**PONTIFÍCIA UNIVERSIDADE CATÓLICA   
DO RIO DE JANEIRO**

**PROGRAMAÇÃO MODULAR – INF1301**

TRABALHO G1 – T2

Alunos: Bruno D’Almeida Franco

Leonardo Lages de Alencar

Luís Fernando Teixeira Bicalho

Sumário

[1) Especificação de Requisitos: 3](#_Toc449032763)

[2) Arquitetura do Programa: 5](#_Toc449032764)

[3) Modelo Estrutural: 6](#_Toc449032765)

# Especificação de Requisitos:

O programa fará uma simulação do jogo de cartas denominado “Truco”, feito para duas, quatro ou seis pessoas jogarem, com o objetivo do jogador, da dupla ou do trio, ou seja, o jogo só ocorre no escopo de um contra um, dois contra dois ou três contra três, totalizarem 12 ou mais pontos.

O jogo utiliza um baralho ítalo-espanhol, que possui 40 cartas no total, cada uma delas contendo o seu valor, que varia de 4 à 7, passando por Q (Dama), J (Valete), K(Rei) e A(Ás), voltando para os números 2 e 3, sendo esta a ordem de força entre elas, ou seja, indo da que vale menos até a que vale mais dentro do jogo, e um naipe, representado por Ouros, Espadas, Damas e Paus. Como pode ser visto, excluímos as cartas de valores 8 à 10 dos quatro naipes citados, bem como cartas promocionais e coringas.

O jogo em si, que chamaremos aqui especificamente de Partida é dividido em várias Mãos, que possuem até três Rodadas. Em cada Rodada o programa embaralha as cartas no baralho, entregando três delas do topo para cada jogador. Ao final da distribuição, uma carta é virada e exposta no centro da mesa. Esta chamada “Vira” que definirá as manilhas, que serão as numerações logo acima dela (no caso da “Vira” ter valor 6, as manilhas serão as cartas de cada um dos quatro naipes que possuam valor 7), consideradas as maiores cartas do jogo, ou seja, com valor logo acima do 3. Porém, se em uma rodada forem baixadas mais de uma manilha, estas são comparadas com valor de força através dos naipes, sendo a ordem, do mais fraco pro mais forte: Ouros, Espadas, Copas e Paus.

A divisão de equipes funciona com jogadores pares de um “lado” e jogadores ímpares do outro, considerando que os jogadores sejam enumerados. Em relação à sequência de quem joga, na primeira Rodada da primeira Mão começa jogando o jogador de número 1. Para as subsequentes rodadas da mesma mão, joga o ganhador da rodada anterior, se houver. Em caso de empate, repete-se o jogador da rodada anterior.

Para cada Rodada, o jogador deve baixar uma das cartas da sua mão, sendo que o ganhador desta será aquele que possuir o maior valor dentre os baixados à mesa, ganhando um ponto para a Mão. O grupo ganhador da Mão será aquele que for melhor de três, ou seja, dentre três Rodadas, aquele que ganhar duas será o vencedor, somando um ponto na Partida. Considerando casos de empate, temos várias possibilidades:

-Se a primeira rodada empatar, quem ganhar a segunda vence a Mão;  
 -Se a segunda rodada empatar, quem ganhou a primeira vence a Mão;  
 -Se a primeira e a segunda rodadas empatarem, vence quem vencer a terceira;  
 -Se a terceira empatar, vence quem ganhou a primeira;  
 -Se as três rodadas empatarem, passa-se para a próxima Mão, porém sem pontos ganhos pra nenhum dos grupos.

Em relação ao aumento na pontuação de cada Mão, temos quatro casos: Truco, Seis, Nove e Doze.Em caso de Truco, o jogador pede este aumento de aposta na hora em que for descer a carta escolhida. Se o adversário, ou seja, o próximo a jogar na sequência definida anteriormente, aceitar, a Mão começa a valer 3 pontos ao invés de 1, caso contrário, o jogador que pediu o Truco receberá 1 ponto na Mão. Para os casos de Seis, Nove e Doze, estes funcionam da mesma maneira que o Truco, com a diferença na pontuação que a Mão começa a valer se aceito (Mão passa a valer 6, 9 e 12 pontos, respectivamente) e se não aceito (Mão continua valendo 1 ponto, porém quem pede o Truco ganha, respectivamente, 3, 6 e 9 pontos). A única condição para tais aumentos é que eles devem vir em sequência, ou seja, o jogador do time adversário não pode pedir Seis, nem Nove, nem Doze se o do outro grupo já não tenha pedido Truco, Seis e Nove respectivamente, funcionando como uma sobre aposta.

Além disso, temos dois casos de Partidas excepcionais: Mão de Onze e Mão de Ferro. No primeiro caso, temos um dos grupos chegando a onze pontos na Partida, ganhando a habilidade (se for dupla ou trio) de visualizarem as cartas de seus parceiros de grupo. Se eles analisarem que não têm chance de ganhar, eles desistem da rodada e o grupo adversário ganha 1 ponto, passando, assim, para a próxima rodada; caso contrário, a mão valerá três pontos. No segundo caso, temos ambos os grupos chegando a onze pontos na Partida, levando esta a ser realizada com todas as cartas viradas para baixo, ou seja, o jogador escolhe uma das três cartas para ser baixada sem saber o respectivo valor de cada uma delas.

Requisitos não-funcionais:

.Todos os dados de entrada (exemplo: cartas baixadas) serão validados pelo jogo. Caso a entrada seja inválida, o jogador será informado e terá uma nova oportunidade de digitar.

.Todos os módulos devem ser testados individualmente, onde cada função dos módulos é testada em diferentes circunstâncias (utilizando o "Arcabouço" de teste automatizado)

.Para acelerar o processo de reutilização de projeto, implementação e teste, deve-se maximizar a reutilização de módulos

.Todas as funções e módulos deverão ser desenvolvidos utilizando padrões de documentação, garantido assim que o programa seja de fácil manutenção.

# Arquitetura do Programa:



# Modelo Físico:



\*\*Os tipos enumerados não possuem a mesma estrutura mostrada em sala, pois ocupa muito espaço, além de ser inviável representar arcos na plataforma de edição “draw.io”. Portanto, decidimos, por recomendação, substituir por “OU” em cada uma das opões do tipo enumerado representado

OBS.: A estrutura principal é uma cabeça de lista que percorre uma lista de listas, sendo que cada um dos elementos da lista pode representar jogadores (lista com 3 cartas cada), as cartas na mesa (lista com tantas cartas quanto foram descidas pelos jogadores), dentre outras, porém mantendo o padrão de lista de listas comumente com o tipo BAR\_tppCarta.

# Assertivas:

* Para todo \*pOrigemLista deverá existir um \*pAnt desse nó da lista apontando para NULL;
* Para todo \*pFimLista deverá existir um \*pProx desse nó da lista apontando para NULL;
* Todos os elementos de uma mesma lista, com o mesmo indexLista, devem aointar para a mesma cabeça de lista;
* O campo anterior do próximo elemento deve referenciar o elemento atual/corrente (\*pElemCorr), tanto na estrutura de lista quanto na estrutura provida pelo CESPDIN;
* Se um elemento do meio da lista for retirado, os ponteiros para o anterior e para próximo (\*pAnt e \*pProx) devem ser encadeados;
* Não se pode alterar as informações de um elemento que não é indicado como ativo, ou seja, que já foi liberado;
* Para cada elemento da lista (nó), deverá haver um ponteiro para a cabeça de lista, possuindo esta já um campo que aponta de volta para o elemento corrente (\*pElemCorr);
* Tipo de cabeça de lista deve ser compatível com o esperado;
* Tipo do elemento de lista deve ser compatível com o esperado;
* O espaço de controle de cada elemento da lista não pode ser ultrapassado;
* O espaço de controle da cabeça de lista não pode ser ultrapassado;