

Aplicação: Jogo Duvido(Bullshit)

# Especificação de Requisitos de Software

Versão 1.0

## Histórico do documento

Versão	Autor(es)	Data	Ação
1.0	Matheus Gomes Samuel Landre	09/09/2019	Elaboração inicial da especificação de requisitos
1.1	Matheus Gomes Samuel Landre	08/12/2019	Adição do esboço da interface gráfica do programa

Conteúdo:

1. Introdução
2. Visão geral do Sistema
3. Requisitos da aplicação
4. Esboço da interface

## **1. Introdução**

### **1.1. Objetivos do desenvolvimento**

Desenvolver um um programa multijogador do jogo de cartas Duvido. As partidas do jogo serão disputadas em rede entre três jogadores.

### **1.2. Definições**

#### **1.2.1. Regras do jogo**

- O jogo inicia com um baralho de cartas tradicional, sem coringas, sendo embaralhado e distribuído entre os jogadores.
- Sorteia-se entre quem tem mais cartas na mão para iniciar a partida. Este deve jogar de um a quatro ases virados para baixo no monte e dizer o que está jogando. O jogo continua no sentido horário. O próximo jogador deve jogar de uma a quatro cartas do próximo valor, viradas para baixo e dizendo o que jogou. E assim por diante.
- Sempre que um jogador diz o que está jogando, esta declaração deve seguir as regras: de uma a quatro cartas e todas as cartas do mesmo valor, que é subsequente ao valor da rodada anterior.
- Ao contrário da declaração, as cartas que o jogador joga no monte não necessariamente precisam estar de acordo com o que foi declarado, uma vez que os outros jogadores não veem as mesmas.
- Sempre que algum jogador declara o que está jogando, os outros jogadores da mesa tem a opção de duvidar da jogada. Neste caso, inicia-se um desafio.
- Em um desafio inspeciona-se as cartas que foram jogadas. Caso elas estejam de acordo com o que for declarado o desafiante fica com todas as cartas do monte. Caso contrário, quem jogou as cartas fica com o monte. Após um desafio inicia-se outra rodada pelo início.

- O jogo acaba quando algum jogador ficar sem cartas na mão, este jogador é o vencedor.
- As rodadas seguem a seguinte ordem de valores do baralho: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K. Ao chegar no rei(K), o próximo valor será o ás(A).

### **1.3. Referências**

<https://www.pagat.com/beating/cheat.html> - Histórico do jogo e como jogar (inglês).

<https://pt.wikihow.com/Jogar-Duvido> - Como jogar em português. As regras não são exatamente as mesmas da aplicação a ser desenvolvida.

## **2. Visão geral do sistema**

### **2.1. Arquitetura da aplicação**

Programa orientado a objetos, distribuído no formato cliente servidor. Deve ser escrito na linguagem Java.

### **2.2. Premissas de desenvolvimento**

Deve apresentar uma interface bidimensional.

Deve utilizar o framework “NetGamesNRT”, de modo que realize a interação usuário *versus* usuário.

Deve ser implementado em Java, executando em qualquer plataforma que possua Java Virtual Machine.

## **3. Requisitos da aplicação**

### **3.1. Requisitos funcionais**

#### **3.1.1. Conectar**

A aplicação deve disponibilizar uma opção de menu para que o usuário possa conectar-se com o servidor.

#### **3.1.2. Iniciar partida**

Deve estar disponível a opção de iniciar partida através de uma opção de menu. Esta ação está condicionada a uma conexão ao servidor e o número mínimo de jogadores atingido.

#### **3.1.3. Distribuição de cartas**

A aplicação deve ser capaz de distribuir as cartas devidamente embaralhadas entre os jogadores presentes na mesa.

#### **3.1.4. Realizar jogada**

O jogador deve ser capaz de jogar as cartas que escolher no monte e realizar a declaração correspondente. A declaração limita-se a selecionar a quantidade de cartas do valor que deve ser jogado na rodada.

#### **3.1.5. Duvidar**

Após a jogada de um adversário o jogador deve ser capaz de desafiar a veracidade da mesma. Como consequência, o monte da mesa irá para a mão do jogador que perder o desafio, seguindo as regras do jogo.

#### **3.1.6. Desconectar**

A aplicação deve disponibilizar uma opção de menu para que o usuário possa desconectar-se do servidor.

### **3.2. Requisitos não funcionais**

#### **3.2.1. Especificação do projeto**

Código em linguagem Java, com especificação de projeto baseada na UML 2;

#### **3.2.2. Interface gráfica:**

Deverá possuir uma interface única que é compartilhada entre os usuários;

#### **3.2.3. Conectividade:**

Deverá suportar conexão à rede através da arquitetura cliente/servidor, usando o framework NetGamesNRT;

#### 3.2.4. Tecnologia da interface gráfica:

A Interface gráfica será baseada em *Java-Swing*;

#### 3.2.5. Padronização do código:

Fazer uso de lowerCamelCase (iniciados por letras minúsculas) na definição de variáveis e métodos, e utilizar UpperCamelCase (iniciados por letras maiúsculas) para a definição de Classes;

#### 3.2.6. Internacionalização:

O jogo deve estar disponível em Português;

### 4. Esboço da interface

