

# ESPECIFICAÇÃO DE REQUISITOS

## Projeto DADO MENTIROSO

*Versão 1.0*

*05/11/2015*

Versão	Autores	Data	Ação
1.0	Arthur da Silva Augusto M. Galego	05/11/2015	Estabelecimento de requisitos
2.0	Arthur da Silva Augusto M. Galego	10/12/2015	Estabelecimento de requisitos

### Conteúdo:

1. Introdução;
2. Visão geral;
3. Requisitos de software.
4. Esboço da interface gráfica.

## 1.0 Introdução

### 1.1 Objetivo

O objetivo é desenvolver um software com disputa multiplayer do jogo DADO MENTIROSO. Neste documento fica especificado a visão geral do jogo, assim como seus requisitos de software.

### 1.2 Referencias:

<<http://pt.wikihow.com/Jogar-Dado-Mentiroso>>

## 2.0 Visão geral

### 2.1 Arquitetura da aplicação

Cliente-servidor.

### 2.2 Regras do jogo

Introdução

Dado mentiroso é um jogo de aposta, baseado em turnos, onde de dois jogadores tentam adivinhar a quantidade de dados com a face X virada para cima da mesa, apenas olhando os seus próprios dados.

### Objetivo

O objetivo do jogo é acertar a acusação. A acusação é feita após a aposta do inimigo. Se o jogador acusar, e seu adversário tiver feito uma aposta equivocada, o acusador ganha. Se a aposta tiver sido válida, o acusador perde.

### Funcionamento

Cada jogador recebe um copo com cinco dados cada, todos os batem na mesa com a boca para baixo de modo que ninguém veja como estão os dados. O primeiro jogador é decidido aleatoriamente, ele olha apenas os seus dados e faz uma aposta de quantos números virados para cima existem na mesa inteira.

*Ex.: dados do jogador 1 (2, 2, 4, 5, 5).*

*Jogador 1 aposta que tem três 2 virados para cima.*

O jogador seguinte tem duas opções, ou aumentar a aposta (sempre com números maiores que a aposta anterior), ou acusar o jogador anterior de estar mentindo. Caso a acusação feita seja verdadeira, o jogador que acusou ganha.

*Ex.: dados do jogador 2 (2, 3, 3, 4, 6)*

*Opções do jogador 2:*

- *acusar o jogador 1 de mentiroso (ele perderia pois realmente existem três dois virados pra cima)*
- *fazer uma aposta maior (apostar que tem quatro dois virados para cima por exemplo, ou que existem três 3 virados para cima)*

Após fazer a aposta o jogador perde automaticamente o turno e o jogo segue para o outro jogador.

## **2.3 Premissas de desenvolvimento**

- \* Deve apresentar interface gráfica bidimensional.
- \* Deve ser implementado em linguagem java.
- \* Deve ser executado em qualquer plataforma que disponha da máquina virtual java.
- \* Deve utilizar o framework NetGames.

### 3.0 Requisitos da aplicação

#### 3.1 Requisitos funcionais

**Requisito funcional 1 - Escolher o nome:** Deve ser disponibilizado ao usuário uma interface para informar o nome do jogador.

**Requisito funcional 2 - Escolher aposta:** Deve ser disponibilizado ao usuário uma interface para informar a quantidade de dados da aposta.

**Requisito funcional 3 - Começar partida:** Ao começar uma nova partida, o programa deve apresentar a tela principal com os dados, e as opções de iniciar partida, apostar, acusar, desconectar, bem como a interface grafica ja inicializada.

**Requisito funcional 4 - Procedimentos de jogada:** A cada turno, deve ser disponibilizado ao jogador atual a possíveis ações que este pode tomar (acusar ou apostar), com algumas restrições:

- O programa deve permitir apenas lances válidos.
- No final da partida o programa deverá indicar o vencedor da mesma.

**Requisito funcional 5 - Decidir vencedor:** A aplicação deve ser capaz de verificar se há um vencedor a cada turno finalizado conforme regras do jogo. Se houver, é decretado o fim da partida.

#### 3.2 Requisitos não funcionais

**Requisito não funcional 1 - Especificação do projeto:** Deve ser produzida especificação do projeto baseada em UML 2.

**Requisito não funcional 2 - Interface gráfica com usuário:** O software deverá apresentar interface gráfica bidimensional, uma interface para cada jogador. Esta interface é rotativa de acordo com o jogador atual, e deve mostrar a última aposta feita.

**Requisito não funcional 3 - Tecnologia da interface gráfica:** A interface gráfica será baseada em JAVA SWING.

**Requisito não funcional 4 - Arquitetura distribuída:** O programa deve rodar num modelo cliente-servidor, de forma distribuída.

4.0 Esboço da interface gráfica

