**Lista**

A lista é um ponteiro para cabeça da lista

Nó é um ponteiro para outro nó da lista

A lista possui na cabeça:

-Origem

-Fim

-Corrente

-Id

No da lista possui:

-Valor

-Anterior

-Posterior

If no->pProx != NULL

no->pProx->pAnt == no;

if no->pAnt != NULL

 no->pAnt->pProx == no;

if lista->numElem == 0

{

lista->pElemCorr == NULL;

e lista->pOrigemLista == NULL;

e pFimLista == NULL.;

}

if lista->numElem > 0,

 lista->pElemCorr != NULL;

if lista->numElem == 1

{

lista->pElemCorr == lista->pOrigemLista;

 lista->pOrigemLista == lista->pFimLista;

}

## Tabuleiro

Tabuleio é um vetor de ponteiros

If tabuleiro != NULL

Tabuleiro->x[0]->[y0] ... Tabuleiro->x[0]->y[f]

.

.

.

Tabuleiro->x[f]->[y0] ... Tabuleiro->x[f]->y[f]

Tabuleiro é uma struct que possui:

-Uma matriz de struct casa

-struct casa possui

-Uma lista de ameaçados que contém as peças ameaçadas pela peça da casa

-Uma lista de ameaçantes que contém as peças que ameaçam a peça da casa

-Ponteiro para o elemento contido na casa

## Peça

Peça é uma struct que possui

-Uma struct movimento

-Identificação

-Uma cor

-Quantidade de movimentos

-Verificados se ela pode ir pra trás ou naão

OBS:

Struct movimento te as coordenadas x e y

Seja k uma peça:

if k.cor

k.cor = (char\*)val;

if k.id

k.id = (char\*)val;

if k.qtdMov

k.qtdMov = (int\*)val;

if k.movPraTras

k.movPraTras = 0 || k.movPraTras = 1 ;

if k->movPeca->x

k->movPeca->x = (int\*) val;

0 < val < 7

if k->movPeca->y

k->movPeca->y = (int\*) val

0 < val < 7