

TrabalhoMP

1.0

Generated by Doxygen 1.8.1.2

Sat Nov 21 2015 02:24:21

Contents

1	TrabalhoMP	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	7
4.1	adaptador Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Field Documentation	7
4.1.2.1	entradas	7
4.1.2.2	nome	8
4.1.2.3	posicao	8
4.1.2.4	proximo	8
4.1.2.5	quantidadeSaidas	8
4.1.2.6	recursoRecebido	8
4.1.2.7	saidas	8
4.2	cidade Struct Reference	8
4.2.1	Detailed Description	8
4.2.2	Field Documentation	9
4.2.2.1	entradas	9
4.2.2.2	nome	9
4.2.2.3	posicao	9
4.2.2.4	proximo	9
4.2.2.5	recursoGasto	9
4.2.2.6	recursoNecessario	9
4.2.2.7	recursoRecebido	9
4.3	gerador Struct Reference	9
4.3.1	Detailed Description	10
4.3.2	Field Documentation	10

4.3.2.1	custo	10
4.3.2.2	nome	10
4.3.2.3	posicao	10
4.3.2.4	proximo	10
4.3.2.5	recursoProduzido	10
4.3.2.6	saida	10
4.3.2.7	taxaProducao	11
4.4	interconexao Struct Reference	11
4.4.1	Detailed Description	11
4.4.2	Field Documentation	12
4.4.2.1	capacidadeMaxima	12
4.4.2.2	chanceFalha	12
4.4.2.3	custoConserto	12
4.4.2.4	entradaAdaptador	12
4.4.2.5	entradaGerador	12
4.4.2.6	entradaInterconexao	12
4.4.2.7	nome	12
4.4.2.8	numeroFalha	13
4.4.2.9	posicaoFinal	13
4.4.2.10	posicaoInicial	13
4.4.2.11	proximo	13
4.4.2.12	proximoEntradaAdaptador	13
4.4.2.13	proximoEntradaCidade	13
4.4.2.14	proximoSaidaAdaptador	13
4.4.2.15	recursoTransportado	13
4.4.2.16	saidaAdaptador	13
4.4.2.17	saidaCidade	13
4.4.2.18	saidaInterconexao	13
4.4.2.19	tagDestino	13
4.4.2.20	tagFalha	14
4.4.2.21	tempoConserto	14
5	File Documentation	15
5.1	app/header/Adaptadores.h File Reference	15
5.1.1	Function Documentation	15
5.1.1.1	adaptadorVazio	15
5.1.1.2	criaListaAdaptador	16
5.1.1.3	defineDistribuicao	16
5.1.1.4	imprimeListaAdaptador	17
5.1.1.5	insereAdaptador	17

5.1.1.6	liberaListaAdaptador	18
5.2	app/header/Cidades.h File Reference	19
5.2.1	Function Documentation	19
5.2.1.1	cidadeVazia	19
5.2.1.2	criaListaCidade	19
5.2.1.3	imprimeListaCidade	20
5.2.1.4	insereCidade	20
5.2.1.5	liberaListaCidade	21
5.2.1.6	recursoGastoTotal	22
5.3	app/header/Geradores.h File Reference	22
5.3.1	Function Documentation	22
5.3.1.1	criaListaGerador	22
5.3.1.2	custoGeradores	23
5.3.1.3	geradorVazio	23
5.3.1.4	imprimeListaGerador	23
5.3.1.5	insereGerador	24
5.3.1.6	liberaListaGerador	25
5.3.1.7	mandarRecursoProduzido	25
5.3.1.8	recursoProduzidoTotal	25
5.4	app/header/Geral.h File Reference	26
5.4.1	Function Documentation	26
5.4.1.1	conecta	26
5.4.1.2	verifica	27
5.5	app/header/Interconexoes.h File Reference	28
5.5.1	Function Documentation	29
5.5.1.1	calculaFalha	29
5.5.1.2	criaListaInterconexao	29
5.5.1.3	imprimeListaInterconexao	29
5.5.1.4	insereInterconexao	30
5.5.1.5	interconexaoVazia	31
5.5.1.6	liberaListaInterconexao	31
5.5.1.7	mandarRecursoTransportado	32
5.5.1.8	tamanhoConexao	32
5.5.1.9	tamanhoTotalConexao	33
5.5.1.10	totalGastoConserto	33
5.6	app/header/Principal.h File Reference	34
5.6.1	Typedef Documentation	34
5.6.1.1	Adaptador	34
5.6.1.2	Cidade	35
5.6.1.3	Destino	35

5.6.1.4	Falha	35
5.6.1.5	Gerador	35
5.6.1.6	Interconexao	35
5.6.1.7	Vazia	36
5.6.1.8	Vazio	36
5.6.2	Enumeration Type Documentation	36
5.6.2.1	destino	36
5.6.2.2	falha	36
5.6.2.3	vazia	37
5.6.2.4	vazio	37
5.7	app/src/Adaptadores.c File Reference	37
5.7.1	Function Documentation	37
5.7.1.1	adaptadorVazio	37
5.7.1.2	criaListaAdaptador	37
5.7.1.3	defineDistribuicao	38
5.7.1.4	imprimeListaAdaptador	38
5.7.1.5	insereAdaptador	38
5.7.1.6	liberaListaAdaptador	39
5.8	app/src/Cidades.c File Reference	40
5.8.1	Function Documentation	40
5.8.1.1	cidadeVazia	40
5.8.1.2	criaListaCidade	40
5.8.1.3	imprimeListaCidade	40
5.8.1.4	insereCidade	40
5.8.1.5	liberaListaCidade	41
5.8.1.6	recursoGastoTotal	42
5.9	app/src/Geradores.c File Reference	42
5.9.1	Function Documentation	42
5.9.1.1	criaListaGerador	42
5.9.1.2	geradorVazio	42
5.9.1.3	imprimeListaGerador	42
5.9.1.4	insereGerador	43
5.9.1.5	liberaListaGerador	43
5.10	app/src/Geral.c File Reference	44
5.10.1	Function Documentation	44
5.10.1.1	conecta	44
5.10.1.2	verifica	45
5.11	app/src/Interconexoes.c File Reference	46
5.11.1	Function Documentation	46
5.11.1.1	calculaFalha	46

5.11.1.2	criaListaInterconexao	46
5.11.1.3	imprimeListaInterconexao	47
5.11.1.4	insereInterconexao	47
5.11.1.5	interconexaoVazia	48
5.11.1.6	liberaListaInterconexao	48
5.11.1.7	mandarRecursoTransportado	48
5.11.1.8	tamanhoConexao	49
5.11.1.9	tamanhoTotalConexao	49
5.12	app/src/Principal.c File Reference	49
5.12.1	Function Documentation	49
5.12.1.1	main	49
5.13	app/test/Adaptadores_unittest.c File Reference	50
5.13.1	Function Documentation	50
5.13.1.1	TEST	50
5.13.1.2	TEST	50
5.13.1.3	TEST	50
5.13.1.4	TEST	50
5.13.1.5	TEST	50
5.14	app/test/Interconexoes_unittest.c File Reference	51
5.15	app/test/main_unittest.c File Reference	51
5.15.1	Function Documentation	51
5.15.1.1	main	51
5.16	README.md File Reference	51

Chapter 1

TrabalhoMP

Trabalho final da disciplina de Métodos de Programação (2/2015) na Universidade de Brasília.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

adaptador	7
cidade	8
gerador	9
interconexao	11

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

app/header/ Adaptadores.h	15
app/header/ Cidades.h	19
app/header/ Geradores.h	22
app/header/ Geral.h	26
app/header/ Interconexoes.h	28
app/header/ Principal.h	34
app/src/ Adaptadores.c	37
app/src/ Cidades.c	40
app/src/ Geradores.c	42
app/src/ Geral.c	44
app/src/ Interconexoes.c	46
app/src/ Principal.c	49
app/test/ Adaptadores_unittest.c	50
app/test/ Interconexoes_unittest.c	51
app/test/ main_unittest.c	51

Chapter 4

Data Structure Documentation

4.1 adaptador Struct Reference

```
#include <Principal.h>
```

Data Fields

- char * [nome](#)
- int [posicao](#) [2]
- int [recursoRecebido](#)
- struct [interconexao](#) * [saidas](#)
- struct [interconexao](#) * [entradas](#)
- int [quantidadeSaidas](#)
- struct [adaptador](#) * [proximo](#)

4.1.1 Detailed Description

-----Adaptadores----- Cabecalho do elemento Adaptador

nome: nome do adaptador

posicao: vetor posicao, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

recursoRecebido: quantidade de recurso recebido por segundo pelo Adaptador

entrada: ponteiro que representa a interconexao de entrada do adaptador

saidas: representa as interconexoes realizadas/apontadas pelos adaptadores

quantidadeSaidas: quantidade de conexoes que o adaptador possui

proximo: representa a proxima Adaptador da lista de Adaptadores

Definition at line 113 of file Principal.h.

4.1.2 Field Documentation

4.1.2.1 struct [interconexao](#)* [entradas](#)

Definition at line 119 of file Principal.h.

4.1.2.2 char* nome

Definition at line 114 of file Principal.h.

4.1.2.3 int posicao[2]

Definition at line 115 of file Principal.h.

4.1.2.4 struct adaptador* proximo

Definition at line 123 of file Principal.h.

4.1.2.5 int quantidadeSaidas

Definition at line 121 of file Principal.h.

4.1.2.6 int recursoRecebido

Definition at line 116 of file Principal.h.

4.1.2.7 struct interconexao* saidas

Definition at line 118 of file Principal.h.

The documentation for this struct was generated from the following file:

- app/header/[Principal.h](#)

4.2 cidade Struct Reference

```
#include <Principal.h>
```

Data Fields

- char * [nome](#)
- int [posicao](#) [2]
- int [recursoNecessario](#)
- int [recursoRecebido](#)
- int [recursoGasto](#)
- struct [cidade](#) * [proximo](#)
- struct [interconexao](#) * [entradas](#)

4.2.1 Detailed Description

-----Cidades----- Cabecalho do elemento Cidade

nome: nome da cidade

posicao: vetor posicao, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

recursoNecessario: quantidade de recurso que a cidade precisa por segundo

recursoRecebido: quantidade de recurso recebido por segundo pela Cidade

recursoGasto: quantidade de recurso que a cidade usou

proximo: representa a proxima cidade da rede de cidades

entradas: ponteiro que representa a interconexao de entrada da cidade

Definition at line 72 of file Principal.h.

4.2.2 Field Documentation

4.2.2.1 struct interconexao* entradas

Definition at line 81 of file Principal.h.

4.2.2.2 char* nome

Definition at line 73 of file Principal.h.

4.2.2.3 int posicao[2]

Definition at line 74 of file Principal.h.

4.2.2.4 struct cidade* proximo

Definition at line 80 of file Principal.h.

4.2.2.5 int recursoGasto

Definition at line 78 of file Principal.h.

4.2.2.6 int recursoNecessario

Definition at line 76 of file Principal.h.

4.2.2.7 int recursoRecebido

Definition at line 77 of file Principal.h.

The documentation for this struct was generated from the following file:

- [app/header/Principal.h](#)

4.3 gerador Struct Reference

```
#include <Principal.h>
```

Data Fields

- char * [nome](#)
- int [posicao](#) [2]

- int [taxaProducao](#)
- int [recursoProduzido](#)
- int [custo](#)
- struct [gerador](#) * [proximo](#)
- struct [interconexao](#) * [saida](#)

4.3.1 Detailed Description

-----Geradores----- Cabecalho do elemento Gerador

nome: nome do gerador

posicao: vetor posicao, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

taxaProducao: quantidade de recurso que a cidade precisa por segundo

recursoProduzido: quantidade total de recurso produzido pelo gerador

custo: custo de geração por segundo

proximo: representa a proxima cidade da rede de cidades

saida: representa as interconexoes realizadas/apontadas pelos adaptadores

Definition at line 263 of file Principal.h.

4.3.2 Field Documentation

4.3.2.1 int custo

Definition at line 269 of file Principal.h.

4.3.2.2 char* nome

Definition at line 264 of file Principal.h.

4.3.2.3 int posicao[2]

Definition at line 265 of file Principal.h.

4.3.2.4 struct [gerador](#)* [proximo](#)

Definition at line 271 of file Principal.h.

4.3.2.5 int [recursoProduzido](#)

Definition at line 268 of file Principal.h.

4.3.2.6 struct [interconexao](#)* [saida](#)

Definition at line 272 of file Principal.h.

4.3.2.7 int taxaProducao

Definition at line 267 of file Principal.h.

The documentation for this struct was generated from the following file:

- app/header/[Principal.h](#)

4.4 interconexao Struct Reference

```
#include <Principal.h>
```

Data Fields

- char * [nome](#)
- int [posicaoInicial](#) [2]
- int [posicaoFinal](#) [2]
- [Destino](#) tagDestino
- float [chanceFalha](#)
- int [tempoConserto](#)
- int [custoConserto](#)
- int [numeroFalha](#)
- [Falha](#) tagFalha
- int [capacidadeMaxima](#)
- int [recursoTransportado](#)
- struct [interconexao](#) * [proximo](#)
- struct [adaptador](#) * [entradaAdaptador](#)
- struct [adaptador](#) * [saidaAdaptador](#)
- struct [interconexao](#) * [entradaInterconexao](#)
- struct [interconexao](#) * [saidaInterconexao](#)
- struct [interconexao](#) * [proximoEntradaAdaptador](#)
- struct [interconexao](#) * [proximoSaidaAdaptador](#)
- struct [interconexao](#) * [proximoEntradaCidade](#)
- struct [gerador](#) * [entradaGerador](#)
- struct [cidade](#) * [saidaCidade](#)

4.4.1 Detailed Description

-----Interconexoes----- Cabecalho do elemento Inerconexao

nome: nome da interconexao

posicaoInicial: vetor posicaoInicial, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

posicaoFinal: vetor posicaoFinal, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

tagDestino: tag para indentificar qual é a ligação final de cada conexao, seja cidade ou adaptador

chanceFalha: chance de falha por segundo

tempoConcerto: tempo de concerto em caso de falha em segundos

custoConcerto: custo do concerto em segundos

numeroFalha: total de falhas

tagFalha: indica se houve falha

capacidadeMaxima: capacidade maxima da interconexao

recursoTranstornado: quantidade de recuso que esta sendo transportado pela conexao no turno

proximo: representa a proxima interconexao na lista de interconexoes

entradaAdaptador: aponta, caso a entrada seja um adaptador, para o adaptador cuja saída eh esta interconexao

saidaAdaptador: aponta, caso a saida seja um adaptador, para o adaptador cuja entrada eh esta interconexao

entradaInterconexao: aponta, caso a entrada seja interconexao, para a interconexao cuja saida eh esta interconexao

saidaInterconexao: aponta, caso a saida seja interconexao, para a interconexao cuja entrada eh esta interconexao

proximoEntradaAdaptador: ponteiro para a proxima entrada do adaptador relacionado a interconexao

proximoSaidaAdaptador: ponteiro para a proxima saida do adaptador relacionado a interconexao

proximoEntradaCidade: ponteiro para a proxima cidade de destino

entradaGerador: aponta, caso a entrada seja um gerador, para o gerador cuja saida eh esta interconexao

saidaCidade: aponta, caso a saida seja uma cidade, para a cidade cuja entrada eh esta interconexao

Definition at line 199 of file Principal.h.

4.4.2 Field Documentation

4.4.2.1 int capacidadeMaxima

Definition at line 212 of file Principal.h.

4.4.2.2 float chanceFalha

Definition at line 205 of file Principal.h.

4.4.2.3 int custoConserto

Definition at line 207 of file Principal.h.

4.4.2.4 struct adaptador* entradaAdaptador

Definition at line 217 of file Principal.h.

4.4.2.5 struct gerador* entradaGerador

Definition at line 228 of file Principal.h.

4.4.2.6 struct interconexao* entradaInterconexao

Definition at line 220 of file Principal.h.

4.4.2.7 char* nome

Definition at line 200 of file Principal.h.

4.4.2.8 int numeroFalha

Definition at line 209 of file Principal.h.

4.4.2.9 int posicaoFinal[2]

Definition at line 202 of file Principal.h.

4.4.2.10 int posicaoInicial[2]

Definition at line 201 of file Principal.h.

4.4.2.11 struct interconexao* proximo

Definition at line 215 of file Principal.h.

4.4.2.12 struct interconexao* proximoEntradaAdaptador

Definition at line 223 of file Principal.h.

4.4.2.13 struct interconexao* proximoEntradaCidade

Definition at line 226 of file Principal.h.

4.4.2.14 struct interconexao* proximoSaidaAdaptador

Definition at line 224 of file Principal.h.

4.4.2.15 int recursoTransportado

Definition at line 213 of file Principal.h.

4.4.2.16 struct adaptador* saidaAdaptador

Definition at line 218 of file Principal.h.

4.4.2.17 struct cidade* saidaCidade

Definition at line 230 of file Principal.h.

4.4.2.18 struct interconexao* saidaInterconexao

Definition at line 221 of file Principal.h.

4.4.2.19 Destino tagDestino

Definition at line 203 of file Principal.h.

4.4.2.20 Falha tagFalha

Definition at line 210 of file Principal.h.

4.4.2.21 int tempoConcerto

Definition at line 206 of file Principal.h.

The documentation for this struct was generated from the following file:

- app/header/[Principal.h](#)

Chapter 5

File Documentation

5.1 app/header/Adaptadores.h File Reference

```
#include "Principal.h"
```

Functions

- [Adaptador *](#) [criaListaAdaptador](#) ()
- [Vazio](#) [adaptadorVazio](#) ([Adaptador *](#))
- [Adaptador *](#) [insereAdaptador](#) (char *, [Adaptador *](#))
- void [imprimeListaAdaptador](#) ([Adaptador *](#))
- void [liberaListaAdaptador](#) ([Adaptador *](#))
- void [defineDistribuicao](#) ([Adaptador *](#))

5.1.1 Function Documentation

5.1.1.1 [Vazio](#) [adaptadorVazio](#) ([Adaptador *](#) *listaAlvo*)

Funcao: [adaptadorVazio](#)

Verifica se a lista de adaptadores esta vazia

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de adaptadores
------------------	--

Returns

variavel do tipo [Vazio](#), indicando se a lista esta vazia

Assertiva de entrada: estrutura do tipo [Adaptador](#)

Assertiva de saida: condicao do [Adaptador](#) sendo [vazio](#) ou [nao vazio](#)

Funcao: [adaptadorVazio](#)

AssertivaSaida: [VAZIO](#) || [NAO_VAZIO](#); AE: [listaAlvo](#) eh vazia

AE: [listaAlvo](#) nao eh vazia

AS: o retorno deve ser uma variavel do tipo [Vazia](#)

Definition at line 22 of file [Adaptadores.c](#).

5.1.1.2 Adaptador* criaListaAdaptador ()

-----Funcoes Basicas----- Funcao: criaListaAdaptador

Inicia um ponteiro que sera para Adaptador

AssertivaSaida: NULL;

-----Funcoes Basicas----- Funcao: criaListaAdaptador

AssertivaSaida: NULL;

Definition at line 12 of file Adaptadores.c.

5.1.1.3 void defineDistribuicao (Adaptador * listaAlvo)

-----Funcoes de Calculo-----

Define como sera a distribuicao entre as conexoes que esta ligada a cada adaptador e manda a quantidade que sera passada para as correspondentes conexoes

$\text{interconexao} \rightarrow \text{recursoTransportado} = \text{interconexao} \rightarrow \text{capacidadeMax} * \text{adaptador} \rightarrow \text{recursoRecebido} / \text{soma de todas as capacidadeMax}$

Essa funcao espera que a lista de interconexoes ja foi previamente estabelecida //TODO retirar essa dependencia

Parameters

<i>listaAlvo</i>	ponteiro para o inico da lista de adaptadores;
------------------	--

-----Funcoes de Calculo-----

Funcao: defineDistribuicao

AssertivaEntrada: $\text{adaptadorVazio}(\text{listaAlvo}) == \text{NAO_VAZIO}$; para cada adaptador da lista: $\text{adaptador} \rightarrow \text{saidas}[i] \neq \text{null}$; < conexoes do adaptador

< quantidade de recurso transportado

Assertiva estrutural: aux é a lista nao-nula de adaptadores

AE: aux nao chegou ao fim da lista de adaptadores

Assertiva estrutural: somatorio eh a soma das capacidades maximas do adaptador corrente

AE: o adaptador corrente possui saidas

Assertiva estrutural: conexao eh a lista de saidas do adaptador corrente

Comentarios de argumentacao

Enquanto percorre a lista de conexoes que saem do adaptador somam-se a capacidade maxima de todas as conexoes

AE: a lista de saidas do adaptador nao chegou ao fim

AS: a lista de saidas do adaptador chegou ao fim

Assertiva estrutural: conexao eh a lista de saidas do adaptador corrente

AE: a lista de saidas do adaptador nao chegou ao fim

AE: a saida corrente do adaptador nao possui falha

Assertiva estrutural: $\text{recursoTransportado}$ é a quantidade de recurso que cada conexao vai transportar no turno

Comentarios de argumentacao

Se a a capacidade maxima da saida corrente do adaptador for maior ou igual ao $\text{recursoTransportado}$ Entao o $\text{recursoTransportado}$ da saida corrente do adaptador recebe o $\text{recursoTransportado}$ Senao recebe a capacidade-Maxima da saida corrente do adaptador FimSe

AS: a saída corrente do adaptador possui falha

AS: a lista de saídas do adaptador chegou ao fim

AS: a lista de adaptadores chegou ao fim

Definition at line 211 of file Adaptadores.c.

5.1.1.4 void imprimeListaAdaptador (Adaptador * listaAlvo)

Funcao: imprimeListaAdaptador

Imprime de todas as celulas de lista de adaptador as respectivas caracteristicas: nome posicao x posicao y recurso recebido quantidade de saidas

Parameters

<i>listaAlvo</i>	lista que sera impressa
------------------	-------------------------

AssertivaEntrada: A lista nao deve ser vazia

AssertivaSaida: Se a lista de adaptadores a ser imprimida nao eh vazia Entao ela eh imprimida Senao a lista de adaptadores nao eh imprimida FimSe

Funcao: imprimeListaAdaptador

AssertivaEntrada: adaptadorVazio(listaAlvo) == NAO_VAZIO; Asseriva estrutural: aux é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Imprime os atributos do adaptador corrente

AS: listaAlvo chegou ao fim

Definition at line 149 of file Adaptadores.c.

5.1.1.5 Adaptador* insereAdaptador (char * registro, Adaptador * listaAlvo)

Insere uma nova celula de adaptador na lista dos adaptadores

Parameters

<i>registro</i>	String que sera decodificada e inserida
<i>listaAlvo</i>	lista a qual essa nova celula sera inserida

Returns

Adaptador novo ponteiro de referencia para o inicio da lista

Assertiva de entrada: registro - eh um vetor contendo o conteudo do txt, deve ser diferente de NULL

Assertiva de saida: A lista recebida pela funcao, deve ser o proximo adaptador apontado pela lista retornada

Funcao: insereAdaptador

AssertivaEntrada: registro != NULL; < Alocacao de um novo adaptador

< Alocacao de um vetor do tamanho do registro

< Variaveis de auxilio

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Comecando de registro[2], enquanto registro[i] for um caracter irrelevante, soma-se 1 a variavel de auxilio i

AS: a posicao corrente do registro possui um caracter irrelevante

Asseriva estrutural: o nome do novo adaptador possui tamanho i-1

AE: o valor da variavel auxiliar j deve ser menor ou igual ao numero total de atributos lidos do adaptador

AE: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AE: o valor de j eh igual a 0

Comentarios de argumentacao

A (posicao corrente-k) recebe um caracter finalizador

AS: o valor de j eh maior que 0

AE: o valor de j eh maior que 0

Comentarios de argumentacao

A (posicao corrente-k) do vetor numChar recebe um caracter finalizador

Comentarios de argumentacao

De acordo com o valor da variavel auxiliar j, armazena-se o vetor numChar no seu respectivo atributo lido

AS: o valor de j eh maior que 4

AS: a posicao corrente do registro possui um caracter relevante

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Se o valor da variavel de auxilio j for 0 Entao armazena-se a posicao corrente do registro na (posicao corrente-k) do nome do adaptador Senao armazena-se a posicao corrente do registro na (posicao corrente-k) do vetor numChar FimSe

AS: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AS: o valor da variavel auxiliar j ultrapassou o numero total de atributos lidos da adaptador

Comentarios de argumentacao

Os atributos nao lidos do adaptador inserido recebem o valor nulo, e o proximo adaptador da lista que contem o novo adaptador inserido na cabeca recebe a lista de adaptadores atual

Definition at line 38 of file Adaptadores.c.

5.1.1.6 void liberaListaAdaptador (Adaptador * listaAlvo)

Libera o espaco de memoria alocado para a lista de adaptadores

Parameters

<i>listaAlvo</i>	lista a qual sera desalocada
------------------	------------------------------

AssertivaEntrada: A lista nao deve ser vazia

AssertivaSaida: A lista deve estar vazia

Funcao: liberaListaAdaptador

AssertivaEntrada: adaptadorVazio(listaAlvo) == NAO_VAZIO;

AssertivaSaida: adaptadorVazio(listaAlvo) == VAZIO; Asseriva estrutural: aux1 é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Liberam os atributos alocados dinamicamente do elemento Adaptador corrente

AS: listaAlvo chegou ao fim

Definition at line 179 of file Adaptadores.c.

5.2 app/header/Cidades.h File Reference

```
#include "Adaptadores.h"
```

Functions

- [Cidade *](#) [criaListaCidade](#) ()
- [Vazia cidadeVazia](#) ([Cidade *](#))
- [Cidade *](#) [insereCidade](#) (char *, [Cidade *](#))
- void [imprimeListaCidade](#) ([Cidade *](#))
- void [liberaListaCidade](#) ([Cidade *](#))
- int [recursoGastoTotal](#) ([Cidade *](#))

5.2.1 Function Documentation

5.2.1.1 Vazia cidadeVazia ([Cidade *](#) *listaAlvo*)

Funcao: cidadeVazia

Verifica se a lista de cidades esta vazia

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de cidades
------------------	--

Returns

variavel do tipo Vazia, indicando se a lista esta vazia

Assertiva de entrada: estrutura do tipo Cidade

Assertiva de saida: condicao da Cidade sendo vazia ou nao vazia

Funcao: cidadeVazia

AssertivaSaida: VAZIA || NAO_VAZIA; AE: listaAlvo eh vazia

AE: listaAlvo nao eh vazia

AS: o retorno deve ser uma variavel do tipo Vazia

Definition at line 22 of file Cidades.c.

5.2.1.2 [Cidade*](#) [criaListaCidade](#) ()

-----Funcoes Basicas----- Funcao: [criaListaCidade](#)

Inicia um ponteiro que sera para Cidade

Returns

null

Assertiva de saida: estrutura do tipo Cidade nula

-----Funcoes Basicas----- Funcao: [criaListaCidade](#)

AssertivaSaida: NULL;

Definition at line 12 of file Cidades.c.

5.2.1.3 void imprimeListaCidade (Cidade * listaAlvo)

Funcao: imprimeListaCidade

Imprime de todas as celulas de lista de cidade as respectivas caracteristicas: nome posicao x posicao y recurso necessario recurso recebido recurso gasto

Parameters

<i>listaAlvo</i>	lista que sera impressa
------------------	-------------------------

AssertivaEntrada: A lista nao deve ser vazia

AssertivaSaida: Se a lista de cidades a ser imprimida nao eh vazia Entao ela eh imprimida Senao a lista de cidades nao eh imprimida FimSe

Funcao: imprimeListaCidade

AssertivaEntrada: cidadeVazia(listaAlvo) == NAO_VAZIA; Asseriva estrutural: aux é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Imprime os atributos da cidade corrente

AS: listaAlvo chegou ao fim

Definition at line 149 of file Cidades.c.

5.2.1.4 Cidade* insereCidade (char * registro, Cidade * listaAlvo)

Funcao: insereCidade

Inserir uma nova celula na lista de cidades a insercao se da pelo inicio da lista e é retornadado o novo ponterio para lista.

Parameters

<i>registro</i>	string que sera lida do arquivo representando Cidade
<i>listaAlvo</i>	lista de cidades onde a nova celula sera inserida

Returns

novos ponteiros para a o inicio da lista de cidades

Assertiva de entrada: registro - eh um vetor contendo o conteudo do txt, deve ser diferente de NULL

Assertiva de saida: A lista recebida pela funcao, deve ser a proxima cidade apontada pela lista retornada

Funcao: insereCidade

AssertivaEntrada: registro != NULL; < Alocacao da nova cidade

< Alocacao de um vetor do tamanho do registro

< Variaveis de auxilio

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Comecando de registro[2], enquanto registro[i] for um caracter irrelevante, soma-se 1 a variavel de auxilio i

AS: a posicao corrente do registro possui um caracter irrelevante

Asseriva estrutural: o nome da nova cidade possui tamanho i-1

AE: o valor da variavel auxiliar j deve ser menor ou igual ao numero total de atributos lidos da cidade

AE: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AE: o valor de j eh igual a 0

Comentarios de argumentacao

A (posicao corrente-k) recebe um caracter finalizador

AS: o valor de j eh maior que 0

AE: o valor de j eh maior que 0

Comentarios de argumentacao

A (posicao corrente-k) do vetor numChar recebe um caracter finalizador

Comentarios de argumentacao

De acordo com o valor da variavel auxiliar j, armazena-se o vetor numChar no seu respectivo atributo lido

AS: o valor de j eh maior que 4

AS: a posicao corrente do registro possui um caracter relevante

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Se o valor da variavel de auxilio j for 0 Entao armazena-se a posicao corrente do registro na (posicao corrente-k) do nome da cidade Senao armazena-se a posicao corrente do registro na (posicao corrente-k) do vetor numChar FimSe

AS: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AS: o valor da variavel auxiliar j ultrapassou o numero total de atributos lidos da cidade

Comentarios de argumentacao

Os atributos nao lidos da cidade inserida recebem o valor nulo, e a proxima cidade da lista que contem a nova cidade inserido na cabeca recebe a lista de cidades atual

Definition at line 38 of file Cidades.c.

5.2.1.5 void liberaListaCidade (Cidade * listaAlvo)

Funcao: liberaListaCidade

Desaloca a memoria reservada para toda celula pertecente a lista de cidades

Parameters

<i>listaAlvo</i>	lista a ser desalocada
------------------	------------------------

AssertivaEntrada: A lista nao deve ser vazia

AssertivaSaida: A lista deve estar vazia

Funcao: liberaListaCidade

AssertivaEntrada: cidadeVazia(listaAlvo) == NAO_VAZIA;

AssertivaSaida: cidadeVazia(listaAlvo) == VAZIA; Asseriva estrutural: aux1 é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Liberam os atributos alocados dinamicamente do elemento Cidade corrente

AS: listaAlvo chegou ao fim

Definition at line 179 of file Cidades.c.

5.2.1.6 int recursoGastoTotal (Cidade * cidade)

-----Funcoes de Calculo----- Funcao: recursoGastoTotal

Resultado da soma de todos recursos gasto pelas cidades

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de cidades
------------------	--

Returns

total de recursos gasto pelas cidades

Assertiva de entrada:

Assertiva de saida:

-----Funcoes de Calculo----- Funcao: recursoGastoTotal

Definition at line 207 of file Cidades.c.

5.3 app/header/Geradores.h File Reference

```
#include "Cidades.h"
```

Functions

- Gerador * [criaListaGerador](#) ()
- Vazio [geradorVazio](#) (Gerador *)
- Gerador * [insereGerador](#) (char *, Gerador *)
- void [imprimeListaGerador](#) (Gerador *)
- void [liberaListaGerador](#) (Gerador *)
- int [recursoProduzidoTotal](#) (Gerador *)
- int [custoGeradores](#) (Gerador *)
- void [mandarRecursoProduzido](#) (Gerador *)

5.3.1 Function Documentation

5.3.1.1 Gerador* criaListaGerador ()

-----Funcoes Basicas----- Funcao: criaListaGerador

Inicia um ponteiro que sera para Gerador

Returns

null

Assertiva de saida: estrutura do tipo Gerador nula

Funcao: criaListaGerador

AssertivaSaida: NULL;

Definition at line 10 of file Geradores.c.

5.3.1.2 int custoGeradores (Gerador *)

Funcao: custoGeradores

Calcula a soma de custo por segundo de todos os geradores da lista

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de geradores
------------------	--

Returns

Soma do custo por segundo de todos os geradores

5.3.1.3 Vazio geradorVazio (Gerador * listaAlvo)

Funcao: geradorVazio

Verifica se a lista de geradores esta vazia

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de geradores
------------------	--

Returns

variavel do tipo Vazio, indicando se a lista esta vazia

Assertiva de entrada: estrutura do tipo Gerador

Assertiva de saida: condicao do Gerador sendo vazio ou nao vazio

Funcao: criaListaCidade

AssertivaSaida: VAZIO || NAO_VAZIO; AE: listaAlvo eh vazia

AE: listaAlvo nao eh vazia

AS: o retorno deve ser uma variavel do tipo Vazia

Definition at line 20 of file Geradores.c.

5.3.1.4 void imprimeListaGerador (Gerador * listaAlvo)

Funcao: imprimeListaGerador

Imprime de todas as celulas de lista de gerador as respectivas caracteristicas: nome posicao x posicao y taxa de producao recurso produzido custo

Parameters

<i>listaAlvo</i>	lista que sera impressa
------------------	-------------------------

AssertivaEntrada: A lista nao deve ser vazia

AssertivaSaida: Se a lista de geradores a ser imprimida nao eh vazia Entao ela eh imprimida Senao a lista de geradores nao eh imprimida FimSe

Funcao: imprimeListaGerador

AssertivaEntrada: geradorVazio(listaAlvo) == NAO_VAZIO; Asseriva estrutural: aux é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Imprime os atributos do gerador corrente

AS: listaAlvo chegou ao fim

Definition at line 147 of file Geradores.c.

5.3.1.5 Gerador* insereGerador (char * *registro*, Gerador * *listaAlvo*)

Funcao: insereGerador

Inserir uma nova celula na lista de geradores a insercao se da pelo inicio da lista e é retornadado o novo ponterio para lista.

Parameters

<i>registro</i>	string que sera lida do arquivo representando Gerador
<i>listaAlvo</i>	lista de cidade a qual a nova celula sera inserida

Returns

novo pontero para a o inicio da lista de geradores

Assertiva de entrada: registro - eh um vetor contendo o conteudo do txt, deve ser diferente de NULL

Assertiva de saida: A lista recebida pela funcao, deve ser o proximo gerador apontado pela lista retornada

Funcao: insereGerador

AssertivaEntrada: registro != NULL; < Alocacao de um novo gerador

< Alocacao de um vetor do tamanho do registro

< Variaveis de auxilio

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Comecando de registro[2], enquanto registro[i] for um caracter irrelevante, soma-se 1 a variavel de auxilio i

AS: a posicao corrente do registro possui um caracter irrelevante

Asseriva estrutural: o nome do novo adaptador possui tamanho i-1

AE: o valor da variavel auxiliar j deve ser menor ou igual ao numero total de atributos lidos do gerador

AE: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AE: o valor de j eh igual a 0

Comentarios de argumentacao

A (posicao corrente-k) recebe um caracter finalizador

AS: o valor de j eh maior que 0

AE: o valor de j eh maior que 0

Comentarios de argumentacao

A (posicao corrente-k) do vetor numChar recebe um caracter finalizador

Comentarios de argumentacao

De acordo com o valor da variavel auxiliar j, armazena-se o vetor numChar no seu respectivo atributo lido

AS: o valor de j eh maior que 5

AS: a posicao corrente do registro possui um caracter relevante

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Se o valor da variavel de auxilio j for 0 Entao armazena-se a posicao corrente do registro na (posicao corrente-k) do nome do gerador Senao armazena-se a posicao corrente do registro na (posicao corrente-k) do vetor numChar FimSe

AS: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AS: o valor da variavel auxiliar j ultrapassou o numero total de atributos lidos do gerador

Comentarios de argumentacao

Os atributos nao lidos do gerador inserido recebem o valor nulo, e o proximo gerador da lista que contem o novo gerador inserido na cabeca recebe a lista de geradores atual

Definition at line 36 of file Geradores.c.

5.3.1.6 void liberaListaGerador (Gerador * listaAlvo)

Funcao: liberaListaGerador

Desaloca a memoria reservada para toda celula pertencente a lista de geradores

Parameters

<i>listaAlvo</i>	lista a ser desalocada
------------------	------------------------

AssertivaEntrada: A lista nao deve ser vazia

AssertivaSaida: A lista deve estar vazia

Funcao: liberaListaGerador

AssertivaEntrada: geradorVazio(listaAlvo) == NAO_VAZIO;

AssertivaSaida: geradorVazio(listaAlvo) == VAZIO; Asseriva estrutural: aux1 é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Liberam os atributos alocados dinamicamente do elemento Gerador corrente

AS: listaAlvo chegou ao fim

Definition at line 177 of file Geradores.c.

5.3.1.7 void mandarRecursoProduzido (Gerador *)

Funcao: mandarRecursoProduzido

Altera os nos de conexão do grafo com o recurso enviado

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de geradores
------------------	--

5.3.1.8 int recursoProduzidoTotal (Gerador *)

-----Funcoes de Calculo----- Funcao: recursoProduzidoTotal

Calcula a soma do total de recursos produzido pelas celualas.

Parameters

<i>listaAlvo</i>	inicio da lista de geradores;
------------------	-------------------------------

Returns

soma de todo o recurso produzido pelos geradores

5.4 app/header/Geral.h File Reference

```
#include "Interconexoes.h"
```

Functions

- void [conecta](#) ([Cidade](#) *, [Gerador](#) *, [Interconexao](#) *, [Adaptador](#) *)
- void [verifica](#) ([Cidade](#) *, [Gerador](#) *, [Interconexao](#) *, [Adaptador](#) *)

5.4.1 Function Documentation

5.4.1.1 void conecta (Cidade * cidades, Gerador * geradores, Interconexao * interconexoes, Adaptador * adaptadores)

-----Funcoes Gerais----- Funcao: conecta

Realiza a conexao entre as listas

Parameters

<i>cidade</i>	lista de cidades
<i>gerador</i>	lista de geradores
<i>interconexao</i>	lista de interconexoes
<i>adaptador</i>	lista de adaptadores

Assertiva de entrada:

Assertiva de saida:

Funcao: conecta

AssertivaEntrada:

AssertivaSaida: < Ponteiro auxiliar para cidade

< Ponteiro auxiliar para gerador

< Ponteiro auxiliar para interconexao

< Ponteiro auxiliar para interconexao

< Ponteiro auxiliar para adaptador

AE: o auxI1 ainda nao chegou ao fim da lista de interconexoes

AE: o auxA ainda nao chegou ao fim da lista de adaptadores

AE: a posicao inicial da interconexao apontada por auxI1 coincide com a posicao do adaptador apontado por auxA

Comentarios de argumentacao

insere-se o adaptador como entrada da interconexao insere-se a interconexao na lista de saidas do adaptador

AE: a posicao final da interconexao apontada por auxI1 coincide com a posicao do adaptador apontado por auxA

Comentarios de argumentacao

insere-se o adaptador como saida da interconexao insere-se a interconexao na lista de entradas do adaptador

AS: o auxA chegou ao fim da lista de adaptadores

AE: o auxG ainda nao chegou ao fim da lista de geradores

AE: a posicao inicial da interconexao apontada por auxI1 coincide com a posicao do gerador apontado por auxG

Comentarios de argumentacao

insere-se a interconexao como saida do gerador insere-se o gerador como entrada da interconexao

AS: o auxG chegou ao fim da lista de geradores

AE: o auxC ainda nao chegou ao fim da lista de cidades

AE: a posicao final da interconexao apontada por auxI1 coincide com a posicao da cidade apontado por auxC

Comentarios de argumentacao

insere-se a cidade como saida da lista de interconexao insere-se a interconexao na lista de entradas da cidade

AS: o auxG chegou ao fim da lista de geradores

AE: o auxI2 ainda nao chegou ao fim da lista de interconexoes

AE: a posicao inicial da interconexao apontada por auxI1 coincide com a posicao final da interconexao apontada por auxI2

Comentarios de argumentacao

insere-se a interconexao auxI1 como saida da interconexao auxI2 insere-se a interconexao auxI2 como entrada da interconexao auxI1

AE: a posicao final da interconexao apontada por auxI1 coincide com a posicao inicial da interconexao apontada por auxI2

Comentarios de argumentacao

insere-se a interconexao auxI2 como saida da interconexao auxI1 insere-se a interconexao auxI1 como entrada da interconexao auxI2

AS: o auxI2 chegou ao fim da lista de interconexoes

AS: o auxI1 chegou ao fim da lista de interconexoes

Definition at line 11 of file Geral.c.

5.4.1.2 void verifica (Cidade * cidades, Gerador * geradores, Interconexao * interconexoes, Adaptador * adaptadores)

Funcao: verifica

Realiza a verificacao das conexoes, checando se as listas estao vazias

Parameters

<i>cidade</i>	lista de cidades
<i>gerador</i>	lista de geradores
<i>interconexao</i>	lista de interconexoes
<i>adaptador</i>	lista de adaptadores

Assertiva de entrada:

Assertiva de saida:

Funcao: verifica

AssertivaEntrada:

AssertivaSaida: < Ponteiro auxiliar para cidade

< Ponteiro auxiliar para gerador

< Ponteiro auxiliar para interconexao

< Ponteiro auxiliar para adaptador

AE: a lista de cidades nao eh vazia

AE: o auxC ainda nao chegou ao fim da lista de cidades

AE: a cidade corrente nao possui entradas

AS: o auxC chegou ao fim da lista de cidades

AE: a lista de cidades eh vazia

AE: a lista de geradores nao eh vazia

AE: o auxG ainda nao chegou ao fim da lista de geradores

AE: o gerador corrente nao possui saidas

AS: o auxG chegou ao fim da lista de geradores

AE: a lista de geradores eh vazia

AE: a lista de interconexoes nao eh vazia

AE: o auxI ainda nao chegou ao fim da lista de interconexoes

AE: a interconexao nao possui entradas em geradores, adaptadores e interconexoes

AE: a interconexao nao possui saidas em geradores, adaptadores e interconexoes

AS: o auxI chegou ao fim da lista de interconexoes

AE: a lista de interconexoes eh vazia

AE: a lista de adaptadores nao eh vazia

AE: o auxA ainda nao chegou ao fim da lista de adaptadores

AE: o adaptador corrente nao possui entradas

AE: o adaptador corrente nao possui saidas

AS: o auxA chegou ao fim da lista de adaptadores

AE: a lista de adaptadores eh vazia

Definition at line 132 of file Geral.c.

5.5 app/header/Interconexoes.h File Reference

```
#include "Geradores.h"
```

Functions

- [Interconexao *](#) [criaListaInterconexao](#) ()
- [Vazia](#) [interconexaoVazia](#) ([Interconexao *](#))
- [Interconexao *](#) [inserirInterconexao](#) ([char *](#), [Interconexao *](#))
- void [imprimeListaInterconexao](#) ([Interconexao *](#))
- void [liberaListaInterconexao](#) ([Interconexao *](#))
- float [tamanhoConexao](#) ([Interconexao *](#))
- float [tamanhoTotalConexao](#) ([Interconexao *](#))
- int [totalGastoConserto](#) ([Interconexao *](#))
- [Falha](#) [calculaFalha](#) ()
- void [mandarRecursoTransportado](#) ([Interconexao *](#))

5.5.1 Function Documentation

5.5.1.1 Falha calculaFalha ()

Funcao: calculaFalha

Calcula a possibilidade de falha

Returns

variavel do tipo Falha, indicando se houve falha

Assertiva de saida: Se a chance de falha for maior que 0 e maior que um numero aleatorio Entao ocorre um falha Senao nao ocorre uma falha FimSe

Funcao: calculaFalha

AssertivaSaida: FALHA || SEM_FALHA; Asseriva estrutural: num eh um numero gerado aleatoriamente

AE: chance deve ser maior que 0 e maior ou igual a num

AE: chance deve ser menor que 0 ou menor que num

AS: o retorno deve ser uma variavel do tipo Falha

Definition at line 322 of file Interconexoes.c.

5.5.1.2 Interconexao* criaListaInterconexao ()

-----Funcoes Basicas----- Funcao: criaListaInterconexao

Inicia um ponteiro que sera para Interconexao

AssertivaSaida: NULL;

-----Funcoes Basicas----- Funcao: criaListaInterconexao

AssertivaSaida: NULL;

Definition at line 12 of file Interconexoes.c.

5.5.1.3 void imprimeListaInterconexao (Interconexao * listaAlvo)

Funcao: imprimeListaInterconexao

Imprime de todas as celulas de lista de interconexoes as respectivas caracteristicas: nome posicao inicial x posicao inicial y posicao final x posicao final y tag de destino chance de falha tempo de concerto custo de concerto numero de falhas tag de falha capacidade maxima recurso transportado

Parameters

<i>listaAlvo</i>	lista que sera impressa
------------------	-------------------------

AssertivaEntrada: A lista nao deve ser vazia

AssertivaSaida: Se a lista de interconexoes a ser imprimida nao eh vazia Entao ela eh imprimida Senao a lista de interconexoes nao eh imprimida FimSe

Funcao: imprimeListaInterconexao

AssertivaEntrada: interconexaoVazia(listaAlvo) == NAO_VAZIA; Asseriva estrutural: aux é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Imprime os atributos da interconexao corrente

AS: listaAlvo chegou ao fim

Definition at line 189 of file Interconexoes.c.

5.5.1.4 Interconexao* insereInterconexao (char * *registro*, Interconexao * *listaAlvo*)

Funcao: insereInterconexao

Inserir uma nova celula na lista de interconexoes a insercao se da pelo inicio da lista e é retornadado o novo ponterio para lista.

Parameters

<i>registro</i>	string que sera lida do arquivo representando Inteconexao
<i>listaAlvo</i>	lista de interconexoes onde a nova celula sera inserida

Returns

novo ponterio para a o inicio da lista de interconexoes

Assertiva de entrada: registro - eh um vetor contendo o conteudo do txt, deve ser diferente de NULL

Assertiva de saida: A lista recebida pela funcao, deve ser a proxima interconexao apontada pela lista retornada

Funcao: insereInterconexao

AssertivaEntrada: registro != NULL; < Alocacao da nova Interconexao

< Variavel auxiliar para percorrer a lista e inserir o elemento no final

< Alocacao de um vetor do tamanho do registro

< Variaveis de auxilio

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Comecando de registro[2], enquanto registro[i] for um caracter irrelevante, soma-se 1 a variavel de auxilio i

AS: a posicao corrente do registro possui um caracter irrelevante

Asseriva estrutural: o nome da nova interconexao possui tamanho i-1

AE: o valor da variavel auxiliar j deve ser menor ou igual ao numero total de atributos lidos da interconexao

AE: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AE: o valor de j eh igual a 0

Comentarios de argumentacao

A (posicao corrente-k) recebe um caracter finalizador

AS: o valor de j eh maior que 0

AE: o valor de j eh maior que 0

Comentarios de argumentacao

A (posicao corrente-k) do vetor numChar recebe um caracter finalizador

Comentarios de argumentacao

De acordo com o valor da variavel auxiliar j, armazena-se o vetor numChar no seu respectivo atributo lido

AS: o valor de j eh maior que 8

AS: a posicao corrente do registro possui um caracter relevante

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Se o valor da variavel de auxilio j for 0 Entao armazena-se a posicao corrente do registro na (posicao corrente-k) do nome da interconexao Senao armazena-se a posicao corrente do registro na (posicao corrente-k) do vetor numChar FimSe

AS: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AS: o valor da variavel auxiliar j ultrapassou o numero total de atributos lidos da interconexao

Comentarios de argumentacao

Os atributos nao lidos da inteconexao inserida recebem o valor nulo, e a proxima interconexao da lista que contem a nova inteconexao inserido na cabeca recebe a lista de interconexoes atual

AE: se a listaAlvo nao for vazia

AE: o aux nao chegou ao final da lista de interconexoes

AE: se a listaAlvo for vazia

Definition at line 38 of file Interconexoes.c.

5.5.1.5 Vazia interconexaoVazia (Interconexao * listaAlvo)

Funcao: interconexaoVazia

Verifica se a lista de interconexoes esta vazia

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de interconexoes
------------------	--

Returns

variavel do tipo Vazia, indicando se a lista esta vazia

Assertiva de entrada: estrutura do tipo Interconexao

Assertiva de saida: condicao da Interconexao sendo vazia ou nao vazia

Funcao: interconexaoVazia

AssertivaSaida: VAZIA || NAO_VAZIA; AE: listaAlvo eh vazia

AE: listaAlvo nao eh vazia

AS: o retorno deve ser uma variavel do tipo Vazia

Definition at line 22 of file Interconexoes.c.

5.5.1.6 void liberaListaInterconexao (Interconexao * listaAlvo)

Funcao: liberaListaInterconexao

Desaloca a memoria reservada para toda celula pertecente a lista de interconexoes

Parameters

<i>listaAlvo</i>	lista a ser desalocada
------------------	------------------------

AssertivaEntrada: A lista nao deve ser vazia

AssertivaSaida: A lista deve estar vazia

Funcao: liberaListaInterconexao

AssertivaEntrada: interconexaoVazia(listaAlvo) == NAO_VAZIA;

AssertivaSaida: interconexaoVazia(listaAlvo) == VAZIA; Assertiva estrutural: aux1 é a listaAlvo, porem sendo per-

corrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Liberam os atributos alocados dinamicamente do elemento Interconexao corrente

AS: listaAlvo chegou ao fim

Definition at line 219 of file Interconexoes.c.

5.5.1.7 void mandarRecursoTransportado (Interconexao * listaAlvo)

Funcao: mandarRecursoTransportado

Muda cada referencia seja para adaptador ou para cidade, dependendo do em qual esta ligada, alterando o valor do recurso atual.

Parameters

<i>listaAlvo</i>	ponterio de referencia para o inicio da lista de interconexoes
------------------	--

Assertiva de entrada: interconexao - eh uma lista de interconexoes nao vazia

Assertiva de saida: Se o destino da interconexao for um adaptador Entao soma-se o recurso transportado ao Adaptador da lista de interconexoes Senao soma-se o recurso transportado ao Adaptador da lista de interconexoes FimSe

Funcao: mandarRecursoTransportado

AssertivaEntrada: interconexaoVazia(listaAlvo) == NAO_VAZIA; Asseriva estrutural: aux é a lista nao-nula de interconexoes

AE: aux nao chegou ao fim da lista de inteconexoes

AE: a interconexao corrente nao possui falha

Comentarios de argumentacao

Enquanto a lista de interconexoes eh percorrida, dependendo do destino da interconexao, soma-se o recurso ao respectivo destino

AE: o destino da interconexao eh um Adaptador

AE: o destino da interconexao eh uma Cidade

AS: a lista de interconexoes chegou ao fim

Definition at line 347 of file Interconexoes.c.

5.5.1.8 float tamanhoConexao (Interconexao * listaAlvo)

-----Funcoes de Calculo----- Funcao: tamanhoConexao

Calcula o tamanho da celula de conexao que eh passada para a funcao

Parameters

<i>listaAlvo</i>	ponteiro para a celula de interconexao
------------------	--

Returns

tamanho da celula de conexao que foi passada

Assertiva de entrada: interconexao - eh uma lista de interconexoes nao vazia

Assertiva de saida: tamanho de uma conexao

-----Funcoes de Calculo----- Funcao: tamanhoConexao

AssertivaEntrada: `interconexaoVazia(listaAlvo) == NAO_VAZIA;`

AssertivaSaida: `distancia > 0; < Valor no eixo x da posicao inicial`

`< Valor no eixo y da posicao inicial`

`< Valor no eixo x da posicao final`

`< Valor no eixo y da posicao final`

Comentarios de argumentacao

Calculando a distancia utilizando $((x_b - x_a)^2 + (y_b - y_a)^2)^{(1/2)}$

Definition at line 252 of file Interconexoes.c.

5.5.1.9 float tamanhoTotalConexao (Interconexao * listaAlvo)

Funcao: `tamanhoTotalConexao`

Calcula o tamanho total das conexoes da lista que eh passada

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de celulas de conexao
------------------	---

Returns

retorna o tamanho total das conexoes da lista

Assertiva de entrada: `interconexao` - eh uma lista de interconexoes nao vazia

Assertiva de saida: resultado da soma dos tamanhos de todas as conexoes

Funcao: `tamanhoTotalConexao`

AssertivaEntrada: `interconexaoVazia(listaAlvo) == NAO_VAZIA;`

AssertivaSaida: `resultado > 0;` Asseriva estrutural: aux é a lista nao-nula de interconexoes

AE: aux nao chegou ao fim da lista de inteconexoes

Comentarios de argumentacao

Enquanto a lista de interconexoes eh percorrida, o tamanho das conexoes sao somados e armazenados na variavel resultado

AS: a lista de interconexoes chegou ao fim

Definition at line 289 of file Interconexoes.c.

5.5.1.10 int totalGastoConserto (Interconexao *)

Funcao: `totalGastoConserto`

Calcula o custo total gasto com conserto de todas as celulas de conexao

Parameters

<i>listaAlvo</i>	ponteiro para o inicio da lista de interconexao
------------------	---

Returns

total gasto com conserto de todas as conexoes

5.6 app/header/Principal.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
```

Data Structures

- struct [cidade](#)
- struct [adaptador](#)
- struct [interconexao](#)
- struct [gerador](#)

Typedefs

- typedef enum [vazia](#) [Vazia](#)
- typedef enum [vazio](#) [Vazio](#)
- typedef enum [falha](#) [Falha](#)
- typedef enum [destino](#) [Destino](#)
- typedef struct [cidade](#) [Cidade](#)
- typedef struct [adaptador](#) [Adaptador](#)
- typedef struct [interconexao](#) [Interconexao](#)
- typedef struct [gerador](#) [Gerador](#)

Enumerations

- enum [vazia](#) { [VAZIA](#), [NAO_VAZIA](#) }
- enum [vazio](#) { [VAZIO](#), [NAO_VAZIO](#) }
- enum [falha](#) { [FALHA](#), [SEM_FALHA](#) }
- enum [destino](#) { [ADAPTADOR](#), [CIDADE](#) }

5.6.1 Typedef Documentation

5.6.1.1 typedef struct [adaptador](#) [Adaptador](#)

-----Adaptadores----- Cabecalho do elemento Adaptador

nome: nome do adaptador

posicao: vetor posicao, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

recursoRecebido: quantidade de recurso recebido por segundo pelo Adaptador

entrada: ponteiro que representa a interconexao de entrada do adaptador

saidas: representa as interconexoes realizadas/apontadas pelos adaptadores

quantidadeSaidas: quantidade de conexoes que o adaptador possui

proximo: representa a proxima Adaptador da lista de Adaptadores

5.6.1.2 typedef struct cidade Cidade

-----Cidades----- Cabecalho do elemento Cidade

nome: nome da cidade

posicao: vetor posicao, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

recursoNecessario: quantidade de recurso que a cidade precisa por segundo

recursoRecebido: quantidade de recurso recebido por segundo pela Cidade

recursoGasto: quantidade de recurso que a cidade usou

proximo: representa a proxima cidade da rede de cidades

entradas: ponteiro que representa a interconexao de entrada da cidade

5.6.1.3 typedef enum destino Destino

Enumeracao para detectar qual eh o ponto final de ligacao da conexao

5.6.1.4 typedef enum falha Falha

Enumeracao para detectar uma falha

5.6.1.5 typedef struct gerador Gerador

-----Geradores----- Cabecalho do elemento Gerador

nome: nome do gerador

posicao: vetor posicao, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

taxaProducao: quantidade de recurso que a cidade precisa por segundo

recursoProduzido: quantidade total de recurso produzido pelo gerador

custo: custo de geração por segundo

proximo: representa a proxima cidade da rede de cidades

saida: representa as interconexoes realizadas/apontadas pelos adaptadores

5.6.1.6 typedef struct interconexao Interconexao

-----Interconexoes----- Cabecalho do elemento Interconexao

nome: nome da interconexao

posicaoInicial: vetor posicaoInicial, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

posicaoFinal: vetor posicaoFinal, representando x na posicao[0] e y na posicao[1], ambas em km, representando tambem a posicao do adaptador na interface

tagDestino: tag para indentificar qual é a ligação final de cada conexao, seja cidade ou adaptador

chanceFalha: chance de falha por segundo

tempoConcerto: tempo de concerto em caso de falha em segundos

custoConcerto: custo do concerto em segundos

numeroFalha: total de falhas

tagFalha: indica se houve falha

capacidadeMaxima: capacidade maxima da interconexao

recursoTranstortado: quantidade de recuso que esta sendo transportado pela conexao no turno

proximo: representa a proxima interconexao na lista de interconexoes

entradaAdaptador: aponta, caso a entrada seja um adaptador, para o adaptador cuja saída eh esta interconexao

saidaAdaptador: aponta, caso a saida seja um adaptador, para o adaptador cuja entrada eh esta interconexao

entradaInterconexao: aponta, caso a entrada seja interconexao, para a interconexao cuja saida eh esta interconexao

saidaInterconexao: aponta, caso a saida seja interconexao, para a interconexao cuja entrada eh esta interconexao

proximoEntradaAdaptador: ponteiro para a proxima entrada do adaptador relacionado a interconexao

proximoSaidaAdaptador: ponteiro para a proxima saida do adaptador relacionado a interconexao

proximoEntradaCidade: ponteiro para a proxima cidade de destino

entradaGerador: aponta, caso a entrada seja um gerador, para o gerador cuja saida eh esta interconexao

saidaCidade: aponta, caso a saida seja uma cidade, para a cidade cuja entrada eh esta interconexao

5.6.1.7 typedef enum vazia Vazia

-----Enumeracoes----- Enumeracao para detectar se a lista esta vazia ou nao

5.6.1.8 typedef enum vazio Vazio

Enumeracao para detectar se a lista esta vazia ou nao

5.6.2 Enumeration Type Documentation

5.6.2.1 enum destino

Enumeracao para detectar qual eh o ponto final de ligacao da conexao

Enumerator:

ADAPTADOR

CIDADE

Definition at line 39 of file Principal.h.

5.6.2.2 enum falha

Enumeracao para detectar uma falha

Enumerator:

FALHA

SEM_FALHA

Definition at line 30 of file Principal.h.

5.6.2.3 enum vazia

-----Enumeracoes----- Enumeracao para detectar se a lista esta vazia ou nao

Enumerator:

VAZIA
NAO_VAZIA

Definition at line 12 of file Principal.h.

5.6.2.4 enum vazio

Enumeracao para detectar se a lista esta vazia ou nao

Enumerator:

VAZIO
NAO_VAZIO

Definition at line 21 of file Principal.h.

5.7 app/src/Adaptadores.c File Reference

```
#include "../header/Adaptadores.h"
#include <assert.h>
```

Functions

- [Adaptador * criaListaAdaptador \(\)](#)
- [Vazio adaptadorVazio \(Adaptador *listaAlvo\)](#)
- [Adaptador * insereAdaptador \(char *registro, Adaptador *listaAlvo\)](#)
- void [imprimeListaAdaptador \(Adaptador *listaAlvo\)](#)
- void [liberaListaAdaptador \(Adaptador *listaAlvo\)](#)
- void [defineDistribuicao \(Adaptador *listaAlvo\)](#)

5.7.1 Function Documentation

5.7.1.1 Vazio adaptadorVazio (Adaptador * listaAlvo)

Funcao: adaptadorVazio

AssertivaSaida: VAZIO || NAO_VAZIO; AE: listaAlvo eh vazia

AE: listaAlvo nao eh vazia

AS: o retorno deve ser uma variavel do tipo Vazia

Definition at line 22 of file Adaptadores.c.

5.7.1.2 Adaptador* criaListaAdaptador ()

-----Funcoes Basicas----- Funcao: criaListaAdaptador

AssertivaSaida: NULL;

Definition at line 12 of file Adaptadores.c.

5.7.1.3 void defineDistribuicao (Adaptador * listaAlvo)

Funcoes de Calculo

Funcao: defineDistribuicao

AssertivaEntrada: adaptadorVazio(listaAlvo) == NAO_VAZIO; para cada adaptador da lista: adaptador->saidas[i] != null; < conexoes do adaptador

< quantidade de recurso transportado

Assertiva estrutural: aux é a lista nao-nula de adaptadores

AE: aux nao chegou ao fim da lista de adaptadores

Assertiva estrutural: somatorio eh a soma das capacidades maximas do adaptador corrente

AE: o adaptador corrente possui saidas

Assertiva estrutural: conexao eh a lista de saidas do adaptador corrente

Comentarios de argumentacao

Enquanto percorre a lista de conexoes que saem do adaptador somam-se a capacidade maxima de todas as conexoes

AE: a lista de saidas do adaptador nao chegou ao fim

AS: a lista de saidas do adaptador chegou ao fim

Assertiva estrutural: conexao eh a lista de saidas do adaptador corrente

AE: a lista de saidas do adaptador nao chegou ao fim

AE: a saida corrente do adaptador nao possui falha

Assertiva estrutural: recursoTransportado é a quantidade de recurso que cada conexao vai transportar no turno

Comentarios de argumentacao

Se a a capacidade maxima da saida corrente do adaptador for maior ou igual ao recursoTransportado Entao o recursoTransportado da saida corrente do adaptador recebe o recursoTransportado Senao recebe a capacidade-Maxima da saida corrente do adaptador FimSe

AS: a saida corrente do adaptador possui falha

AS: a lista de saidas do adaptador chegou ao fim

AS: a lista de adaptadores chegou ao fim

Definition at line 211 of file Adaptadores.c.

5.7.1.4 void imprimeListaAdaptador (Adaptador * listaAlvo)

Funcao: imprimeListaAdaptador

AssertivaEntrada: adaptadorVazio(listaAlvo) == NAO_VAZIO; Asseriva estrutural: aux é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Imprime os atributos do adaptador corrente

AS: listaAlvo chegou ao fim

Definition at line 149 of file Adaptadores.c.

5.7.1.5 Adaptador* insereAdaptador (char * registro, Adaptador * listaAlvo)

Funcao: insereAdaptador

AssertivaEntrada: registro != NULL; < Alocacao de um novo adaptador

< Alocacao de um vetor do tamanho do registro

< Variaveis de auxilio

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Comecando de registro[2], enquanto registro[i] for um caracter irrelevante, soma-se 1 a variavel de auxilio i

AS: a posicao corrente do registro possui um caracter irrelevante

Asseriva estrutural: o nome do novo adaptador possui tamanho i-1

AE: o valor da variavel auxiliar j deve ser menor ou igual ao numero total de atributos lidos do adaptador

AE: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AE: o valor de j eh igual a 0

Comentarios de argumentacao

A (posicao corrente-k) recebe um caracter finalizador

AS: o valor de j eh maior que 0

AE: o valor de j eh maior que 0

Comentarios de argumentacao

A (posicao corrente-k) do vetor numChar recebe um caracter finalizador

Comentarios de argumentacao

De acordo com o valor da variavel auxiliar j, armazena-se o vetor numChar no seu respectivo atributo lido

AS: o valor de j eh maior que 4

AS: a posicao corrente do registro possui um caracter relevante

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Se o valor da variavel de auxilio j for 0 Entao armazena-se a posicao corrente do registro na (posicao corrente-k) do nome do adaptador Senao armazena-se a posicao corrente do registro na (posicao corrente-k) do vetor numChar FimSe

AS: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AS: o valor da variavel auxiliar j ultrapassou o numero total de atributos lidos da adaptador

Comentarios de argumentacao

Os atributos nao lidos do adaptador inserido recebem o valor nulo, e o proximo adaptador da lista que contem o novo adaptador inserido na cabeca recebe a lista de adaptadores atual

Definition at line 38 of file Adaptadores.c.

5.7.1.6 void liberaListaAdaptador (Adaptador * listaAlvo)

Funcao: liberaListaAdaptador

AssertivaEntrada: adaptadorVazio(listaAlvo) == NAO_VAZIO;

AssertivaSaida: adaptadorVazio(listaAlvo) == VAZIO; Asseriva estrutural: aux1 é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Liberam os atributos alocados dinamicamente do elemento Adaptador corrente

AS: listaAlvo chegou ao fim

Definition at line 179 of file Adaptadores.c.

5.8 app/src/Cidades.c File Reference

```
#include "../header/Cidades.h"
#include <assert.h>
```

Functions

- [Cidade * criaListaCidade \(\)](#)
- [Vazia cidadeVazia \(Cidade *listaAlvo\)](#)
- [Cidade * insereCidade \(char *registro, Cidade *listaAlvo\)](#)
- [void imprimeListaCidade \(Cidade *listaAlvo\)](#)
- [void liberaListaCidade \(Cidade *listaAlvo\)](#)
- [int recursoGastoTotal \(Cidade *cidade\)](#)

5.8.1 Function Documentation

5.8.1.1 Vazia cidadeVazia (Cidade * listaAlvo)

Funcao: cidadeVazia

AssertivaSaida: VAZIA || NAO_VAZIA; AE: listaAlvo eh vazia

AE: listaAlvo nao eh vazia

AS: o retorno deve ser uma variavel do tipo Vazia

Definition at line 22 of file Cidades.c.

5.8.1.2 Cidade* criaListaCidade ()

-----Funcoes Basicas----- Funcao: criaListaCidade

AssertivaSaida: NULL;

Definition at line 12 of file Cidades.c.

5.8.1.3 void imprimeListaCidade (Cidade * listaAlvo)

Funcao: imprimeListaCidade

AssertivaEntrada: cidadeVazia(listaAlvo) == NAO_VAZIA; Asseriva estrutural: aux é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Imprime os atributos da cidade corrente

AS: listaAlvo chegou ao fim

Definition at line 149 of file Cidades.c.

5.8.1.4 Cidade* insereCidade (char * registro, Cidade * listaAlvo)

Funcao: insereCidade

AssertivaEntrada: registro != NULL; < Alocacao da nova cidade

< Alocacao de um vetor do tamanho do registro

< Variaveis de auxilio

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Comecando de registro[2], enquanto registro[i] for um caracter irrelevante, soma-se 1 a variavel de auxilio i

AS: a posicao corrente do registro possui um caracter irrelevante

Asseriva estrutural: o nome da nova cidade possui tamanho i-1

AE: o valor da variavel auxiliar j deve ser menor ou igual ao numero total de atributos lidos da cidade

AE: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AE: o valor de j eh igual a 0

Comentarios de argumentacao

A (posicao corrente-k) recebe um caracter finalizador

AS: o valor de j eh maior que 0

AE: o valor de j eh maior que 0

Comentarios de argumentacao

A (posicao corrente-k) do vetor numChar recebe um caracter finalizador

Comentarios de argumentacao

De acordo com o valor da variavel auxiliar j, armazena-se o vetor numChar no seu respectivo atributo lido

AS: o valor de j eh maior que 4

AS: a posicao corrente do registro possui um caracter relevante

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Se o valor da variavel de auxilio j for 0 Entao armazena-se a posicao corrente do registro na (posicao corrente-k) do nome da cidade Senao armazena-se a posicao corrente do registro na (posicao corrente-k) do vetor numChar FimSe

AS: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AS: o valor da variavel auxiliar j ultrapassou o numero total de atributos lidos da cidade

Comentarios de argumentacao

Os atributos nao lidos da cidade inserida recebem o valor nulo, e a proxima cidade da lista que contem a nova cidade inserido na cabeca recebe a lista de cidades atual

Definition at line 38 of file Cidades.c.

5.8.1.5 void liberaListaCidade (Cidade * listaAlvo)

Funcao: liberaListaCidade

AssertivaEntrada: cidadeVazia(listaAlvo) == NAO_VAZIA;

AssertivaSaida: cidadeVazia(listaAlvo) == VAZIA; Asseriva estrutural: aux1 é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Liberam os atributos alocados dinamicamente do elemento Cidade corrente

AS: listaAlvo chegou ao fim

Definition at line 179 of file Cidades.c.

5.8.1.6 `int recursoGastoTotal (Cidade * cidade)`

-----Funcoes de Calculo----- Funcao: recursoGastoTotal

Definition at line 207 of file Cidades.c.

5.9 app/src/Geradores.c File Reference

```
#include "../header/Geradores.h"
#include <assert.h>
```

Functions

- `Gerador * criaListaGerador ()`
- `Vazio geradorVazio (Gerador *listaAlvo)`
- `Gerador * insereGerador (char *registro, Gerador *listaAlvo)`
- `void imprimeListaGerador (Gerador *listaAlvo)`
- `void liberaListaGerador (Gerador *listaAlvo)`

5.9.1 Function Documentation

5.9.1.1 `Gerador* criaListaGerador ()`

Funcao: criaListaGerador

AssertivaSaida: NULL;

Definition at line 10 of file Geradores.c.

5.9.1.2 `Vazio geradorVazio (Gerador * listaAlvo)`

Funcao: criaListaCidade

AssertivaSaida: VAZIO || NAO_VAZIO; AE: listaAlvo eh vazia

AE: listaAlvo nao eh vazia

AS: o retorno deve ser uma variavel do tipo Vazia

Definition at line 20 of file Geradores.c.

5.9.1.3 `void imprimeListaGerador (Gerador * listaAlvo)`

Funcao: imprimeListaGerador

AssertivaEntrada: geradorVazio(listaAlvo) == NAO_VAZIO; Asseriva estrutural: aux é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Imprime os atributos do gerador corrente

AS: listaAlvo chegou ao fim

Definition at line 147 of file Geradores.c.

5.9.1.4 Gerador* insereGerador (char * *registro*, Gerador * *listaAlvo*)

Funcao: insereGerador

AssertivaEntrada: registro != NULL; < Alocacao de um novo gerador

< Alocacao de um vetor do tamanho do registro

< Variaveis de auxilio

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Comecando de registro[2], enquanto registro[i] for um caracter irrelevante, soma-se 1 a variavel de auxilio i

AS: a posicao corrente do registro possui um caracter irrelevante

Asseriva estrutural: o nome do novo adaptador possui tamanho i-1

AE: o valor da variavel auxiliar j deve ser menor ou igual ao numero total de atributos lidos do gerador

AE: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AE: o valor de j eh igual a 0

Comentarios de argumentacao

A (posicao corrente-k) recebe um caracter finalizador

AS: o valor de j eh maior que 0

AE: o valor de j eh maior que 0

Comentarios de argumentacao

A (posicao corrente-k) do vetor numChar recebe um caracter finalizador

Comentarios de argumentacao

De acordo com o valor da variavel auxiliar j, armazena-se o vetor numChar no seu respectivo atributo lido

AS: o valor de j eh maior que 5

AS: a posicao corrente do registro possui um caracter relevante

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Se o valor da variavel de auxilio j for 0 Entao armazena-se a posicao corrente do registro na (posicao corrente-k) do nome do gerador Senao armazena-se a posicao corrente do registro na (posicao corrente-k) do vetor numChar FimSe

AS: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AS: o valor da variavel auxiliar j ultrapassou o numero total de atributos lidos do gerador

Comentarios de argumentacao

Os atributos nao lidos do gerador inserido recebem o valor nulo, e o proximo gerador da lista que contem o novo gerador inserido na cabeca recebe a lista de geradores atual

Definition at line 36 of file Geradores.c.

5.9.1.5 void liberaListaGerador (Gerador * *listaAlvo*)

Funcao: liberaListaGerador

AssertivaEntrada: geradorVazio(listaAlvo) == NAO_VAZIO;

AssertivaSaida: geradorVazio(listaAlvo) == VAZIO; Asseriva estrutural: aux1 é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Liberam os atributos alocados dinamicamente do elemento Gerador corrente

AS: listaAlvo chegou ao fim

Definition at line 177 of file Geradores.c.

5.10 app/src/Geral.c File Reference

```
#include "../header/Geral.h"
#include <assert.h>
```

Functions

- void [conecta](#) ([Cidade](#) *cidades, [Gerador](#) *geradores, [Interconexao](#) *interconexoes, [Adaptador](#) *adaptadores)
- void [verifica](#) ([Cidade](#) *cidades, [Gerador](#) *geradores, [Interconexao](#) *interconexoes, [Adaptador](#) *adaptadores)

5.10.1 Function Documentation

5.10.1.1 void conecta ([Cidade](#) * *cidades*, [Gerador](#) * *geradores*, [Interconexao](#) * *interconexoes*, [Adaptador](#) * *adaptadores*)

Funcao: conecta

AssertivaEntrada:

AssertivaSaida: < Ponteiro auxiliar para cidade

< Ponteiro auxiliar para gerador

< Ponteiro auxiliar para interconexao

< Ponteiro auxiliar para interconexao

< Ponteiro auxiliar para adaptador

AE: o auxI1 ainda nao chegou ao fim da lista de interconexoes

AE: o auxA ainda nao chegou ao fim da lista de adaptadores

AE: a posicao inicial da interconexao apontada por auxI1 coincide com a posicao do adaptador apontado por auxA

Comentarios de argumentacao

insere-se o adaptador como entrada da interconexao insere-se a interconexao na lista de saidas do adaptador

AE: a posicao final da interconexao apontada por auxI1 coincide com a posicao do adaptador apontado por auxA

Comentarios de argumentacao

insere-se o adaptador como saida da interconexao insere-se a interconexao na lista de entradas do adaptador

AS: o auxA chegou ao fim da lista de adaptadores

AE: o auxG ainda nao chegou ao fim da lista de geradores

AE: a posicao inicial da interconexao apontada por auxI1 coincide com a posicao do gerador apontado por auxG

Comentarios de argumentacao

insere-se a interconexao como saida do gerador insere-se o gerador como entrada da interconexao

AS: o auxG chegou ao fim da lista de geradores

AE: o auxC ainda nao chegou ao fim da lista de cidades

AE: a posicao final da interconexao apontada por auxI1 coincide com a posicao da cidade apontado por auxC

Comentarios de argumentacao

insere-se a cidade como saida da lista de interconexao insere-se a interconexao na lista de entradas da cidade

AS: o auxG chegou ao fim da lista de geradores

AE: o auxI2 ainda nao chegou ao fim da lista de interconexoes

AE: a posicao inicial da interconexao apontada por auxI1 coincide com a posicao final da interconexao apontada por auxI2

Comentarios de argumentacao

insere-se a interconexao auxI1 como saida da interconexao auxI2 insere-se a interconexao auxI2 como entrada da interconexao auxI1

AE: a posicao final da interconexao apontada por auxI1 coincide com a posicao inicial da interconexao apontada por auxI2

Comentarios de argumentacao

insere-se a interconexao auxI2 como saida da interconexao auxI1 insere-se a interconexao auxI1 como entrada da interconexao auxI2

AS: o auxI2 chegou ao fim da lista de interconexoes

AS: o auxI1 chegou ao fim da lista de interconexoes

Definition at line 11 of file Geral.c.

5.10.1.2 void verifica (Cidade * cidades, Gerador * geradores, Interconexao * interconexoes, Adaptador * adaptadores)

Funcao: verifica

AssertivaEntrada:

AssertivaSaida: < Ponteiro auxiliar para cidade

< Ponteiro auxiliar para gerador

< Ponteiro auxiliar para interconexao

< Ponteiro auxiliar para adaptador

AE: a lista de cidades nao eh vazia

AE: o auxC ainda nao chegou ao fim da lista de cidades

AE: a cidade corrente nao possui entradas

AS: o auxC chegou ao fim da lista de cidades

AE: a lista de cidades eh vazia

AE: a lista de geradores nao eh vazia

AE: o auxG ainda nao chegou ao fim da lista de geradores

AE: o gerador corrente nao possui saidas

AS: o auxG chegou ao fim da lista de geradores

AE: a lista de geradores eh vazia

AE: a lista de interconexoes nao eh vazia

AE: o auxI ainda nao chegou ao fim da lista de interconexoes

AE: a interconexao nao possui entradas em geradores, adaptadores e interconexoes

AE: a interconexao nao possui saidas em geradores, adaptadores e interconexoes

AS: o auxI chegou ao fim da lista de interconexoes

AE: a lista de interconexoes eh vazia

AE: a lista de adaptadores nao eh vazia

AE: o auxA ainda nao chegou ao fim da lista de adaptadores

AE: o adaptador corrente nao possui entradas

AE: o adaptador corrente nao possui saidas

AS: o auxA chegou ao fim da lista de adaptadores

AE: a lista de adaptadores eh vazia

Definition at line 132 of file Geral.c.

5.11 app/src/Interconexoes.c File Reference

```
#include "../header/Interconexoes.h"
#include <assert.h>
```

Functions

- [Interconexao * criaListaInterconexao \(\)](#)
- [Vazia interconexaoVazia \(Interconexao *listaAlvo\)](#)
- [Interconexao * insereInterconexao \(char *registro, Interconexao *listaAlvo\)](#)
- void [imprimeListaInterconexao \(Interconexao *listaAlvo\)](#)
- void [liberaListaInterconexao \(Interconexao *listaAlvo\)](#)
- float [tamanhoConexao \(Interconexao *listaAlvo\)](#)
- float [tamanhoTotalConexao \(Interconexao *listaAlvo\)](#)
- [Falha calculaFalha \(\)](#)
- void [mandarRecursoTransportado \(Interconexao *listaAlvo\)](#)

5.11.1 Function Documentation

5.11.1.1 Falha calculaFalha ()

Funcao: calculaFalha

AssertivaSaida: FALHA || SEM_FALHA; Asseriva estrutural: num eh um numero gerado aleatoriamente

AE: chance deve ser maior que 0 e maior ou igual a num

AE: chance deve ser menor que 0 ou menor que num

AS: o retorno deve ser uma variavel do tipo Falha

Definition at line 322 of file Interconexoes.c.

5.11.1.2 Interconexao* criaListaInterconexao ()

-----Funcoes Basicas----- Funcao: criaListaInterconexao

AssertivaSaida: NULL;

Definition at line 12 of file Interconexoes.c.

5.11.1.3 void imprimeListaInterconexao (Interconexao * listaAlvo)

Funcao: imprimeListaInterconexao

AssertivaEntrada: interconexaoVazia(listaAlvo) == NAO_VAZIA; Asseriva estrutural: aux é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Imprime os atributos da interconexao corrente

AS: listaAlvo chegou ao fim

Definition at line 189 of file Interconexoes.c.

5.11.1.4 Interconexao* insereInterconexao (char * registro, Interconexao * listaAlvo)

Funcao: insereInterconexao

AssertivaEntrada: registro != NULL; < Alocacao da nova Interconexao

< Variavel auxiliar para percorrer a lista e inserir o elemento no final

< Alocacao de um vetor do tamanho do registro

< Variaveis de auxilio

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Comecando de registro[2], enquanto registro[i] for um caracter irrelevante, soma-se 1 a variavel de auxilio i

AS: a posicao corrente do registro possui um caracter irrelevante

Asseriva estrutural: o nome da nova interconexao possui tamanho i-1

AE: o valor da variavel auxiliar j deve ser menor ou igual ao numero total de atributos lidos da interconexao

AE: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AE: o valor de j eh igual a 0

Comentarios de argumentacao

A (posicao corrente-k) recebe um caracter finalizador

AS: o valor de j eh maior que 0

AE: o valor de j eh maior que 0

Comentarios de argumentacao

A (posicao corrente-k) do vetor numChar recebe um caracter finalizador

Comentarios de argumentacao

De acordo com o valor da variavel auxiliar j, armazena-se o vetor numChar no seu respectivo atributo lido

AS: o valor de j eh maior que 8

AS: a posicao corrente do registro possui um caracter relevante

AE: a posicao corrente do registro possui um caracter relevante

Comentarios de argumentacao

Se o valor da variavel de auxilio j for 0 Entao armazena-se a posicao corrente do registro na (posicao corrente-k) do nome da interconexao Senao armazena-se a posicao corrente do registro na (posicao corrente-k) do vetor numChar FimSe

AS: a posicao corrente do registro possui um caracter irrelevante, ou um caracter finalizador

AS: o valor da variavel auxiliar j ultrapassou o numero total de atributos lidos da interconexao

Comentarios de argumentacao

Os atributos nao lidos da inteconexao inserida recebem o valor nulo, e a proxima interconexao da lista que contem a nova inteconexao inserido na cabeca recebe a lista de interconexoes atual

AE: se a listaAlvo nao for vazia

AE: o aux nao chegou ao final da lista de interconexoes

AE: se a listaAlvo for vazia

Definition at line 38 of file Interconexoes.c.

5.11.1.5 **Vazia interconexao**Vazia (**Interconexao** * *listaAlvo*)

Funcao: interconexaoVazia

AssertivaSaida: VAZIA || NAO_VAZIA; AE: listaAlvo eh vazia

AE: listaAlvo nao eh vazia

AS: o retorno deve ser uma variavel do tipo Vazia

Definition at line 22 of file Interconexoes.c.

5.11.1.6 **void liberaListaInterconexao** (**Interconexao** * *listaAlvo*)

Funcao: liberaListaInterconexao

AssertivaEntrada: interconexaoVazia(listaAlvo) == NAO_VAZIA;

AssertivaSaida: interconexaoVazia(listaAlvo) == VAZIA; Assertiva estrutural: aux1 é a listaAlvo, porem sendo percorrida

AE: listaAlvo nao chegou ao fim Comentarios de argumentacao

Liberam os atributos alocados dinamicamente do elemento Interconexao corrente

AS: listaAlvo chegou ao fim

Definition at line 219 of file Interconexoes.c.

5.11.1.7 **void mandarRecursoTransportado** (**Interconexao** * *listaAlvo*)

Funcao: mandarRecursoTransportado

AssertivaEntrada: interconexaoVazia(listaAlvo) == NAO_VAZIA; Asseriva estrutural: aux é a lista nao-nula de interconexoes

AE: aux nao chegou ao fim da lista de inteconexoes

AE: a interconexao corrente nao possui falha

Comentarios de argumentacao

Enquanto a lista de interconexoes eh percorrida, dependendo do destino da interconexao, soma-se o recurso ao respectivo destino

AE: o destino da interconexao eh um Adaptador

AE: o destino da interconexao eh uma Cidade

AS: a lista de interconexoes chegou ao fim

Definition at line 347 of file Interconexoes.c.

5.11.1.8 float tamanhoConexao (Interconexao * listaAlvo)

-----Funcoes de Calculo----- Funcao: tamanhoConexao

AssertivaEntrada: interconexaoVazia(listaAlvo) == NAO_VAZIA;

AssertivaSaida: distancia > 0; < Valor no eixo x da posicao inicial

< Valor no eixo y da posicao inicial

< Valor no eixo x da posicao final

< Valor no eixo y da posicao final

Comentarios de argumentacao

Calculando a distancia utilizando $((x_b - x_a)^2 + (y_b - y_a)^2)^{1/2}$

Definition at line 252 of file Interconexoes.c.

5.11.1.9 float tamanhoTotalConexao (Interconexao * listaAlvo)

Funcao: tamanhoTotalConexao

AssertivaEntrada: interconexaoVazia(listaAlvo) == NAO_VAZIA;

AssertivaSaida: resultado > 0; Asseriva estrutural: aux é a lista nao-nula de interconexoes

AE: aux nao chegou ao fim da lista de inteconexoes

Comentarios de argumentacao

Enquanto a lista de interconexoes eh percorrida, o tamanho das conexoes sao somados e armazenados na variavel resultado

AS: a lista de interconexoes chegou ao fim

Definition at line 289 of file Interconexoes.c.

5.12 app/src/Principal.c File Reference

```
#include "../header/Geral.h"
```

Functions

- int [main](#) ()

5.12.1 Function Documentation

5.12.1.1 int main ()

< Abre arquivo de entrada

< String auxiliar para obter registros

AE: o arquivo nao chegou ao fim

AE: a linha (registro) obtido do arquivo possui conteudo

AE: O registro obtido eh do tipo Cidade

AE: O registro obtido eh do tipo Gerador

AE: O registro obtido eh do tipo Interconexao

AE: O registro obtido eh do tipo Adaptador

AS: a linha (registro) obtido do arquivo nao possui conteudo, ou seja, o arquivo chegou ao fim

AS: o arquivo chegou ao fim

Comentarios de argumentacao

Imprimindo as listas obtidas a partir do arquivo de entrada

Comentarios de argumentacao

Conectando e verificando as listas

Comentarios de argumentacao

Desalocando as listas obtidas

Definition at line 3 of file Principal.c.

5.13 app/test/Adaptadores_unittest.c File Reference

```
#include <gtest/gtest.h>
#include "../header/Adaptadores.h"
```

Functions

- [TEST](#) ([criaListaAdaptador](#), [vazia](#))
- [TEST](#) ([adaptadorVazio](#), [vazio](#))
- [TEST](#) ([adaptadorVazio](#), [naoVazio](#))
- [TEST](#) ([insereAdaptador](#), [Vazio](#))
- [TEST](#) ([insereAdaptador](#), [naoVazio](#))

5.13.1 Function Documentation

5.13.1.1 TEST ([criaListaAdaptador](#) , [vazia](#))

Definition at line 4 of file Adaptadores_unittest.c.

5.13.1.2 TEST ([adaptadorVazio](#) , [vazio](#))

Definition at line 11 of file Adaptadores_unittest.c.

5.13.1.3 TEST ([adaptadorVazio](#) , [naoVazio](#))

Definition at line 17 of file Adaptadores_unittest.c.

5.13.1.4 TEST ([insereAdaptador](#) , [Vazio](#))

Definition at line 24 of file Adaptadores_unittest.c.

5.13.1.5 TEST ([insereAdaptador](#) , [naoVazio](#))

Definition at line 39 of file Adaptadores_unittest.c.

5.14 app/test/Interconexoes_unittest.c File Reference

```
#include "../header/Interconexoes.h"
```

5.15 app/test/main_unittest.c File Reference

```
#include <gtest/gtest.h>
```

Functions

- int [main](#) (int argc, char **argv)

5.15.1 Function Documentation

5.15.1.1 int main (int *argc*, char ** *argv*)

Definition at line 3 of file main_unittest.c.

5.16 README.md File Reference

Index

ADAPTADOR

Principal.h, 36

Adaptador

Principal.h, 34

adaptador, 7

entradas, 7

nome, 7

posicao, 8

proximo, 8

quantidadeSaidas, 8

recursoRecebido, 8

saidas, 8

adaptadorVazio

Adaptadores.c, 37

Adaptadores.h, 15

Adaptadores.c

adaptadorVazio, 37

criaListaAdaptador, 37

defineDistribuicao, 37

imprimeListaAdaptador, 38

insereAdaptador, 38

liberaListaAdaptador, 39

Adaptadores.h

adaptadorVazio, 15

criaListaAdaptador, 15

defineDistribuicao, 16

imprimeListaAdaptador, 17

insereAdaptador, 17

liberaListaAdaptador, 18

Adaptadores_unittest.c

TEST, 50

app/header/Adaptadores.h, 15

app/header/Cidades.h, 19

app/header/Geradores.h, 22

app/header/Geral.h, 26

app/header/Interconexoes.h, 28

app/header/Principal.h, 34

app/src/Adaptadores.c, 37

app/src/Cidades.c, 40

app/src/Geradores.c, 42

app/src/Geral.c, 44

app/src/Interconexoes.c, 46

app/src/Principal.c, 49

app/test/Adaptadores_unittest.c, 50

app/test/Interconexoes_unittest.c, 51

app/test/main_unittest.c, 51

CIDADE

Principal.h, 36

calculaFalha

Interconexoes.c, 46

Interconexoes.h, 29

capacidadeMaxima

interconexao, 12

chanceFalha

interconexao, 12

Cidade

Principal.h, 34

cidade, 8

entradas, 9

nome, 9

posicao, 9

proximo, 9

recursoGasto, 9

recursoNecessario, 9

recursoRecebido, 9

cidadeVazia

Cidades.c, 40

Cidades.h, 19

Cidades.c

cidadeVazia, 40

criaListaCidade, 40

imprimeListaCidade, 40

insereCidade, 40

liberaListaCidade, 41

recursoGastoTotal, 42

Cidades.h

cidadeVazia, 19

criaListaCidade, 19

imprimeListaCidade, 20

insereCidade, 20

liberaListaCidade, 21

recursoGastoTotal, 22

conecta

Geral.c, 44

Geral.h, 26

criaListaAdaptador

Adaptadores.c, 37

Adaptadores.h, 15

criaListaCidade

Cidades.c, 40

Cidades.h, 19

criaListaGerador

Geradores.c, 42

Geradores.h, 22

criaListaInterconexao

Interconexoes.c, 46

Interconexoes.h, 29

custo

- gerador, 10
- custoConserto
 - interconexao, 12
- custoGeradores
 - Geradores.h, 22
- defineDistribuicao
 - Adaptadores.c, 37
 - Adaptadores.h, 16
- Destino
 - Principal.h, 35
- destino
 - Principal.h, 36
- entradaAdaptador
 - interconexao, 12
- entradaGerador
 - interconexao, 12
- entradaInterconexao
 - interconexao, 12
- entradas
 - adaptador, 7
 - cidade, 9
- FALHA
 - Principal.h, 36
- Falha
 - Principal.h, 35
- falha
 - Principal.h, 36
- Gerador
 - Principal.h, 35
- gerador, 9
 - custo, 10
 - nome, 10
 - posicao, 10
 - proximo, 10
 - recursoProduzido, 10
 - saida, 10
 - taxaProducao, 10
- geradorVazio
 - Geradores.c, 42
 - Geradores.h, 23
- Geradores.c
 - criaListaGerador, 42
 - geradorVazio, 42
 - imprimeListaGerador, 42
 - insereGerador, 42
 - liberaListaGerador, 43
- Geradores.h
 - criaListaGerador, 22
 - custoGeradores, 22
 - geradorVazio, 23
 - imprimeListaGerador, 23
 - insereGerador, 24
 - liberaListaGerador, 25
 - mandarRecursoProduzido, 25
 - recursoProduzidoTotal, 25
- Geral.c
 - conecta, 44
 - verifica, 45
- Geral.h
 - conecta, 26
 - verifica, 27
- imprimeListaAdaptador
 - Adaptadores.c, 38
 - Adaptadores.h, 17
- imprimeListaCidade
 - Cidades.c, 40
 - Cidades.h, 20
- imprimeListaGerador
 - Geradores.c, 42
 - Geradores.h, 23
- imprimeListaInterconexao
 - Interconexoes.c, 46
 - Interconexoes.h, 29
- insereAdaptador
 - Adaptadores.c, 38
 - Adaptadores.h, 17
- insereCidade
 - Cidades.c, 40
 - Cidades.h, 20
- insereGerador
 - Geradores.c, 42
 - Geradores.h, 24
- insereInterconexao
 - Interconexoes.c, 47
 - Interconexoes.h, 30
- Interconexao
 - Principal.h, 35
- interconexao, 11
 - capacidadeMaxima, 12
 - chanceFalha, 12
 - custoConserto, 12
 - entradaAdaptador, 12
 - entradaGerador, 12
 - entradaInterconexao, 12
 - nome, 12
 - numeroFalha, 12
 - posicaoFinal, 13
 - posicaoInicial, 13
 - proximo, 13
 - proximoEntradaAdaptador, 13
 - proximoEntradaCidade, 13
 - proximoSaidaAdaptador, 13
 - recursoTransportado, 13
 - saidaAdaptador, 13
 - saidaCidade, 13
 - saidaInterconexao, 13
 - tagDestino, 13
 - tagFalha, 13
 - tempoConserto, 14
- interconexaoVazia
 - Interconexoes.c, 48
 - Interconexoes.h, 31
- Interconexoes.c

- calculaFalha, 46
- criaListaInterconexao, 46
- imprimeListaInterconexao, 46
- insereInterconexao, 47
- interconexaoVazia, 48
- liberaListaInterconexao, 48
- mandarRecursoTransportado, 48
- tamanhoConexao, 48
- tamanhoTotalConexao, 49
- Interconexoes.h
 - calculaFalha, 29
 - criaListaInterconexao, 29
 - imprimeListaInterconexao, 29
 - insereInterconexao, 30
 - interconexaoVazia, 31
 - liberaListaInterconexao, 31
 - mandarRecursoTransportado, 32
 - tamanhoConexao, 32
 - tamanhoTotalConexao, 33
 - totalGastoConserto, 33
- liberaListaAdaptador
 - Adaptadores.c, 39
 - Adaptadores.h, 18
- liberaListaCidade
 - Cidades.c, 41
 - Cidades.h, 21
- liberaListaGerador
 - Geradores.c, 43
 - Geradores.h, 25
- liberaListaInterconexao
 - Interconexoes.c, 48
 - Interconexoes.h, 31
- main
 - main_unittest.c, 51
 - Principal.c, 49
- main_unittest.c
 - main, 51
- mandarRecursoProduzido
 - Geradores.h, 25
- mandarRecursoTransportado
 - Interconexoes.c, 48
 - Interconexoes.h, 32
- NAO_VAZIA
 - Principal.h, 37
- NAO_VAZIO
 - Principal.h, 37
- nome
 - adaptador, 7
 - cidade, 9
 - gerador, 10
 - interconexao, 12
- numeroFalha
 - interconexao, 12
- posicao
 - adaptador, 8
 - cidade, 9
 - gerador, 10
- posicaoFinal
 - interconexao, 13
- posicaoInicial
 - interconexao, 13
- Principal.h
 - ADAPTADOR, 36
 - CIDADE, 36
 - FALHA, 36
 - NAO_VAZIA, 37
 - NAO_VAZIO, 37
 - SEM_FALHA, 36
 - VAZIA, 37
 - VAZIO, 37
- Principal.c
 - main, 49
- Principal.h
 - Adaptador, 34
 - Cidade, 34
 - Destino, 35
 - destino, 36
 - Falha, 35
 - falha, 36
 - Gerador, 35
 - Interconexao, 35
 - Vazia, 36
 - vazia, 36
 - Vazio, 36
 - vazio, 37
- proximo
 - adaptador, 8
 - cidade, 9
 - gerador, 10
 - interconexao, 13
- proximoEntradaAdaptador
 - interconexao, 13
- proximoEntradaCidade
 - interconexao, 13
- proximoSaidaAdaptador
 - interconexao, 13
- quantidadeSaidas
 - adaptador, 8
- README.md, 51
- recursoGasto
 - cidade, 9
- recursoGastoTotal
 - Cidades.c, 42
 - Cidades.h, 22
- recursoNecessario
 - cidade, 9
- recursoProduzido
 - gerador, 10
- recursoProduzidoTotal
 - Geradores.h, 25
- recursoRecebido
 - adaptador, 8

- cidade, [9](#)
- recursoTransportado
 - interconexao, [13](#)
- SEM_FALHA
 - Principal.h, [36](#)
- saida
 - gerador, [10](#)
- saidaAdaptador
 - interconexao, [13](#)
- saidaCidade
 - interconexao, [13](#)
- saidaInterconexao
 - interconexao, [13](#)
- saidas
 - adaptador, [8](#)
- TEST
 - Adaptadores_unittest.c, [50](#)
- tagDestino
 - interconexao, [13](#)
- tagFalha
 - interconexao, [13](#)
- tamanhoConexao
 - Interconexoes.c, [48](#)
 - Interconexoes.h, [32](#)
- tamanhoTotalConexao
 - Interconexoes.c, [49](#)
 - Interconexoes.h, [33](#)
- taxaProducao
 - gerador, [10](#)
- tempoConserto
 - interconexao, [14](#)
- totalGastoConserto
 - Interconexoes.h, [33](#)
- VAZIA
 - Principal.h, [37](#)
- VAZIO
 - Principal.h, [37](#)
- Vazia
 - Principal.h, [36](#)
- vazia
 - Principal.h, [36](#)
- Vazio
 - Principal.h, [36](#)
- vazio
 - Principal.h, [37](#)
- verifica
 - Geral.c, [45](#)
 - Geral.h, [27](#)