**Arquitetura**

Labirinto.h

**Grafo**

**Lista**

**Labirinto**

**Principal**

S

C

C

S

S

C

Grafo.h

Lista.h

1. **Labirinto.h**

* LAB\_tpCondRet LAB\_CriarLabirinto ( int altura, int largura)
* LAB\_tpCondRet LAB\_CriarCaminho (int x, int y, char direcao)
* LAB\_tpCondRet LAB\_CriarEntrada (int x, int y )
* LAB\_tpCondRet LAB\_CriarSaida (int x, int y )
* LAB\_tpCondRet LAB\_SolucionarLab ( int \*\*\* buffer\_solucao )
* LAB\_tpCondRet LAB\_MostraLab ( void )
* LAB\_tpCondRet LAB\_Andar ( char direcao )
* LAB\_tpCondRet LAB\_Salvar ( char \* nome\_saida )
* LAB\_tpCondRet LAB\_Carregar ( char \* nome\_entrada )
* void LAB\_DestruirLab ( void )

1. **Grafo.h**

* GRF\_tpCondRet GRF\_CriarGrafo ( GRF\_tppGrafo \* ppGrafo ,

void (\* ExcluirValor ) ( void \* pValor ) )

* GRF\_tpCondRet GRF\_CriaVertice ( GRF\_tppGrafo pGrafo ,

void \* pValor , int chave )

* GRF\_tpCondRet GRF\_CriaAresta(GRF\_tppGrafo pGrafo , int chaveA , int chaveB)
* GRF\_tpCondRet GRF\_RemoveAresta(GRF\_tppGrafo pGrafo int chaveA, int chaveB)
* void GRF\_DestroiGrafo (GRF\_tppGrafo pGrafo)
* GRF\_tpCondRet GRF\_EsvaziaGrafo (GRF\_tppGrafo pGrafo)
* GRF\_tpCondRet GRF\_ObtemCaminho ( GRF\_tppGrafo pGrafo ,

int chaveOrigem , int chaveDestino , int \* buffer )

* GRF\_tpCondRet GRF\_ExisteVertice ( GRF\_tppGrafo pGrafo ,

int chave )

* GRF\_tpCondRet GRF\_AlteraCorrente ( GRF\_tppGrafo pGrafo ,

int chave )

* GRF\_tpCondRet GRF\_ExisteAresta ( GRF\_tppGrafo pGrafo ,

int chaveA , int chaveB )

* GRF\_tpCondRet GRF\_IrVizinho ( GRF\_tppGrafo pGrafo ,

int chave )

* GRF\_tpCondRet GRF\_ObterCorr ( GRF\_tppGrafo pGrafo ,

int \* chave , void \*\* ppValor )

* GRF\_tpCondRet GRF\_AlterarValorCorr ( GRF\_tppGrafo pGrafo ,

void \* pValor )

* GRF\_tpCondRet GRF\_Caminhar ( GRF\_tppGrafo pGrafo ,

int chaveDestino )

1. **Lista.h**

* LIS\_tpCondRet LIS\_CriarLista( LIS\_tppLista\* pLista ,

void ( \* ExcluirValor ) ( void \* pDado ) )

* void LIS\_DestruirLista( LIS\_tppLista pLista )
* LIS\_tpCondRet LIS\_EsvaziarLista( LIS\_tppLista pLista )
* LIS\_tpCondRet LIS\_InserirElementoAntes( LIS\_tppLista pLista ,

void \* pValor )

* LIS\_tpCondRet LIS\_InserirElementoApos( LIS\_tppLista pLista ,

void \* pValor )

* LIS\_tpCondRet LIS\_ExcluirElemento( LIS\_tppLista pLista )
* LIS\_tpCondRet LIS\_ObterValor( LIS\_tppLista pLista ,

void \*\* ppValor)

* LIS\_tpCondRet LIS\_IrInicioLista( LIS\_tppLista pLista )
* LIS\_tpCondRet LIS\_IrFinalLista( LIS\_tppLista pLista )
* LIS\_tpCondRet LIS\_AvancarElementoCorrente( LIS\_tppLista pLista ,

int numElem )

* LIS\_tpCondRet LIS\_AlterarValor( LIS\_tppLista pLista ,

void \* pValorNovo )