**TRABALHO 3 MODULAR:**  
Caio Graça Melo  
Lucca Buffara  
Mark Ribeiro

**FUNÇÕES DE ACESSO:**  
**BAR:**  
BAR\_tpCondRet BAR\_CriarBAR( void ) ;

void BAR\_DestruirBAR( void ) ;

BAR\_tpCondRet BAR\_Inserir( char c, int n ) ;

BAR\_tpCondRet BAR\_Excluir( char c ) ;

BAR\_tpCondRet BAR\_NPecas( char c, int \*n ) ;

**DADO:**DAD\_tpCondRet DAD\_JogaDados(int \*d);

**DADO\_PONTOS:**DP\_tpCondRet DP\_criaDadosPontos(void);

DP\_tpCondRet DP\_dobraValor(char jogador);

DP\_tpCondRet DP\_modificaPontos(int valor, char jogador);

DP\_tpCondRet DP\_lerPontos(char jogador, int \*val);

DP\_tpCondRet DP\_lerValorDadosPontos(int \*val);

DP\_tpCondRet DP\_lerUltimoJogador(char \*cor);

DP\_tpCondRet DP\_destroiDadosPontos(void);

**PECA:**PEC\_tpCondRet PEC\_CriaPeca( Peca \*p, char cor );

PEC\_tpCondRet PEC\_ObterCorPeca ( Peca \*p, char \*cor );

PEC\_tpCondRet PEC\_DestroiPeca( Peca \*p );

**PFN:**PFN\_tpCondRet PFN\_CriarPFN( void ) ;

PFN\_tpCondRet PFN\_Inserir( char c, int n ) ;

PFN\_tpCondRet PFN\_NPecas( char c, int \*n ) ;

void PFN\_DestruirPFN( void ) ;

**TABULEIRO:**TAB\_tpCondRet TAB\_CriarTabuleiro( void ) ;

TAB\_tpCondRet TAB\_ArrumarTabuleiro( void );

void TAB\_DestruirTabuleiro( void ) ;

TAB\_tpCondRet TAB\_MoverPeca( int casaIni, int casaFim, char corJogador ) ;

TAB\_tpCondRet TAB\_InserirPecasCasa(int n, char cor, int nCasa ) ;

TAB\_tpCondRet TAB\_RemoverPeca( int nCasa ) ;

TAB\_tpCondRet TAB\_PrintTabuleiro( void ) ;

int TAB\_ChecaCasa (int nCasa, char c);

int TAB\_ChecaQuadrante (char c);

**LISTA:**LIS\_tppLista LIS\_CriarLista(void( \* ExcluirValor ) ( void \* pDado ) ) ;

void LIS\_DestruirLista( LIS\_tppLista pLista ) ;

void 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 ) ;

void \* LIS\_ObterValor( LIS\_tppLista pLista ) ;

LIS\_tpCondRet pegaNumElementos( LIS\_tppLista pLista ,  
 int \* num );

void IrInicioLista( LIS\_tppLista pLista ) ;

void IrFinalLista( LIS\_tppLista pLista ) ;

LIS\_tpCondRet LIS\_AvancarElementoCorrente( LIS\_tppLista pLista ,  
 int numElem );

LIS\_tpCondRet LIS\_ProcurarValor( LIS\_tppLista pLista , void \* pValor ) ;