

# Especificação de Requisitos de Software

## Projeto Pazaak

Versão: 1.0

Data: 13/04/2015

Versão	Autores	Data	Ação
1.0	Diego Marzarotto , Elivelton Delfino, Carlos Alberto Sousa	13/04/2015	Estabelecimento dos Requisitos

### Conteúdo:

1. Introdução
2. Visão Geral
3. Requisitos de Software
4. Esboço da Interface Gráfica

## 1 Introdução

### 1.1 Objetivo:

Desenvolvimento de um programa que suporte disputas em rede na modalidade usuário *versus* usuário.

### 1.2 Referência/Regras:

O manual (em inglês) pode ser encontrado no link:

<http://www.pazaak.org/manual>

Um pequeno tutorial em vídeo mostrando o jogo em ação pode ser visto em: \_

[https://www.youtube.com/watch?v=pG\\_23DY0YNg](https://www.youtube.com/watch?v=pG_23DY0YNg)

## 2 Visão Geral

### 2.1 Arquitetura do Programa:

- Programa orientado a objetos, distribuído e escrito na linguagem Java;
- Aplicação distribuída, cliente-servidor;

### 2.2 Descrição do jogo

O objetivo do Pazaak é vencer 3 de 3 turnos(set). Para vencer um turno deve-se ter na mesa cartões com pontuação total maior do que os cartões do adversário, sem exceder um total de 20, no caso de empate, nenhum dos jogadores leva o turno.

No início do jogo cada jogador começa com 4 cartas na mão selecionadas aleatoriamente das 10 cartas escolhidas das 32 cartas disponíveis. Na vez de cada jogador é baixado para a mesa uma carta do baralho principal, que se soma à sua pontuação atual, frente a isso, o jogador tem a opção de baixar uma carta de sua mão, para ajustar a pontuação final para o mais próximo possível de 20, passar a vez ou desistir.

### **2.3 Premissas de Desenvolvimento**

- O programa deve apresentar uma interface bidimensional;
- O programa deve utilizar o framework “NetGamesNRT” para realizar a interação usuário *versus* usuário;
- O programa deve ser implementado em Java, executando em qualquer plataforma que disponha da máquina virtual Java.

## **3 Requisitos de Software**

### **3.1 Requisitos Funcionais:**

#### **Requisito Funcional 1 - Conectar:**

O programa deve ter a opção de menu “Conectar” para estabelecer uma nova conexão com o servidor, nesta operação é definido a identificação dos jogadores. O sistema confirmará se a conexão foi estabelecida ou não;

#### **Requisito funcional 2 – Iniciar:**

O programa deve apresentar a opção de menu “Iniciar” para início de uma nova partida. Considera-se a seguinte restrição:

- Para iniciar uma nova partida é obrigatório estabelecer uma conexão com o servidor.

#### **Requisito funcional 3 – Desconectar:**

O jogo deve apresentar a opção de menu “Desconectar”, permitindo se desconectar do servidor, encerrando assim uma possível partida em andamento.

#### **Requisito funcional 4 – Jogo:**

O programa deve permitir que sejam disputadas partidas entre dois usuários, garantindo que os jogadores não possam executar jogadas inválidas de acordo com a regra do jogo (ver referência), e que no final tenha somente um vencedor;

### **Requisito funcional 5 – Baralho:**

O jogo deve ter um baralho composto por 4 cartas para cada valor (positivos e negativos) entre 1 e 4. Totalizando 32 cartas.

### **Requisito funcional 6 – Baralho auxiliar:**

O baralho auxiliar contém 10 cartas. Cada jogador terá um baralho auxiliar independente das 32 cartas disponíveis. Quatro cartas são escolhidos aleatoriamente quando a partida começa. Essas quatro cartas ficam na mão do jogador e estão disponíveis em toda a partida . Cada carta da mão pode ser jogado apenas uma vez.

### **Requisito funcional 7 - Partida, Turno e Rodada:**

#### **Partida:**

Uma partida é composta de um turno.

#### **Rodada:**

Uma rodada é o evento aonde os dois jogadores executam um procedimento de lance. Uma rodada começa pela ação do jogador definido como responsável pelo início da partida, e só termina com uma ação do outro jogador.

#### **Turno:**

Um turno é composto de um conjunto de rodadas. Um turno acaba quando:

- Quando for atingido um dos requisitos de vitória (requisito 9)

### **Requisito funcional 8 - Procedimento de lance:**

#### **Primeira rodada:**

Na primeira rodada de cada turno é colocada na mesa uma carta do baralho principal. Depois:

1. O jogador pode jogar uma de suas cartas da mão.
2. O jogador pode passar a sua vez ou jogador pode optar por não fazer nada. Ao selecionar esta opção, até o final do turno o jogador não tem permissão para executar mais nenhuma ação. Ficará apenas assistindo o outro jogador mantendo a pontuação que tiver na mesa.
3. É a vez do outro jogador.

O outro jogador segue exatamente os mesmos passos do primeiro jogador.

### **Demais rodadas: o jogo procede da seguinte maneira:**

1. Se o jogador não escolheu a opção 2 (não fazer nada) é baixada uma carta do baralho principal e a mesma é colocada na mesa.
2. Pode optar por não fazer nada. Ao selecionar esta opção, até o final do turno o jogador não tem permissão para executar mais nenhuma ação. Ficar apenas assistindo o outro jogador, mantendo a pontuação que tiver na mesa.
3. O jogador pode jogar uma de suas cartas da mão se não escolheu a opção 2 na rodada anterior.
4. O jogador pode passar a vez se não escolheu a opção 2 na rodada anterior.
5. É a vez do outro jogador.

### **Requisito funcional 9 - Vitória:**

#### **Vitória em turno:**

Ao final de cada rodada o sistema deve conferir:

1. Se um dos jogadores tem uma pontuação de mais de 20, o adversário é o vencedor.
2. Se ambos os jogadores tiverem, no final do turno, uma pontuação menor que 20 aquele que tiver mais próximo de 20 ganha o turno.
3. Se ao final de um turno a pontuação dos dois jogadores são iguais. Ninguém ganha o turno.
4. Se um dos jogadores ocupar os seus 9 espaços na mesa e tiver somatório igual ou inferior a 20, ele vence o turno.

#### **Vitória na partida:**

O jogador que ganhar 1 turno vence a partida.

**Requisito Funcional 10 - Informação do estado da partida:** a cada tentativa de procedimento de lance dos usuários o novo estado da partida deve ser exibido na interface, considerando as seguintes restrições:

- A matriz de posições deve ser atualizada com a com a adição correta das cartas de acordo com a jogada correspondente;

**Requisito funcional 11 – Sair:** o jogo deve apresentar a opção de menu “sair”, permitindo o encerramento da aplicação

### **3.2 - Requisitos Não Funcionais:**

**Requisito Não Funcional 1 – Especificação de projeto:** Código em linguagem Java e

especificação de projeto baseada em UML 2;

**Requisito Não Funcional 2 – Interface gráfica para usuário:** A aplicação deverá ter interface gráfica única, compartilhada pelos usuários;

**Requisito não funcional 3 – Cor das cartas:** as cartas baixadas a mesas pela aplicação deverão ser representadas pela cor amarela e as dos jogadores pela cor azul.

**Requisito Não Funcional 4 - Tecnologia de interface gráfica para usuário:** Interface gráfica baseada em Java-Swing.

#### 4 - Esboço da interface gráfica



Retirado do Pazaak Tutorial