

Documentação

Grupo:

- 1- Gabriel Heffer
- 2- Lucas Damo
- 3- Yuri Strack

Especificação de requisitos:

- Requisitos Funcionais:

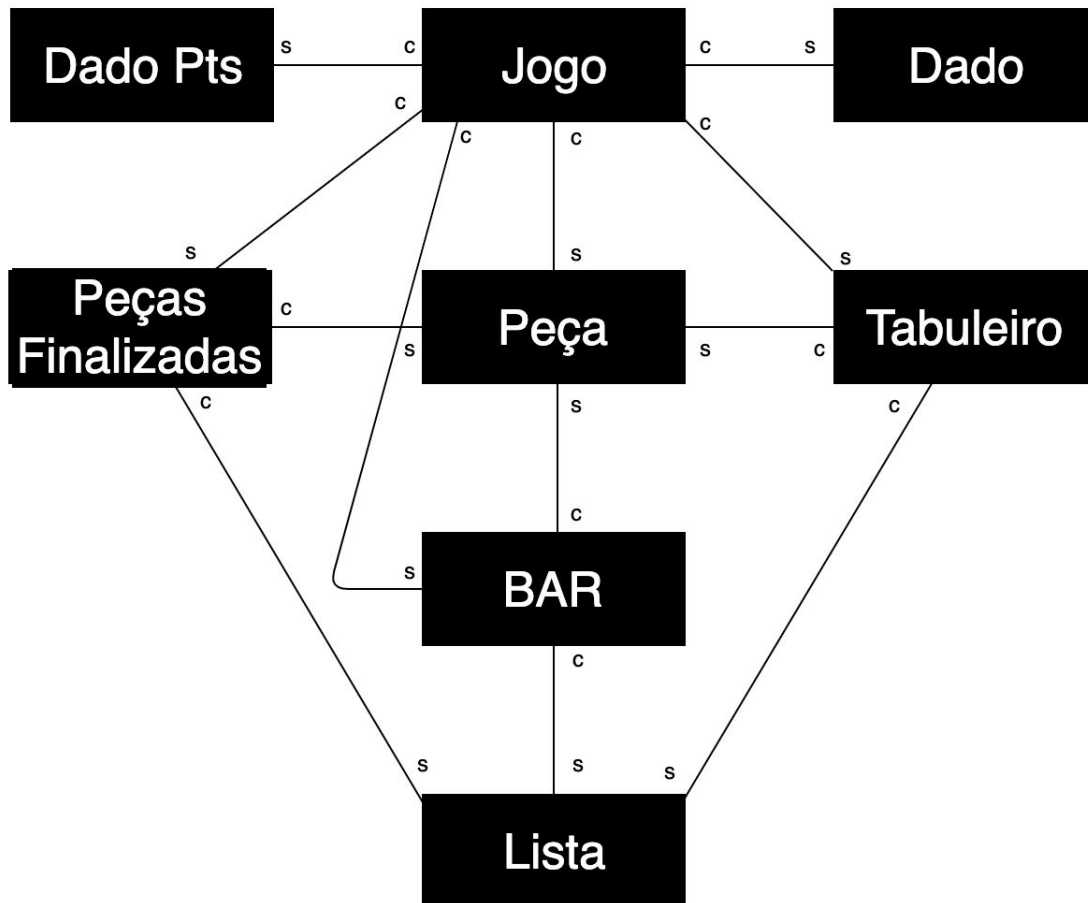
- Exibir menu em formato ASCII com opção para carregar partida ou começar uma nova partida.
- Criar tabuleiro com 24 casas (4 quadrantes com 6 casas cada, divididos em parte interna e externa).
- Barra para peças capturadas se localiza no centro do tabuleiro. Fazendo a divisão de tabuleiro externo (a esquerda) e interno (a direita).
- Tabuleiro interno do jogador de peças pretas é o lado direito superior, e do jogador de peças brancas é o lado direito inferior (cada com 6 casas).
- Tabuleiro externo também possui parte superior e inferior, cada com 6 casas.
- A interface do tabuleiro será em formato ASCII.
- É um jogo para dois jogadores.
- Criar 15 peças brancas e 15 peças pretas.
- Distribuir peças no tabuleiro na forma:
 - Para as brancas: 2 na casa 24; 5 na casa 13; 3 na casa 8 e 5 na casa 6.
 - Para as pretas: 2 na casa 1; 5 na casa 12; 3 na casa 17 e 5 na casa 19
- Possuir dois dados para o início da partida.
- Início da partida, cada jogador joga um dado para ver qual irá iniciar. Jogador com maior pontuação no dado inicia partida. No caso de empate, os dados serão jogados novamente.
- Possuir dado dobrador de pontos da partida.
- A cada rodada os dados são jogados pelos jogadores (em caso de valores iguais, ou dobras, nos dados, joga duas vezes os números mostrados nos dados).
- Movimentar as peças de acordo com os valores nos dados.
- Verificar se a movimentação é possível (se a casa está livre* ou não).
 - *Uma casa está livre quando há somente peças da cor do jogador ou uma peça da cor do adversário ou nenhuma peça.**
- Quando somente um dos números dos dados pode ser legalmente jogado, o jogador deverá jogar somente esse número.
- Quando nenhum dos números dos dados pode ser jogado, o jogador simplesmente não move nada e será a vez do oponente.
- No caso de dobras, se nem todos os 4 números podem ser jogados, o jogador deve jogar o máximo possível deles.

- Se uma peça for capturada, deve ser colocada na barra (Barra de peças capturadas).
- Uma peça só pode ser capturada se estiver sozinha em uma casa x e o oponente mover uma peça sua para a mesma casa x .
- Sempre que um jogador tem uma ou mais peças na barra, sua obrigação é entrar no tabuleiro interno do oponente.
- Para reentrar, a peça deve ser movida para uma casa livre no tabuleiro interno do oponente.
- Na reentrada de uma peça, a contagem dos números dos dados começa no início da casa interna do oponente.
- Após a reentrada da última peça capturada, qualquer número dos dados que não tenha sido deve ser usado para mover qualquer peça.
- Retirada de peças do tabuleiro, quando um dos jogadores possui todas suas peças na sua parte interna do tabuleiro.
- Ao ser iniciada, cada partida de Gamão estará valendo 1 ponto (ou seja, se a partida terminar com esse mesmo valor, o vencedor ganhará 1 ponto).
- Um jogador pode propor que a partida dobre o seu valor inicial e passe, então, a valer 2 pontos.
- O dado duplicador mostra quantos pontos a partida está valendo quando um dobre ou redobre é aceito.
- Um jogador pode dobrar somente quando é sua vez de jogar e antes que ele lance seus dados naquele turno.
- Se um jogador recusa um redobre, seu oponente ganha os pontos que a partida estava valendo antes do redobre recusado.
- Se o redobre é aceito, o jogador que aceitou o redobre passa a ser o novo dono do cubo duplicador e a partida continua.
- Um redobre só pode ser feito pelo jogador que possui o dado multiplicador.
- Se o jogador que a perdeu conseguiu fazer a retirada final de pelo menos uma de suas peças, o vencedor ganhará somente o valor mostrado no cubo duplicador.
- Se o perdedor não fez a retirada final de nenhuma de suas peças, o vencedor receberá o dobro dos pontos mostrados pelo cubo duplicador.
- Se o perdedor não retirou nenhuma de suas peças e ainda tem peças capturadas ou no tabuleiro interno do oponente, o vencedor recebe o triplo do valor mostrado pelo cubo duplicador.
- A partida poder ser salva em qualquer momento do jogo. Armazenando então todas as informações necessárias para que a partida seja retomada no exato momento em que parou.

- Requisitos não funcionais:

- O jogo deve ser compatível com sistemas operacionais Windows.
- Não exige instalação.

- Arquitetura do jogo:



Obs: C representa Cliente e S representa Servidor.

- Funções disponibilizadas em cada interface:

- **DADO:**
 - DAD_tpCondRet DAD_Jogar(int *valor1, int *valor2);
- **DADO PONTOS:**
 - DADP_tpCondRet DADP_CriarDadoPontos(void);
 - void DADP_DestruirDadoPontos (void);
 - DADP_tpCondRet DADP_IniciarDadoPontos (char jogador);
 - DADP_tpCondRet DADP_ObterValorPontuacao(int * pontuacao);
 - DADP_tpCondRet DADP_MudarJogadorPermitido(char jogador);
 - DADP_tpCondRet DADP_DobrarPontuacao(char jogador);

- DADP_DonoDado(char* dono);

- **PEÇA:**

- PCA_tpCondRet PCA_CriarPeca(tpPeca ** peca , int cor);
- void PCA_DestruirPeca (tpPeca * peca);
- PCA_tpCondRet PCA_ObterCor(tpPeca ** peca , int * cor);

- **TABULEIRO:**

- TAB_tpCondRet TAB_CriarTabuleiro (void);
- void TAB_DestruirTabuleiro(void);
- TAB_tpCondRet TAB_InicializarTabuleiro(void);
- TAB_tpCondRet TAB_ObterPeca(tpPeca **tppPeca);
- TAB_tpCondRet TAB_AndarTabuleiro(int numElem);
- TAB_tpCondRet TAB_IrInicioTabuleiro(void);
- TAB_tpCondRet TAB_IrFinalTabuleiro(void);
- TAB_tpCondRet TAB_InserirPeca(tpPeca *peca);
- TAB_tpCondRet TAB_ExcluirPeca(void);
- TAB_tpCondRet TAB_ObterQuantidadePEC(int * quantidade);

- **BAR:**

- BAR_tpCondRet BAR_CriarBarra(void);
- void BAR_DestruirBarra(void);
- BAR_tpCondRet BAR_InserirBarra(tpPeca * peca);
- BAR_tpCondRet BAR_QuantidadeBarra(char cor , int * quantidade);
- BAR_tpCondRet BAR_RetirarBarra(tpPeca ** peca , char cor);

- **PEÇAS FINALIZADAS:**

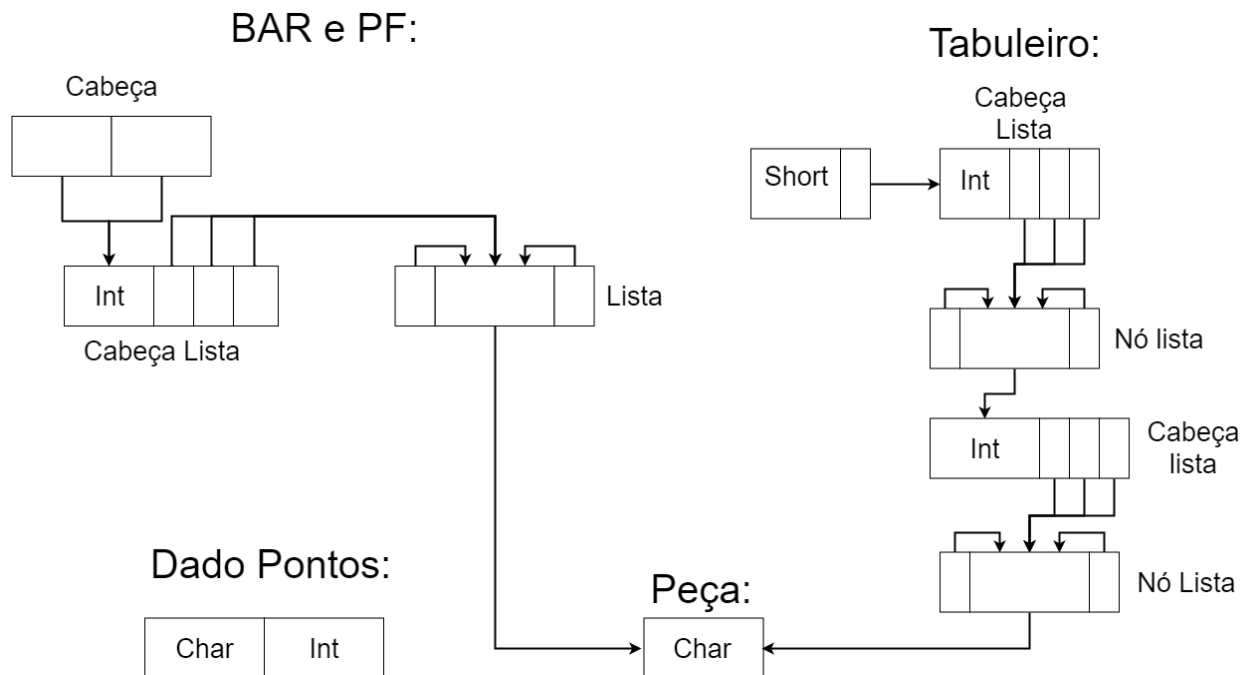
- PEF_tpCondRet PEF_CriarFinalizadas(void);
- void DestruirFinalizadas(void);
- PEF_tpCondRet PEF_InserirFinalizadas(tpPeca * peca);
- PEF_tpCondRet PEF_quantidadeFinalizadas(char cor , int * quantidade);

- **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 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) ;

- Modelo:



- Assertivas:

Bar:

- Quando houver peça (s) nessa estrutura, o jogador será obrigado a move-la para outro lugar antes de mexer em qualquer outra peça.
- Se não há peças no BAR: então cabeça do BAR são dois ponteiros NULL
- Se há peças pretas no BAR e nenhuma branca, então cabeça do BAR aponta para uma lista de peças pretas e o outro ponteiro é NULL
- Se há peças brancas no BAR e nenhuma peça preta, então cabeça do BAR aponta para uma lista de peças brancas e o outro ponteiro é NULL

Peças Finalizadas (PF):

- Uma vez inserida uma peça, essa jamais poderá ser removida antes do término do jogo.
- Se não há peças no PF: então cabeça do PF são dois ponteiros NULL
- Se há peças pretas no PF e nenhuma branca, então cabeça do PF aponta para uma lista de peças pretas e o outro ponteiro é NULL
- Se há peças brancas no PF e nenhuma peça preta, então cabeça do PF aponta para uma lista de peças brancas e o outro ponteiro é NULL

Dado pontos:

- Valor máximo que poderá assumir é 64
- Os valores do jogador só poderão ser 'P' ou 'B', indicando a cor das peças que eles controlam

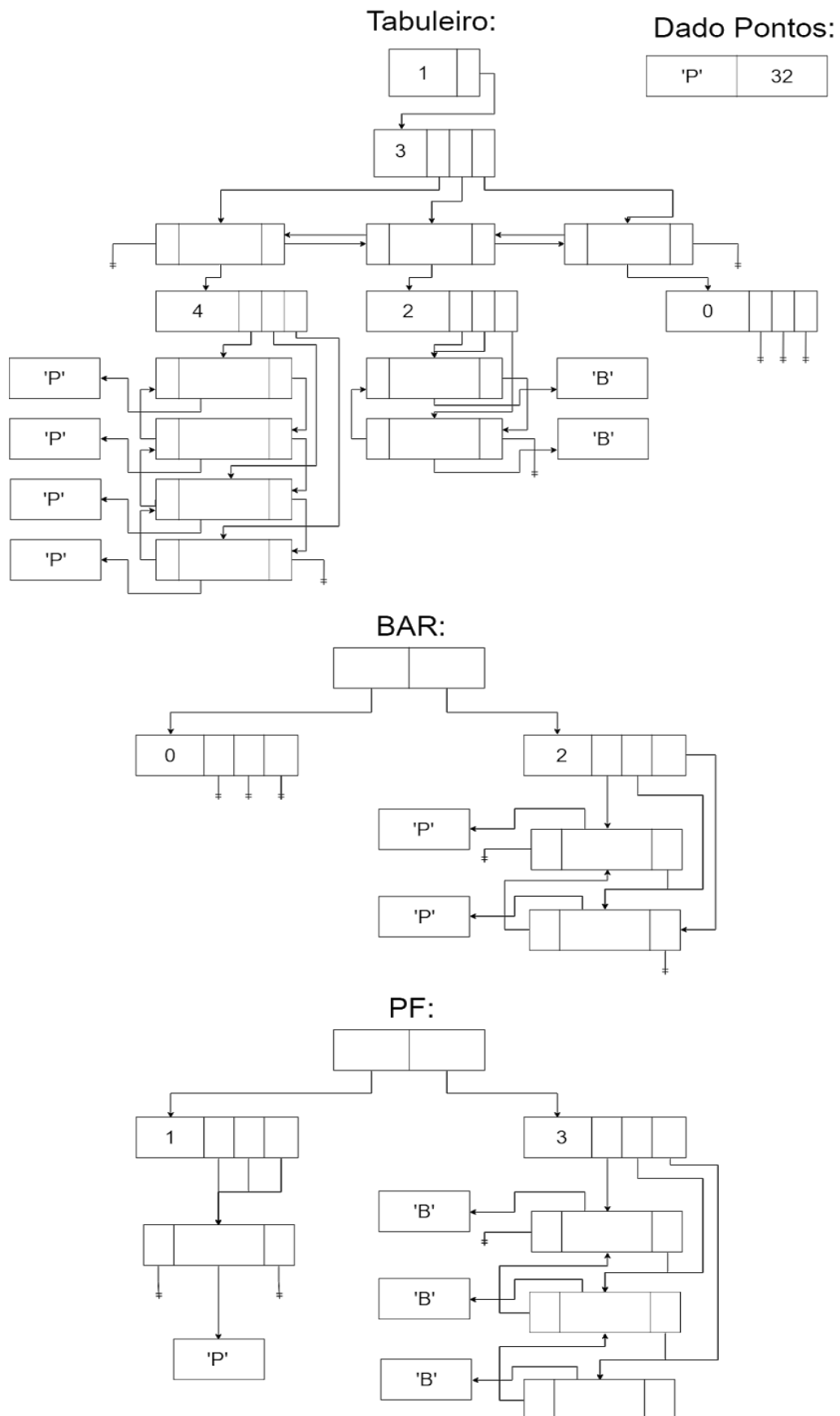
Peça:

- O valor de cada peça só poderá ser 'P' ou 'B' indicando assim sua cor, preta ou branca respectivamente.

Tabuleiro:

- Se o tabuleiro não tiver sido inicializado, então o campo inic será 0 e o ponteiro da cabeça aponta para uma lista vazia
- Caso tenha sido inicializado então o ponteiro da cabeça aponta para uma lista com 24 nós (representando as casas), cada nó apontando para uma outra lista (representando as peças que estão nas casas)

- Exemplo:



No exemplo acima, temos um tabuleiro com 3 casas, contendo 4 peças pretas na primeira casa, 2 brancas na segunda e nenhuma na terceira. O valor da partida é 32 e o jogador que poderá dobrar o valor é o que controla as peças pretas. Tem 2 peças pretas no BAR e nenhuma branca. O jogador que controla as peças brancas já conseguiu finalizar 3 de suas peças enquanto o outro jogador conseguiu apenas 1.