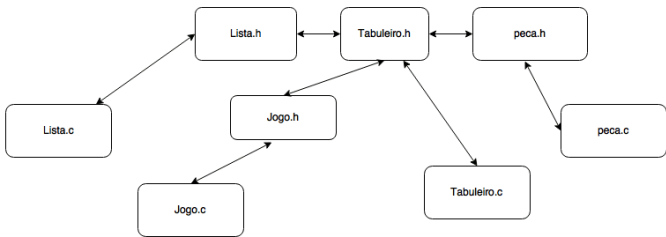


Arquitetura Da Aplicação



Funções de Acesso de Interface de cada Módulo

LISTA.H

```
LIS_tpCondRet LIS_CriarLista(LIS_tppLista *lista,char *idLista,void( * ExcluirValor ) ( void * pDado ));
LIS_tpCondRet LIS_InserirNo(LIS_tppLista lista, void *elemento);
LIS_tpCondRet LIS_ObterIdLista(LIS_tppLista lista, char idLista[4]);

LIS_tpCondRet LIS_ObterNo(LIS_tppLista lista, void **referencia);

LIS_tpCondRet LIS_ExcluirNoCorrente(LIS_tppLista lista);

LIS_tpCondRet LIS_IrProx(LIS_tppLista lista);

LIS_tpCondRet LIS_IrAnt(LIS_tppLista lista);

LIS_tpCondRet LIS_AlterarNoCorrente(LIS_tppLista lista, void *referencia);

voidEsvaziarLista(LIS_tppLista pLista);

LIS_tpCondRet LIS_RetornaNumElementos(LIS_tppLista lista, int *numElem);

LIS_tpCondRet LIS_DestroiLista(LIS_tppLista lista);
```

TABULEIRO.H

```
TAB_tpCondRet TAB_CriaTabuleiro(ptTabuleiro *tabu);

TAB_tpCondRet TAB_InserirPeca(ptTabuleiro tabuleiro,int x, int y,char cor,char id);

TAB_tpCondRet TAB_MoverPeca(ptTabuleiro tabu,int xo,int yi,int xd,int y2);

TAB_tpCondRet TAB_RetirarPeca(ptTabuleiro tabu,int x,int y);

TAB_tpCondRet TAB_ObterPeca(ptTabuleiro tabu,int x, int y, char *cor, char *id);

TAB_tpCondRet TAB_ObterListaAmeacantes(ptTabuleiro tabu,int x, int y,LIS_tppLista *listaAmeacantes);

TAB_tpCondRet TAB_ObterListaAmeacados(ptTabuleiro tabu,int x, int y,LIS_tppLista *listaAmeacados);

TAB_tpCondRet TAB_DestruirTabuleiro(ptTabuleiro tabu);

TAB_tpCondRet TAB_VerificaCheck(ptTabuleiro tabu,int xRei,int yRei);

TAB_tpCondRet TAB_VerificaCheckMate(ptTabuleiro tabu,int xRei,int yRei,int xAmeacante,int yAmeacante);

TAB_tpCondRet TAB_AchaPecaCheck(ptTabuleiro tabu,char cor,char id,int *xAmeacante,int *yAmeacante,int xRei,int yRei);

TAB_tpCondRet TAB_VerificaMovimentoRei(ptTabuleiro tabu,int xRei, int yRei, char cor,int xFim,int yFim);
```

JOGO.H

```
JOG_tpCondRet JOG_EfetuarJogada(ptJudge j, char corDaVez,int posiniX,int posiniY,int posFimX,int posFimY);

JOG_tpCondRet JOG_CriaJuiz(ptJudge j,char nomeJogadorA[30], char nomeJogadorB[30]);

JOG_tpCondRet JOG_ComecarJogo(ptJudge j,void (*InserirPecas)(TAB_tpCondRet(*InserirNoTab))(ptTabuleiro,int x,int y,char cor,char id,ptJudge j));

JOG_tpCondRet JOG_AssasinarJuiz(ptJudge j);

JOG_tpCondRet JOG_ObtemTabuleiro(ptJudge j,ptTabuleiro *tabu);

JOG_tpCondRet JOG_GeraMatrizTabuleiro(ptJudge j,char matriz[8][8][2]);

JOG_tpCondRet JOG_CriaJogador(ptJogador j,char *nome,char cor);

JOG_tpCondRet JOG_RetornoNomeJogador1(ptJudge j,char nome[30]);

JOG_tpCondRet JOG_RetornoNomeJogador2(ptJudge j,char nome[30]);

JOG_tpCondRet JOG_DevolveAmeacantes(ptJudge j,char *corAmeacantes,char *idAmeacantes,int *qtdAmeacantes,int x,int y);

JOG_tpCondRet JOG_DevolveAmeacados(ptJudge j,char *corAmeacados,char *idAmeacados,int *qtdAmeacados,int x,int y);
```

PECA.H

```
PEC_tpCondRet PEC_EnsinaMovimentosPecasConhecidas(Peca **novo);

PEC_tpCondRet PEC_EnsinaMovimentosPecasDesconhecidas(Peca **novo);

PEC_tpCondRet PEC_LiberaPeca(Peca *peca);

PEC_tpCondRet PEC_RetornaId(Peca *peca,char *id);

PEC_tpCondRet PEC_RetornaCor(Peca *peca,char *cor);

PEC_tpCondRet PEC_RetornaQtd_Mov(Peca *peca,int *qtdMov);

PEC_tpCondRet PEC_RetornaPodePular(Peca *peca,int *podePular);

PEC_tpCondRet PEC_RetornaXMovimento(Peca *peca,int i,int *x);

PEC_tpCondRet PEC_RetornaYMovimento(Peca *peca,int i,int *y);

PEC_tpCondRet PEC_EliminarPeca(Peca *peca);

PEC_tpCondRet PEC_AlteraCorEld(Peca *peca,char cor,char id);

PEC_tpCondRet PEC_CriaPeca(Peca **novo,char id,char cor);
```