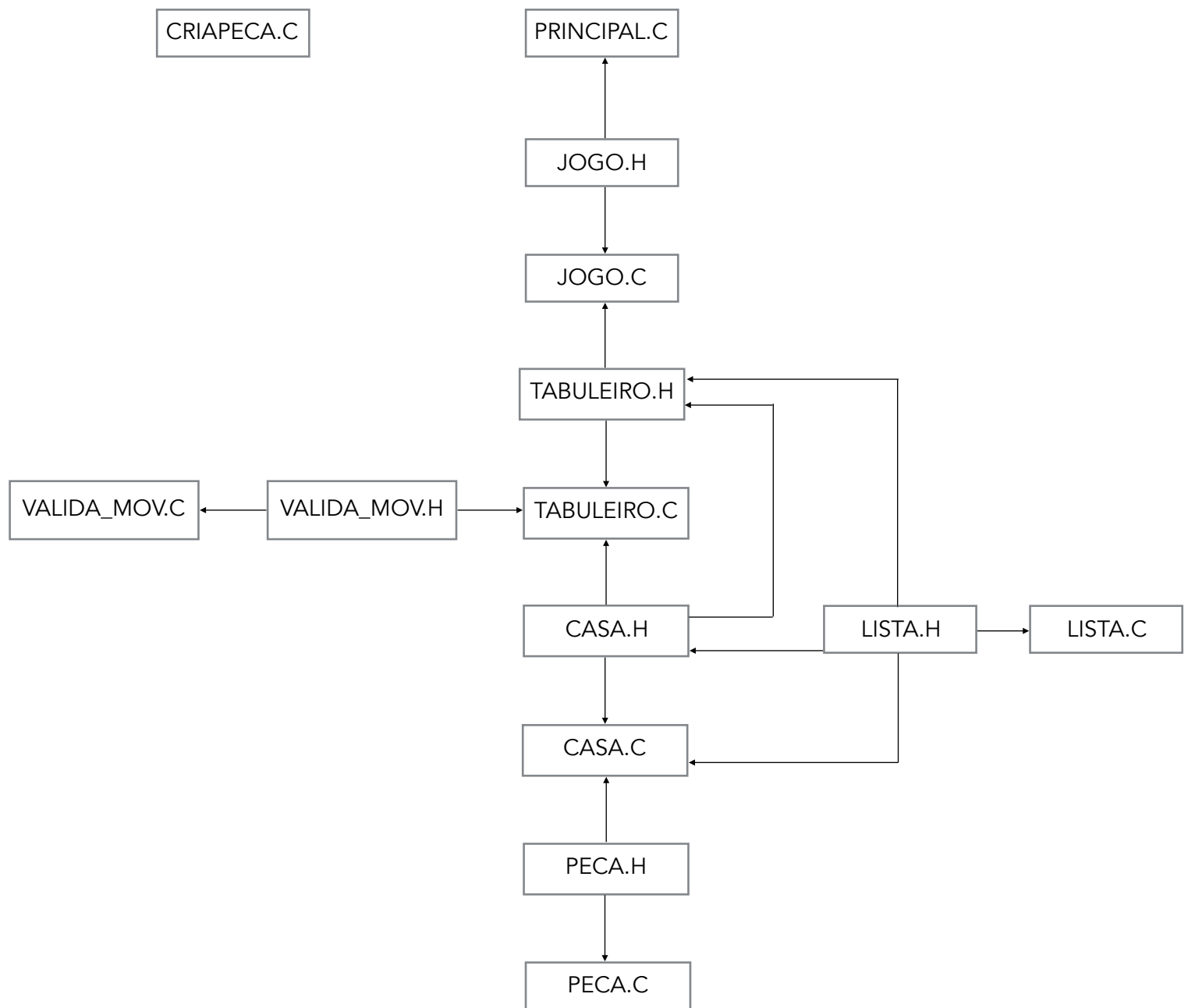


Rio de Janeiro, 13 de novembro de 2016
Pontifícia Universidade Católica do Rio de Janeiro
INF1301 - Programação Modular
Grupo: Ian Albuquerque Raymundo da Silva ;
Lucas Ferraço de Freitas;
Victor Augusto Lima Lins de Souza .

Arquitetura Completa da Aplicação



Funções de Acesso de Interface de cada Módulo:

• JOGO.H

- JGO_tpCondRet JGO_CriarJuiz(JGO_tppJuiz * pJuiz) ;
- JGO_tpCondRet JGO_DestruirJuiz(JGO_tppJuiz pJuiz) ;
- JGO_tpCondRet JGO_IniciarJogo(JGO_tppJuiz pJuiz, char* pathConfig) ;
- JGO_tpCondRet JGO_TerminarJogo(JGO_tppJuiz pJuiz) ;
- JGO_tpCondRet JGO_RealizarJogada(JGO_tppJuiz pJuiz,
JGO_tpCorJogador corJogadorAtual,
JGO_tpEventoOcorrido* eventoOcorrido,
char linhaCasaAtual,
char colunaCasaAtual,
char linhaCasaDestino,
char colunaCasaDestino) ;
- JGO_tpCondRet JGO_GetPrintTabuleiro(JGO_tppJuiz pJuiz, char** print) ;

• TABULEIRO.H

- TAB_tpCondRet TAB_CriarTabuleiro(TAB_tppTabuleiro * pTabuleiro, char* pathConfig) ;
- TAB_tpCondRet TAB_CopiarTabuleiro(TAB_tppTabuleiro * pTabuleiro,
TAB_tppTabuleiro tabuleiroOriginal) ;
- TAB_tpCondRet TAB_DestruirTabuleiro(TAB_tppTabuleiro pTabuleiro) ;
- TAB_tpCondRet TAB_InserirPecaTabuleiro(char coluna ,
char linha ,
char nomePeca ,
char corPeca ,
TAB_tppTabuleiro pTabuleiro) ;
- TAB_tpCondRet TAB_MoverPecaTabuleiro(char colInicial ,
char linInicial ,
char colFinal ,
char linFinal ,
TAB_tppTabuleiro pTabuleiro) ;
- TAB_tpCondRet TAB_RetirarPecaTabuleiro(char coluna , char linha , TAB_tppTabuleiro pTabuleiro) ;
- TAB_tpCondRet TAB_ObterCasaTabuleiro(char coluna ,
char linha ,
CSA_tppCasa * pCasa ,
TAB_tppTabuleiro pTabuleiro) ;
- TAB_tpCondRet TAB_ObterPecaTabuleiro(char coluna ,
char linha ,
char* pNomePeca ,
char* pCorPeca ,
TAB_tppTabuleiro pTabuleiro) ;
- TAB_tpCondRet TAB_ObterListaAmeacantesTabuleiro(char coluna ,
char linha ,
LIS_tppLista * pListaAmeacantesLinhas ,
LIS_tppLista * pListaAmeacantesColunas ,
TAB_tppTabuleiro pTabuleiro) ;
- TAB_tpCondRet TAB_ObterListaAmeacadosTabuleiro(char coluna ,
char linha ,
LIS_tppLista * pListaAmeacadosLinhas ,
LIS_tppLista * pListaAmeacadosColunas ,
TAB_tppTabuleiro pTabuleiro) ;

- TAB_tpCondRet TAB_ObterCasasComPeca(LIS_tppLista * pListaCasasLinhas ,
LIS_tppLista * pListaCasasColunas ,
char peca,
char cor,
TAB_tppTabuleiro pTabuleiro) ;
- TAB_tpCondRet TAB_GetPrintTabuleiro(TAB_tppTabuleiro pTabuleiro, char** print) ;

• VALIDA_MOV.H

- VMV_tpCondRet VMV_CriarConfigDir (VMV_tppConfigDir * pConfigDir ,
const char * pathConfigArq) ;
- VMV_tpCondRet VMV_CopiarConfigDir (VMV_tppConfigDir * pConfigDir ,
VMV_tppConfigDir configDirOriginal) ;
- VMV_tpCondRet VMV_DestruirConfigDir (VMV_tppConfigDir pConfigDir) ;
- VMV_tpCondRet VMV_LerTabuleiroInicial (VMV_tppConfigDir pConfig,
char ** pecas,
char ** cores,
int * num_casas) ;
- VMV_tpCondRet VMV_ChecarMovimentoPeca (VMV_tppConfigDir pConfig ,
VMV_tpMovimentoValido * movimento_valido ,
char peca ,
void * casa_atual ,
void * casa_destino ,
void ** casas ,
int num_casas ,
int num_dimensoes ,
int (** array_dimensao) (void * casa, void * aux) ,
int * array_sinal ,
int (* vazio) (void * casa, void * aux) ,
int (* inimigo) (void * casa, void* casa_atual, void * aux) ,
int * cond_especiais ,
int num_cond_especiais, void* aux) ;

• CASA.H

- CSA_tpCondRet CSA_CriarCasa(CSA_tppCasa * pCasa) ;
- CSA_tpCondRet CSA_CopiarCasa(CSA_tppCasa * pCasa, CSA_tppCasa casaOriginal) ;
- CSA_tpCondRet CSA_DestruirCasa(CSA_tppCasa pCasa) ;
- CSA_tpCondRet CSA_InserirPecaCasa(char nomePeca , char corPeca , CSA_tppCasa pCasa) ;
- CSA_tpCondRet CSA_RetirarPecaCasa(CSA_tppCasa pCasa) ;
- CSA_tpCondRet CSA_ObterPecaCasa(char* pNomePeca ,
char* pCorPeca ,
CSA_tppCasa pCasa) ;
- CSA_tpCondRet CSA_CompararCasa(CSA_tppCasa pCasa1 ,
CSA_tppCasa pCasa2 ,
int * igualdade) ;
- CSA_tpCondRet CSA_ObterListaAmeacantesCasa(LIS_tppLista * pListaAmeacantes ,
CSA_tppCasa pCasa) ;
- CSA_tpCondRet CSA_ObterListaAmeacadosCasa(LIS_tppLista * pListaAmeacados ,
CSA_tppCasa pCasa) ;
- CSA_tpCondRet CSA_ModificarListaAmeacantesCasa(CSA_tppCasa * vetorCasasAmeacantes ,
int qtdCasasAmeacantes ,
CSA_tppCasa pCasa) ;

- CSA_tpCondRet CSA_ModificarListaAmeacadosCasa(CSA_tppCasa * vetorCasasAmeacadas ,
int qtdCasasAmeacadas ,
CSA_tppCasa pCasa) ;
 - CSA_tpCondRet CSA_GetPrintCasa (CSA_tppCasa pCasa, char** print) ;
- PECA.H
 - PCA_tpCondRet PCA_CriarPeca(PCA_tppPeca * pPeca , char nomePeca , char corPeca) ;
 - PCA_tpCondRet PCA_CopiarPeca(PCA_tppPeca * pPeca , PCA_tppPeca pecaOriginal) ;
 - PCA_tpCondRet PCA_AlterarPeca(PCA_tppPeca pPeca , char nomePeca , char corPeca) ;
 - PCA_tpCondRet PCA_ObterValor(PCA_tppPeca pPeca , char * nomePeca , char * corPeca) ;
 - PCA_tpCondRet PCA_ComparaPecas(PCA_tppPeca pPeca_1 , PCA_tppPeca pPeca_2 ,
int * igualdade) ;
 - PCA_tpCondRet PCA_DestruirPeca(PCA_tppPeca pPeca) ;
 - PCA_tpCondRet PCA_GetPrintPeca(PCA_tppPeca pPeca, char** print) ;
- LISTA.H
 - LIS_tpCondRet LIS_CriarLista(LIS_tppLista* pLista ,
char * idLista ,
void (* ExcluirValor) (void * pDado) ,
int (* CompararValores) (void * pDado_1, void * pDado_2) ,
int (* Igual) (void * pDado_1, void * pDado_2)) ;
 - LIS_tpCondRet LIS_CopiarLista(LIS_tppLista* pLista , LIS_tppLista listaOriginal) ;
 - LIS_tpCondRet LIS_DestruirLista(LIS_tppLista pLista) ;
 - LIS_tpCondRet LIS_InserirElementoApos(LIS_tppLista pLista , void * pValor) ;
 - LIS_tpCondRet LIS_ExcluirElemento(LIS_tppLista pLista) ;
 - LIS_tpCondRet LIS_ObterValor(LIS_tppLista pLista , void ** elementoCorrente) ;
 - LIS_tpCondRet LIS_ObterId(LIS_tppLista pLista , char ** idLista) ;
 - LIS_tpCondRet LIS_AvancarElementoCorrente(LIS_tppLista pLista , int numElem) ;
 - LIS_tpCondRet LIS_AlteraValor(LIS_tppLista pLista , void * pValor) ;
 - LIS_tpCondRet LIS_VerificaVazia(LIS_tppLista pLista , int * vazia) ;
 - LIS_tpCondRet LIS_VerificaIgualdade(LIS_tppLista pLista1 , LIS_tppLista pLista2 , int * igualdade) ;
 - LIS_tpCondRet LIS_Esvazia(LIS_tppLista pLista) ;
 - LIS_tpCondRet LIS_ProcurarValor(LIS_tppLista pLista , void * pValor) ;