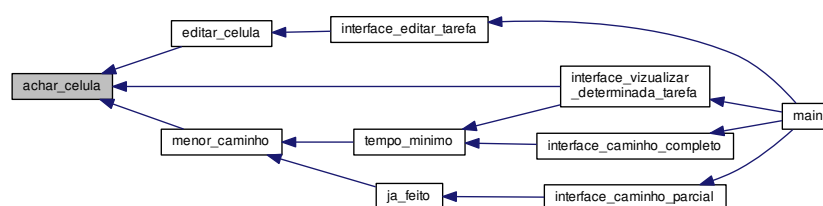
4.3.2.1 `Celula_priv_t`* `achar_celula` (const `grafo_priv_t` * `meu_grafo`, int `id_externo`)

Definição na linha 117 do arquivo grafo.cpp.

Referências `lista_vert_codigo::dado`, `Celula_priv::id_externo`, `lista_vert_codigo::next` e `grafo_priv::tabela`.

Referenciado por `editar_celula()`, `interface_vizualizar_determinada_tarefa()` e `menor_caminho()`.

Este é o diagrama das funções que utilizam esta função:



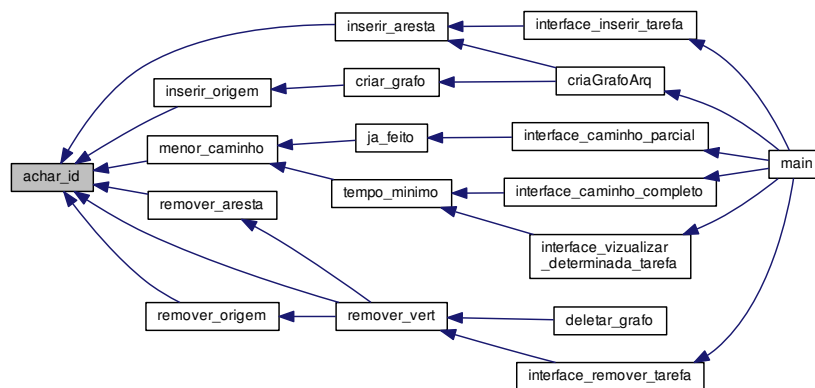
4.3.2.2 `int achar_id (const grafo_priv_t * meu_grafo, int id_externo)`

Definição na linha 107 do arquivo `grafo.cpp`.

Referências `lista_vert_codigo::dado`, `lista_vert_codigo::id`, `Celula_priv::id_externo`, `lista_vert_codigo::next` e `grafo_priv::tabela`.

Referenciado por `inserir_aresta()`, `inserir_origem()`, `menor_caminho()`, `remover_aresta()`, `remover_origem()` e `remover_vert()`.

Este é o diagrama das funções que utilizam esta função:



4.3.2.3 `Celula_priv_t* cria_celula (int id_externo, int executada, int duracao, int ini_min, int pre_req, int * reqs, const char * nome)`

Definição na linha 43 do arquivo `grafo.cpp`.

Referências `Celula_priv::duracao`, `Celula_priv::executada`, `Celula_priv::id_externo`, `Celula_priv::ini_min`, `Celula_priv::nome`, `Celula_priv::pre_req` e `Celula_priv::reqs`.

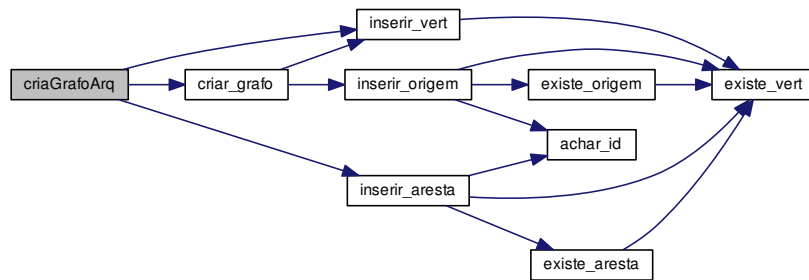
4.3.2.4 `grafo_priv_t* criaGrafoArq (char * nomeArq)`

Definição na linha 767 do arquivo `grafo.cpp`.

Referências `criar_grafo()`, `Celula_priv::duracao`, `Celula_priv::executada`, `Celula_priv::id_externo`, `Celula_priv::ini_min`, `inserir_aresta()`, `inserir_vert()`, `Celula_priv::nome`, `Celula_priv::pre_req` e `Celula_priv::reqs`.

Referenciado por `main()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.3.2.5 grafo_priv_t* criar_grafo (void)

Cria grafo.

4.3.3 Descrição

Aloca a memória necessária e inicializa um grafo

4.3.4 Parâmetros

Não ha parâmetros, a alocação e inicializam não dependem de nenhum parâmetro do usuário

Retorna

Se retorna um ponteiro para a grafo criado

4.3.5 Assertiva de saída

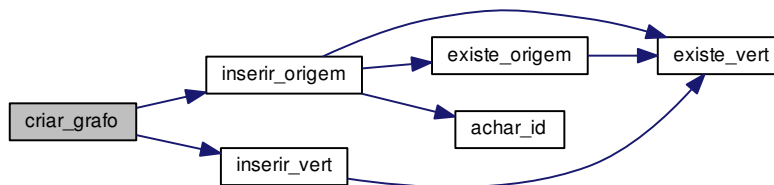
O grafo gerado é consistente e não possui nenhum vértice, origem ou aresta.

Definição na linha 12 do arquivo grafo.cpp.

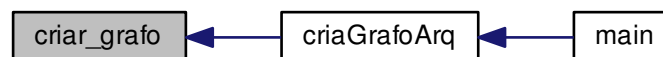
Referências Celula_priv::executada, Celula_priv::id_externo, Celula_priv::ini_min, inserir_origem(), inserir_vert(), Celula_priv::nome, grafo_priv::origem, Celula_priv::pre_req, grafo_priv::tabela e grafo_priv::vert.

Referenciado por criaGrafoArq().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.3.5.1 grafo_priv_t* deletar_grafo (grafo_priv_t * meu_grafo)

Deleta grafo.

4.3.6 Descrição

Desaloca toda a memória utilizada pelo grafo

Parâmetros

<i>meu_grafo</i>	- Deve ser passado um ponteiro para um grafo inicializado
------------------	---

Retorna

Retorna um ponteiro para o grafo, que será NULL. O valor de retorno é muito importante, uma vez que se ele não for utilizado o grafo do usuário apontará para um endereço não alocado e qualquer tentativa de utilizá-lo poderá gerar erros no sistema. Caso o grafo passado não tenha sido inicializado, o programa poderá parar a execução

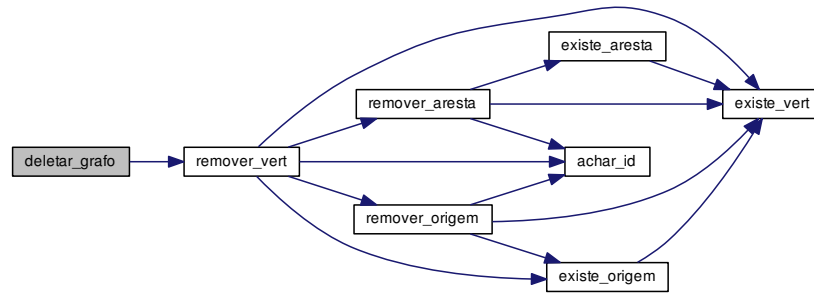
4.3.7 Assertiva de saída

O grafo já deve ter sido inicializado por [criar_grafo\(\)](#) Assertiva de saída Se retornará um ponteiro para o mesmo grafo passado, após a deleção que será NULL.

Definição na linha 32 do arquivo grafo.cpp.

Referências `lista_vert_codigo::dado`, `Celula_priv::id_externo`, `remover_vert()` e `grafo_priv::tabela`.

Este é o diagrama das funções utilizadas por esta função:



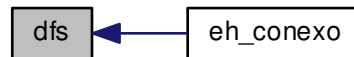
4.3.7.1 `int dfs (const grafo_priv_t * meu_grafo, lista_vert_t * atual, int * marc)`

Definição na linha 565 do arquivo `grafo.cpp`.

Referências `lista_aresta::destino`, `lista_vert::id`, `lista_aresta::next` e `lista_vert::sucessores`.

Referenciado por `eh_conexo()`.

Este é o diagrama das funções que utilizam esta função:



4.3.7.2 `void editar_celula (grafo_priv_t * meu_grafo, int ID)`

Definição na linha 664 do arquivo `grafo.cpp`.

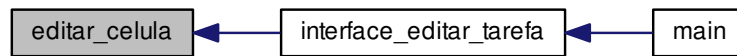
Referências `achar_celula()`, `Celula_priv::duracao`, `Celula_priv::executada`, `Celula_priv::id_externo`, `Celula_priv::ini_min`, `Celula_priv::nome`, `Celula_priv::pre_req` e `Celula_priv::reqs`.

Referenciado por `interface_editar_tarefa()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:

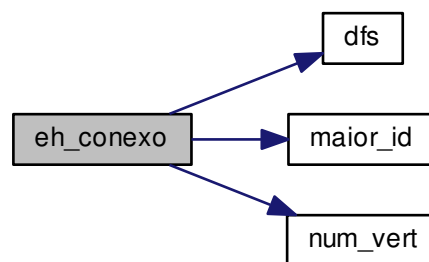


4.3.7.3 resposta eh_conexo (const grafo_priv_t * meu_grafo)

Definição na linha 581 do arquivo grafo.cpp.

Referências lista_origem::destino, dfs(), FALSE_T, lista_vert::id, maior_id(), lista_origem::next, num_vert(), grafo←_priv::origem e TRUE_T.

Este é o diagrama das funções utilizadas por esta função:



4.3.7.4 resposta existe_aresta (const grafo_priv_t * meu_grafo, int id_externo1, int id_externo2)

Definição na linha 87 do arquivo grafo.cpp.

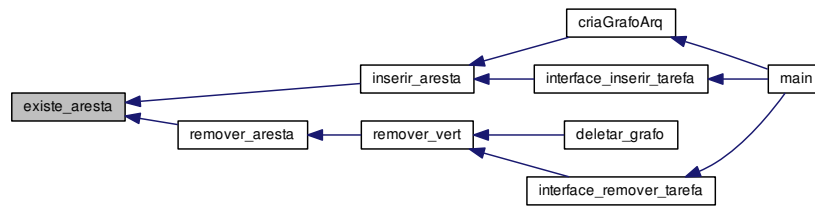
Referências lista_aresta::destino, existe_vert(), FALSE_T, lista_vert::id_externo, lista_aresta::next, lista_vert::next, lista_vert::sucessores, TRUE_T e grafo_priv::vert.

Referenciado por inserir_aresta() e remover_aresta().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.3.7.5 resposta existe_origem (const grafo_priv_t * meu_grafo, int id_externo)

Definição na linha 74 do arquivo grafo.cpp.

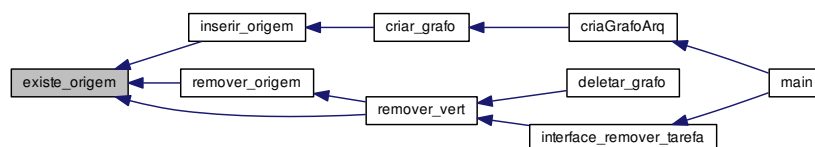
Referências lista_origem::destino, existe_vert(), FALSE_T, lista_vert::id_externo, lista_origem::next, grafo_priv->origem e TRUE_T.

Referenciado por inserir_origem(), remover_origem() e remover_vert().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.3.7.6 resposta existe_vert (const grafo_priv_t * meu_grafo, int id_externo)

Existe vértice nome?

4.3.8 Descrição

Verifica a existência de um vértice dado

Parâmetros

<i>meu_grafo</i>	- Deve ser passado um ponteiro para um grafo inicializado
<i>nome</i>	- Deve ser passado uma string de até 100 caracteres, mais o caracter zero terminal. Se a string tiver mais que 100 caracteres, o vértice não será encontrado, uma vez que 100 é o maior número de caracteres que um vértice pode ter.

Retorna

Retorna uma resposta, que será TRUE caso exista o vértice, e FALSE caso não exista.

4.3.9 Assertiva de entrada

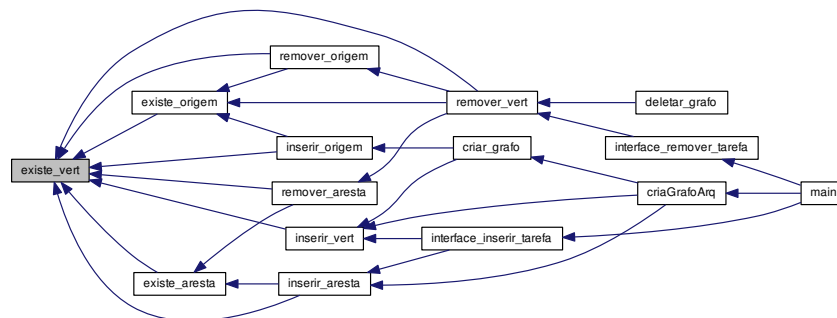
O grafo já deve ter sido inicializado por [criar_grafo\(\)](#), se não for o programa pode ser interrompido. `strlen(nome) <= 100` Assertivas de saída O retorno será TRUE ou FALSE O conteúdo do grafo e da string passados não serão modificados

Definição na linha 64 do arquivo grafo.cpp.

Referências `lista_vert_codigo::dado`, `FALSE_T`, `Celula_priv::id_externo`, `lista_vert_codigo::next`, `grafo_priv::tabela` e `TRUE_T`.

Referenciado por `existe_aresta()`, `existe_origem()`, `inserir_aresta()`, `inserir_origem()`, `inserir_vert()`, `remover_aresta()`, `remover_origem()` e `remover_vert()`.

Este é o diagrama das funções que utilizam esta função:



4.3.9.1 void Grava_Arq (grafo_priv_t * meu_grafo, char * NomeArq)

Definição na linha 637 do arquivo grafo.cpp.

Referências `lista_vert_codigo::dado`, `Celula_priv::duracao`, `Celula_priv::executada`, `Celula_priv::id_externo`, `Celula_priv::ini_min`, `lista_vert_codigo::next`, `Celula_priv::nome`, `Celula_priv::pre_req`, `Celula_priv::reqs` e `grafo_priv::tabela`.

Referenciado por `main()`.

Este é o diagrama das funções que utilizam esta função:



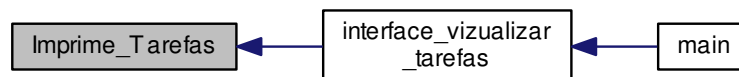
4.3.9.2 void Imprime_Tarefas (const grafo_priv_t * meu_grafo, int linha, int coluna)

Definição na linha 616 do arquivo grafo.cpp.

Referências lista_vert_codigo::dado, Celula_priv::duracao, Celula_priv::executada, Celula_priv::id_externo, Celula_priv::ini_min, lista_vert_codigo::next, Celula_priv::nome, Celula_priv::pre_req, Celula_priv::reqs e grafo_priv::tabela.

Referenciado por interface_vizualizar_tarefas().

Este é o diagrama das funções que utilizam esta função:



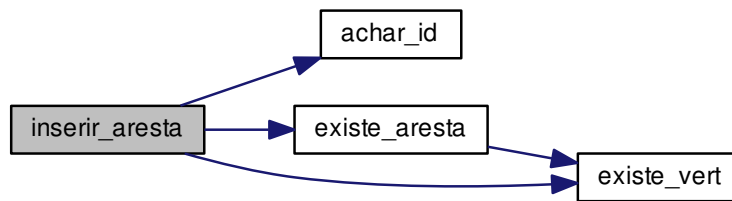
4.3.9.3 void inserir_aresta (grafo_priv_t * meu_grafo, int id_externo1, Celula_priv_t * celula2, int peso)

Definição na linha 250 do arquivo grafo.cpp.

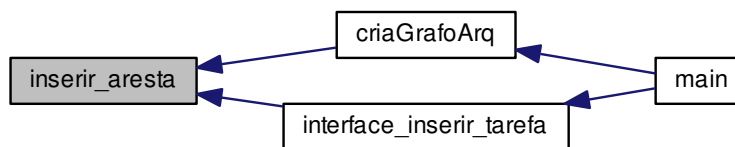
Referências achar_id(), lista_aresta::destino, existe_aresta(), existe_vert(), FALSE_T, Celula_priv::id_externo, lista_aresta::next, lista_vert::next, lista_aresta::peso, lista_vert::sucessores, TRUE_T e grafo_priv::vert.

Referenciado por criaGrafoArq() e interface_inserir_tarefa().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



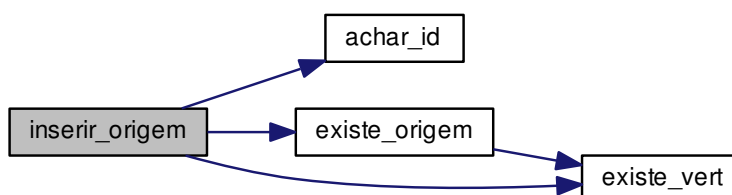
4.3.9.4 void inserir_origem (grafo_priv_t * meu_grafo, Celula_priv_t * celula)

Definição na linha 202 do arquivo `grafo.cpp`.

Referências `achar_id()`, `lista_origem::destino`, `existe_origem()`, `existe_vert()`, `FALSE_T`, `lista_vert::id`, `Celula_priv::id_externo`, `lista_vert::next`, `lista_origem::next`, `Celula_priv::nome`, `grafo_priv::origem`, `TRUE_T` e `grafo_priv::vert`.

Referenciado por `criar_grafo()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.3.9.5 void inserir_vert (grafo_priv_t * meu_grafo, Celula_priv_t * celula)

Definição na linha 128 do arquivo grafo.cpp.

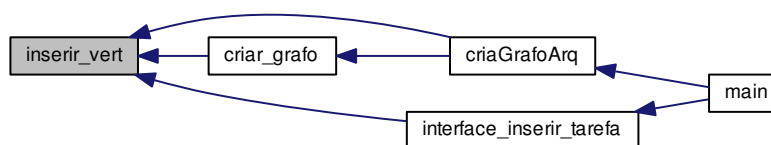
Referências `lista_vert::antecessores`, `existe_vert()`, `FALSE_T`, `lista_vert_codigo::id`, `lista_vert::id`, `Celula_priv::id`, `lista_vert::id_externo`, `Celula_priv::ini_min`, `lista_vert_codigo::next`, `lista_vert::next`, `Celula_priv::nome`, `Celula_priv::pre_req`, `lista_vert::sucessores`, `grafo_priv::tabela` e `grafo_priv::vert`.

Referenciado por `criaGrafoArq()`, `criar_grafo()` e `interface_inserir_tarefa()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



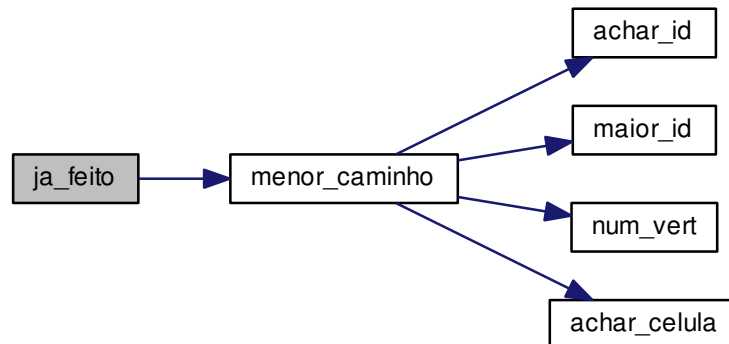
4.3.9.6 void jafeito (const grafo_priv_t * meu_grafo, int d)

Definição na linha 735 do arquivo grafo.cpp.

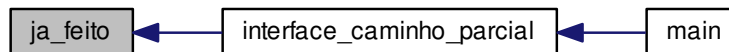
Referências `lista_vert_codigo::id`, `menor_caminho()`, `lista_vert_codigo::next` e `grafo_priv::tabela`.

Referenciado por `interface_caminho_parcial()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



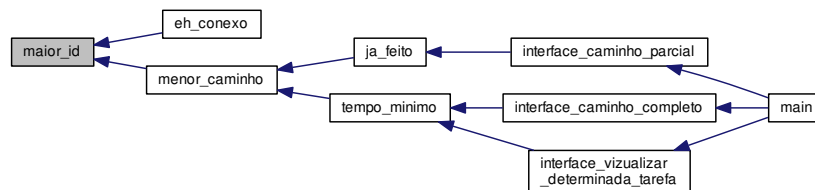
4.3.9.7 int maior_id (const grafo_priv_t * meu_grafo)

Definição na linha 490 do arquivo `grafo.cpp`.

Referências `lista_vert_codigo::id`, `lista_vert_codigo::next` e `grafo_priv::tabela`.

Referenciado por `eh_conexo()` e `menor_caminho()`.

Este é o diagrama das funções que utilizam esta função:



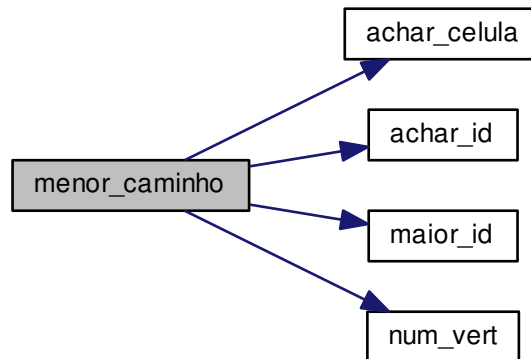
4.3.9.8 `int menor_caminho (const grafo_priv_t * meu_grafo, int ** dist)`

Definição na linha 524 do arquivo grafo.cpp.

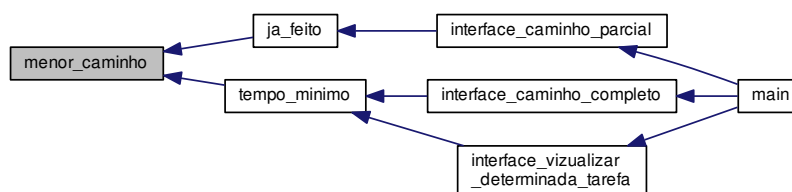
Referências `achar_celula()`, `achar_id()`, `lista_aresta::destino`, `lista_vert::id`, `Celula_priv::ini_min`, `maior_id()`, `lista_aresta::next`, `lista_vert::next`, `num_vert()`, `lista_aresta::peso`, `lista_vert::sucessores` e `grafo_priv::vert`.

Referenciado por `ja_feito()` e `tempo_minimo()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



4.3.9.9 `int num_arestas (const grafo_priv_t * meu_grafo)`

Definição na linha 511 do arquivo grafo.cpp.

Referências `lista_aresta::next`, `lista_vert::next`, `lista_vert::sucessores` e `grafo_priv::vert`.

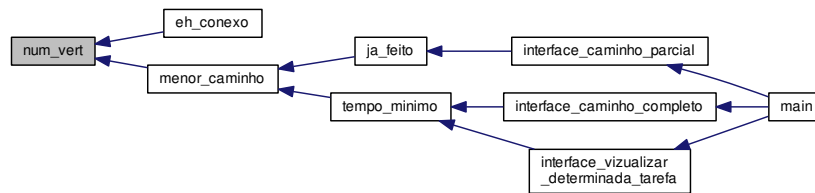
4.3.9.10 `int num_vert (const grafo_priv_t * meu_grafo)`

Definição na linha 501 do arquivo grafo.cpp.

Referências `lista_vert::next` e `grafo_priv::vert`.

Referenciado por `eh_conexo()` e `menor_caminho()`.

Este é o diagrama das funções que utilizam esta função:



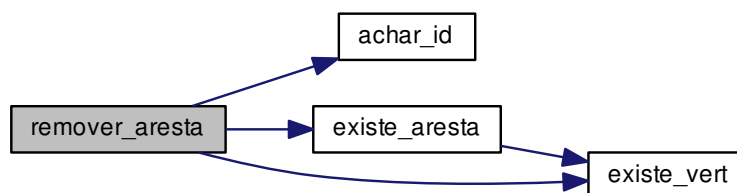
4.3.9.11 void remover_aresta (grafo_priv_t * meu_grafo, int id_externo1, int id_externo2)

Definição na linha 440 do arquivo grafo.cpp.

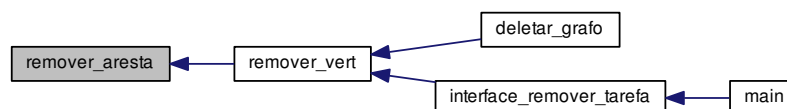
Referências `achar_id()`, `lista_vert::antecessores`, `existe_aresta()`, `existe_vert()`, `lista_vert::id`, `lista_aresta::next`, `lista_vert::next`, `lista_vert::sucessores`, `TRUE_T` e `grafo_priv::vert`.

Referenciado por `remover_vert()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



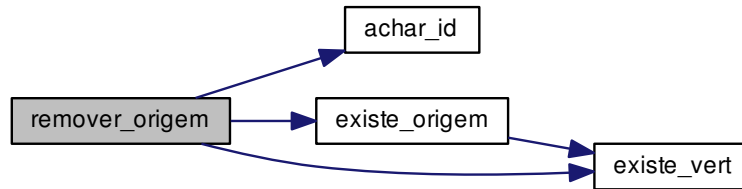
4.3.9.12 void remover_origem (grafo_priv_t * meu_grafo, int id_externo)

Definição na linha 413 do arquivo grafo.cpp.

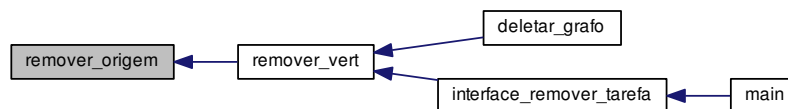
Referências `achar_id()`, `lista_origem::destino`, `existe_origem()`, `existe_vert()`, `lista_vert::id`, `lista_origem::next`, `grafo_priv::origem` e `TRUE_T`.

Referenciado por remover_vert().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



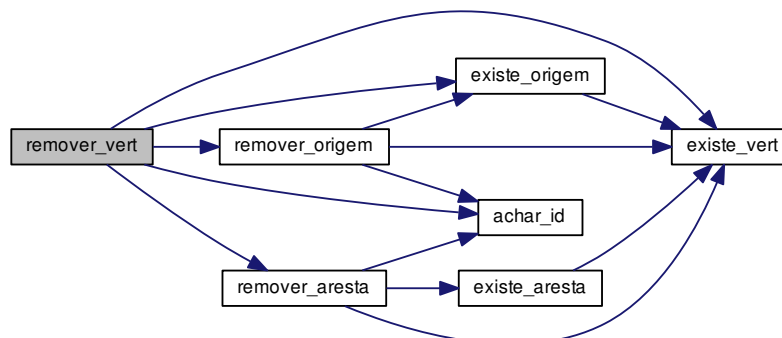
4.3.9.13 void remover_vert (grafo_priv_t * meu_grafo, int id_externo)

Definição na linha 340 do arquivo grafo.cpp.

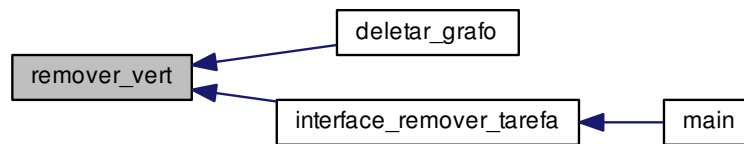
Referências achar_id(), lista_aresta::destino, existe_origem(), existe_vert(), lista_vert_codigo::id, lista_vert::id, lista_vert::id_externo, lista_vert_codigo::next, lista_vert::next, remover_aresta(), remover_origem(), grafo_priv_t::tabela, TRUE_T e grafo_priv::vert.

Referenciado por deletar_grafo() e interface_remover_tarefa().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



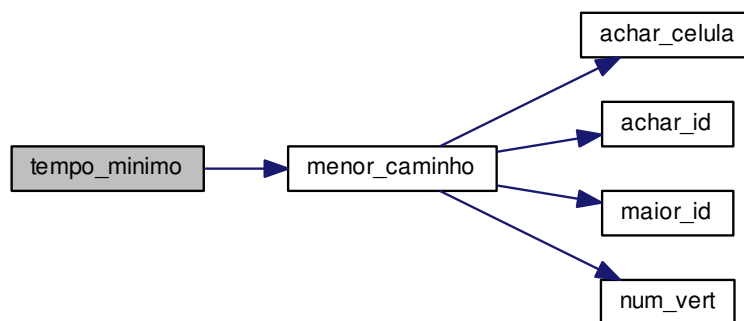
4.3.9.14 `int tempo_minimo (const grafo_priv_t * meu_grafo, int id_fim)`

Definição na linha 720 do arquivo `grafo.cpp`.

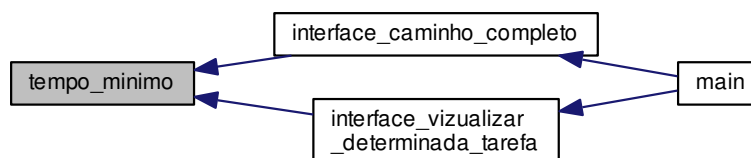
Referências `menor_caminho()`.

Referenciado por `interface_caminho_completo()` e `interface_vizualizar_determinada_tarefa()`.

Este é o diagrama das funções utilizadas por esta função:



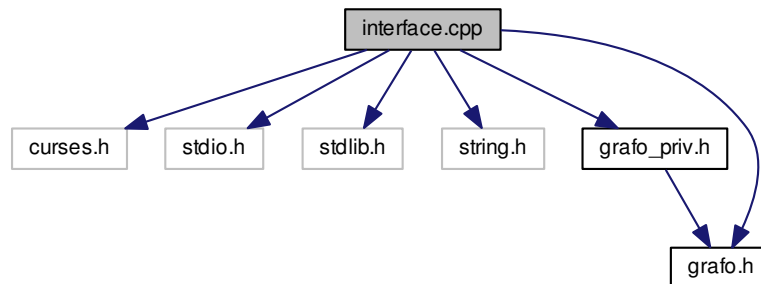
Este é o diagrama das funções que utilizam esta função:



4.4 Referência do Arquivo interface.cpp

```
#include <urses.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "grafo_priv.h"
#include "grafo.h"
```

Gráfico de dependência de inclusões para interface.cpp:



Definições e Macros

- #define NCURSES_CONST

Funções

- void interface_remover_tarefa (grafo_priv_t *meu_grafo)
- void interface_editar_tarefa (grafo_priv_t *meu_grafo)
- void interface_inserir_tarefa (grafo_priv_t *meu_grafo)
- void interface_caminho_completo (const grafo_priv_t *meu_grafo)
- void interface_caminho_parcial (grafo_priv_t *meu_grafo)
- void interface_vizualizar_tarefas (const grafo_priv_t *meu_grafo)
- void interface_vizualizar_determinada_tarefa (const grafo_priv_t *meu_grafo)
- int main ()

4.4.1 Definições e macros

4.4.1.1 #define NCURSES_CONST

Definição na linha 1 do arquivo interface.cpp.

4.4.2 Funções

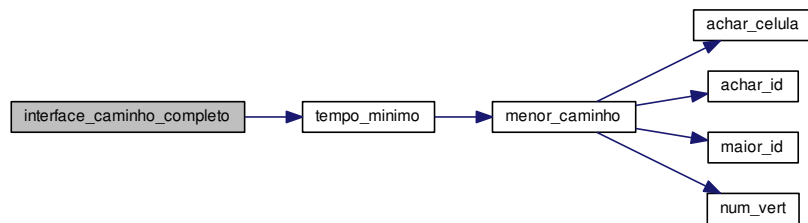
4.4.2.1 void interface_caminho_completo (const grafo_priv_t * meu_grafo)

Definição na linha 114 do arquivo interface.cpp.

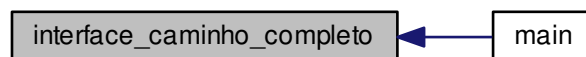
Referências lista_vert_codigo::dato, Celula_priv::id_externo, lista_vert_codigo::next, grafo_priv::tabela e tempo_minimo().

Referenciado por main().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



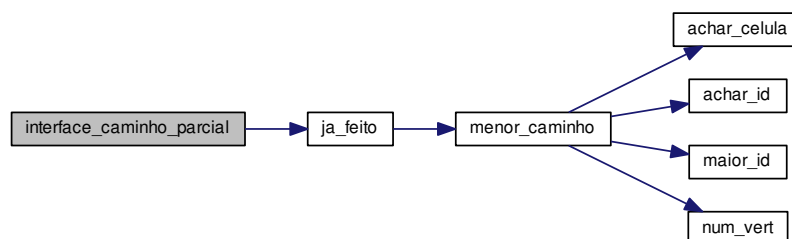
4.4.2.2 void interface_caminho_parcial (grafo_priv_t * meu_grafo)

Definição na linha 141 do arquivo interface.cpp.

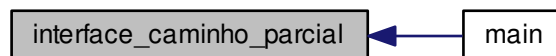
Referências ja_feito().

Referenciado por main().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



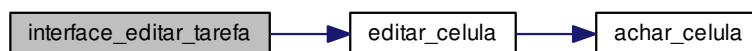
4.4.2.3 void interface_editar_tarefa (grafo_priv_t * meu_grafo)

Definição na linha 43 do arquivo interface.cpp.

Referências editar_celula().

Referenciado por main().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



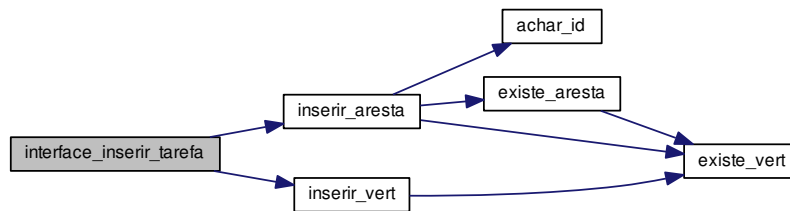
4.4.2.4 void interface_inserir_tarefa (grafo_priv_t * meu_grafo)

Definição na linha 64 do arquivo interface.cpp.

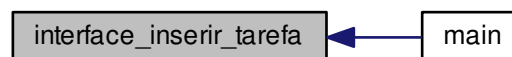
Referências Celula_priv::duracao, Celula_priv::executada, Celula_priv::id_externo, Celula_priv::ini_min, inserir_↔ aresta(), inserir_vert(), Celula_priv::nome, Celula_priv::pre_req e Celula_priv::reqs.

Referenciado por main().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



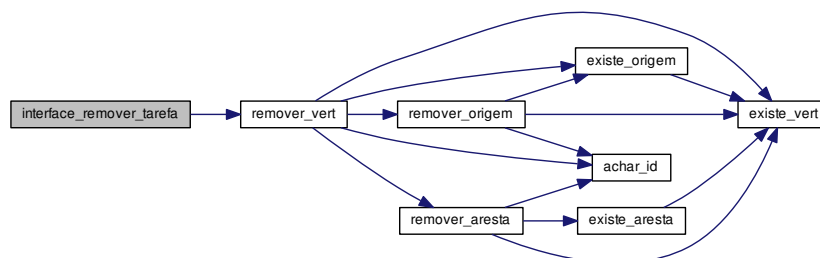
4.4.2.5 void interface_remover_tarefa (grafo_priv_t * meu_grafo)

Definição na linha 21 do arquivo interface.cpp.

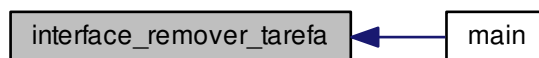
Referências `remover_vert()`.

Referenciado por `main()`.

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



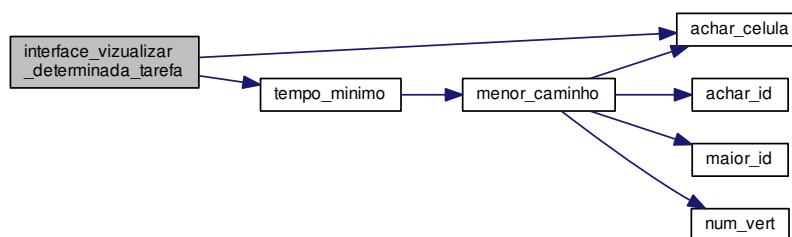
4.4.2.6 void interface_vizualizar_determinada_tarefa (const grafo_priv_t * meu_grafo)

Definição na linha 187 do arquivo interface.cpp.

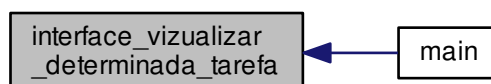
Referências achar_celula(), Celula_priv::duracao, Celula_priv::executada, Celula_priv::id_externo, Celula_priv::ini_min, Celula_priv::nome, Celula_priv::pre_req, Celula_priv::reqs e tempo_minimo().

Referenciado por main().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:



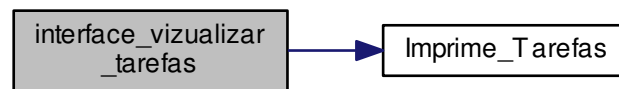
4.4.2.7 void interface_vizualizar_tarefas (const grafo_priv_t * meu_grafo)

Definição na linha 170 do arquivo interface.cpp.

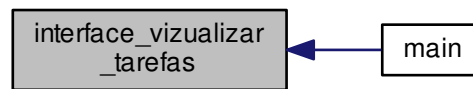
Referências Imprime_Tarefas().

Referenciado por main().

Este é o diagrama das funções utilizadas por esta função:



Este é o diagrama das funções que utilizam esta função:

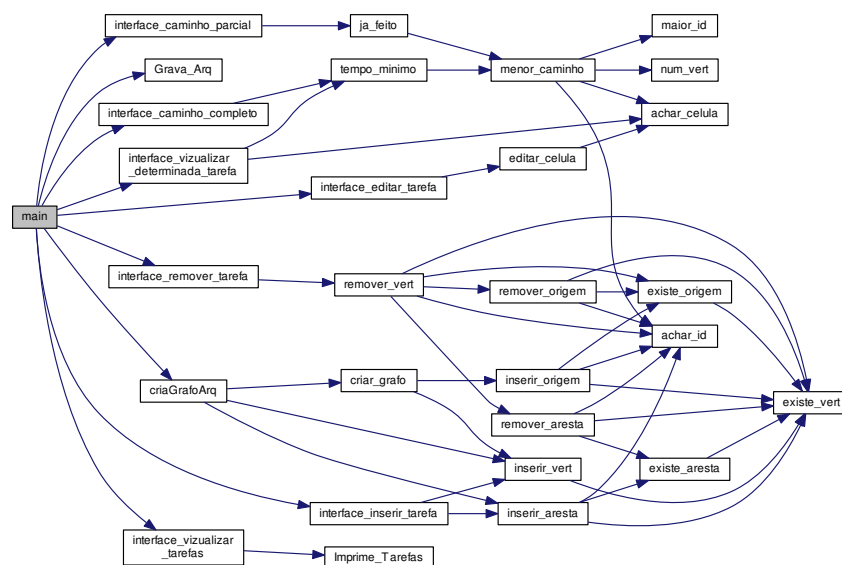


4.4.2.8 int main ()

Definição na linha 225 do arquivo interface.cpp.

Referências criaGrafoArq(), Grava_Arq(), interface_caminho_completo(), interface_caminho_parcial(), interface_↔_editar_tarefa(), interface_inserir_tarefa(), interface_remover_tarefa(), interface_vizualizar_determinada_tarefa() e interface_vizualizar_tarefas().

Este é o diagrama das funções utilizadas por esta função:



Índice Remissivo

/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafo.
h, 13
/home/rafael/ProjetoFinalMP/Fase2Interface/include/grafo.
_priv.h, 28

achar_celula
 grafo.cpp, 33
 grafo.h, 15

achar_id
 grafo.cpp, 33
 grafo.h, 15

antecessores
 lista_vert, 10

Celula_priv, 5
 duracao, 5
 executada, 5
 id_externo, 5
 ini_min, 6
 nome, 6
 pre_req, 6
 reqs, 6

Celula_priv_t
 grafo.h, 14
 grafo_priv.h, 29

cria_celula
 grafo.cpp, 34
 grafo.h, 15

criaGrafoArq
 grafo.cpp, 34
 grafo.h, 16

criar_grafo
 grafo.cpp, 35
 grafo.h, 16

DEBUG
 grafo.cpp, 33

dado
 lista_vert_codigo, 11

deletar_grafo
 grafo.cpp, 36
 grafo.h, 17

destino
 lista_aresta, 8
 lista_origem, 9

dfs
 grafo.cpp, 37

duracao
 Celula_priv, 5

 grafo.cpp, 37
 grafo.h, 18

eh_conexo
 grafo.cpp, 38
 grafo.h, 19

executada
 Celula_priv, 5

existe_aresta
 grafo.cpp, 38
 grafo.h, 19

existe_origem
 grafo.cpp, 39
 grafo_priv.h, 30

existe_vert
 grafo.cpp, 39
 grafo.h, 20

FALSE_T
 grafo.h, 14

grafo.cpp, 32
 achar_celula, 33
 achar_id, 33
 cria_celula, 34
 criaGrafoArq, 34
 criar_grafo, 35
 DEBUG, 33
 deletar_grafo, 36
 dfs, 37
 editar_celula, 37
 eh_conexo, 38
 existe_aresta, 38
 existe_origem, 39
 existe_vert, 39
 Grava_Arq, 40
 Imprime_Tarefas, 41
 inserir_aresta, 41
 inserir_origem, 42
 inserir_vert, 43
 ja_feito, 43
 maior_id, 44
 menor_caminho, 44
 num_arestas, 45
 num_vert, 45
 remover_aresta, 46
 remover_origem, 46
 remover_vert, 47
 tempo_minimo, 48

grafo.h

- achar_celula, 15
- achar_id, 15
- Celula_priv_t, 14
- cria_celula, 15
- criaGrafoArq, 16
- criar_grafo, 16
- deletar_grafo, 17
- editar_celula, 18
- eh_conexo, 19
- existe_aresta, 19
- existe_vert, 20
- FALSE_T, 14
- grafo_priv_t, 14
- Grava_Arq, 21
- Imprime_Tarefas, 21
- inserir_aresta, 22
- inserir_vert, 22
- ja_feito, 23
- Ler_Tarefas, 24
- maior_id, 24
- menor_caminho, 24
- num_arestas, 25
- num_vert, 25
- remover_aresta, 26
- remover_vert, 26
- resp, 14
- resposta, 14
- TRUE_T, 14
- tempo_minimo, 27
- grafo_priv, 6
 - origem, 7
 - tabela, 7
 - vert, 7
- grafo_priv.h
 - Celula_priv_t, 29
 - existe_origem, 30
 - grafo_priv_t, 29
 - inserir_origem, 30
 - lista_aresta_t, 29
 - lista_origem_t, 29
 - lista_vert_codigo_t, 29
 - lista_vert_t, 30
 - remover_origem, 31
- grafo_priv_t
 - grafo.h, 14
 - grafo_priv.h, 29
- Grava_Arq
 - grafo.cpp, 40
 - grafo.h, 21
- id
 - lista_vert, 10
 - lista_vert_codigo, 12
- id_externo
 - Celula_priv, 5
 - lista_vert, 10
- Imprime_Tarefas
 - grafo.cpp, 41
 - grafo.h, 21
- ini_min
 - Celula_priv, 6
- inserir_aresta
 - grafo.cpp, 41
 - grafo.h, 22
- inserir_origem
 - grafo.cpp, 42
 - grafo_priv.h, 30
- inserir_vert
 - grafo.cpp, 43
 - grafo.h, 22
- interface.cpp, 49
 - interface_caminho_completo, 49
 - interface_caminho_parcial, 50
 - interface_editar_tarefa, 51
 - interface_inserir_tarefa, 51
 - interface_remover_tarefa, 52
 - interface_vizualizar_determinada_tarefa, 53
 - interface_vizualizar_tarefas, 53
 - main, 54
 - NCURSES_CONST, 49
- interface_caminho_completo
 - interface.cpp, 49
- interface_caminho_parcial
 - interface.cpp, 50
- interface_editar_tarefa
 - interface.cpp, 51
- interface_inserir_tarefa
 - interface.cpp, 51
- interface_remover_tarefa
 - interface.cpp, 52
- interface_vizualizar_determinada_tarefa
 - interface.cpp, 53
- interface_vizualizar_tarefas
 - interface.cpp, 53
- ja_feito
 - grafo.cpp, 43
 - grafo.h, 23
- Ler_Tarefas
 - grafo.h, 24
- lista_aresta, 7
 - destino, 8
 - next, 8
 - peso, 8
- lista_aresta_t
 - grafo_priv.h, 29
- lista_origem, 9
 - destino, 9
 - next, 9
- lista_origem_t
 - grafo_priv.h, 29
- lista_vert, 9
 - antecessores, 10
 - id, 10
 - id_externo, 10
 - next, 10
 - sucessores, 11

lista_vert_codigo, 11
 dado, 11
 id, 12
 next, 12
lista_vert_codigo_t
 grafo_priv.h, 29
lista_vert_t
 grafo_priv.h, 30

main
 interface.cpp, 54
maior_id
 grafo.cpp, 44
 grafo.h, 24
menor_caminho
 grafo.cpp, 44
 grafo.h, 24
Modulo, 12

NCURSES_CONST
 interface.cpp, 49
next
 lista_aresta, 8
 lista_origem, 9
 lista_vert, 10
 lista_vert_codigo, 12
nome
 Celula_priv, 6
num_arestas
 grafo.cpp, 45
 grafo.h, 25
num_vert
 grafo.cpp, 45
 grafo.h, 25

origem
 grafo_priv, 7

peso
 lista_aresta, 8
pre_req
 Celula_priv, 6

remover_aresta
 grafo.cpp, 46
 grafo.h, 26
remover_origem
 grafo.cpp, 46
 grafo_priv.h, 31
remover_vert
 grafo.cpp, 47
 grafo.h, 26
reqs
 Celula_priv, 6
resp
 grafo.h, 14
resposta
 grafo.h, 14

sucessores
 lista_vert, 11

TRUE_T
 grafo.h, 14
tabela
 grafo_priv, 7
tempo_minimo
 grafo.cpp, 48
 grafo.h, 27

vert
 grafo_priv, 7