



Universidade Federal de Santa Catarina  
Centro Tecnológico  
Bacharelado de Sistemas de Informação  
INE 5608 — Análise e Projeto de Sistemas

# **Especificação de Requisitos Jogo Jeopardy!!**

Alunos:

Barbara Santos  
Nataniel Pereira Borges

Florianópolis, 26 de fevereiro de 2013

**Especificação de Requisitos de Software**

Versão 1. 2

## Histórico do Documento

Versão	Autores	Data	Ação
1.0	Barbara Santos Nataniel Pereira Borges Eduardo Massao Kobayashi	30/09/2012	Estabelecimento dos requisitos de software.
1.1	Barbara Santos Nataniel Pereira Borges	22/12/2012	Correções de definição da arquitetura do programa e alterações nas definições de requisitos de software.
1.2	Barbara Santos Nataniel Pereira Borges	26/02/2012	Correções histórico do documento.

### Conteúdo:

1. Introdução;
2. Visão geral;
3. Requisitos de software.

## 1. Introdução

**Objetivo:** Desenvolvimento de um programa que suporte a disputa de partidas em rede do jogo denominado Jeopardy!, na modalidade usuário contra usuário, com uma adaptação para o tabuleiro. O jogo permitirá disputas entre dois jogadores em rede. Este trabalho faz parte da avaliação final da disciplina Análise e Projeto de Sistemas (INE5608) do curso de Sistemas de Informação – Universidade Federal de Santa Catarina (UFSC), ministrada pelo Professor Ricardo Pereira e Silva.

### Referências:

< <http://www.Jeopardy!.com>>  
<<http://en.wikipedia.org/wiki/Jeopardy!!>>

## 2. Visão Geral

**Arquitetura do programa:** Programa orientado a objetos escrito na linguagem Java. Utiliza-se do servidor NetGames para conexão remota, permitindo que dois clientes em computadores diferentes possam jogar em rede.

**Descrição do Jogo:** O jogo é baseado em perguntas e respostas que versam sobre temas de história, geografia, literatura, atualidades e esportes. O jogador deve cruzar o tabuleiro de 12 casas antes de seu adversário para ser declarado vencedor. Para isto, deve responder corretamente às pistas que seleciona. As pistas serão agrupadas nas

categorias acima citadas e terão valores 1, 2 e 3, conforme sua dificuldade. Este valor representa o número de casas que o jogador avançará caso responda corretamente. A dificuldade é crescente da esquerda para a direita.

Todas as respostas devem ser formuladas em forma de pergunta e devem iniciar invariavelmente com os seguintes padrões: “Quem é”, “Quem foi”, “O que é”, “O que foi” ou “Quando foi”. O padrão é correspondente aquele fornecido pela pista. Por exemplo, ao selecionar determinada pista de história, o jogador pode ser confrontado com a seguinte resposta: “Foi o matemático inventor da máquina de estados”. À qual deverá responder com a pergunta: “Quem foi Allan Turing?”.

Ao responder incorretamente uma pista pela primeira vez, terá a oportunidade de visualizar uma dica, a qual explicitará a primeira letra do cerne da resposta e a quantidade de letras que possui. Ao aceitar a dica o jogador arriscará o valor da pista em posições do tabuleiro, as quais serão descontadas de sua posição atual caso erre. Se estiver na primeira casa, não se moverá.

#### **Premissas de desenvolvimento:**

- Interface gráfica amigável e intuitiva;
- Aspecto visual 2D;
- Deve ser implementado em Java, executando em qualquer plataforma que disponha de JVM;
- Deve suportar 2 usuários em rede.
- Deve utilizar o framework “NetGames” para abstrair a camada cliente-servidor.

### **3. Requisitos de Software**

**1. Conectar:** A aplicação deve permitir que o usuário, através da interface, se conecte a um servidor online, escolhendo ainda um apelido.

**2. Fechar:** A aplicação deve se desconectar do servidor e ser fechada perante solicitação do usuário.

**3. Iniciar:** Deve ter uma opção conectar, que conecta o usuário ao servidor e quando houver outro jogador disponível inicia o jogo.

**4. Estabelecimento de quem inicia:** Inicialmente de forma aleatória, depois haverá intercalação entre os jogadores.

**5. Procedimento de seleção de pistas:** O programa deve suportar o procedimento de permitir que cada usuário selecione uma pista por vez de forma alternada. Deve desabilitar a seleção de pistas que já foram respondidas de forma correta.

**6. Procedimento de inserção de respostas:** O programa deve suportar o procedimento de resposta dos usuários por meio de turnos através de entrada escrita.

**7. Procedimento de verificação de respostas:** Toda resposta fornecida pelo usuário deverá passar por verificação de acerto por comparação textual com a resposta oficial. O

programa deve buscar inferir que as respostas do usuário estão corretas, ainda que não o sejam completamente. Isto se dá por tolerância limitada a 5 erros de digitação, ignorando caracteres especiais e sendo indiferente à caixa de texto e espaços.

**8. Procedimento de posicionamento no tabuleiro:** Ao receber uma resposta correta, o programa deve posicionar o jogador no tabuleiro conforme o valor da pista selecionada. Ao receber uma resposta incorreta, caso o jogador esteja jogando com dica e não esteja ocupando a posição inicial, deve retroceder no tabuleiro conforme o valor da pista selecionada.

**9. Verificar quem venceu:** A qualquer momento do jogo, caso um dos participantes consiga cruzar o tabuleiro, a partida deve chegar ao fim e este jogador será declarado vencedor. O oponente será declarado perdedor.

### 3.2 Requisitos Não-Funcionais

**1. Codificação:** O jogo deve ser desenvolvido utilizando a linguagem Java, permitindo sua execução em diversas plataformas;

**2. Ferramentas:** O ambiente de desenvolvimento NetBeans será utilizado para escrever os códigos fonte da aplicação. Já para a modelagem em UML será utilizado VisualParadigm.

**3. Especificação de projeto:** Deve ser utilizada UML versão 2.0.

**4. Interface gráfica para usuário:** O programa deverá ter interface gráfica semelhante para todos os usuários, mostrando o tabuleiro com a disposição dos jogadores e um quadro com as pistas, selecionáveis ou já desabilitadas. Haverá ainda janelas de diálogo eventuais, para exibição da pista selecionada e inserção de resposta, bem como para avisos.

**5. Símbolos dos jogadores:** Os nomes dos jogadores devem estar presentes em suas respectivas posições no tabuleiro.

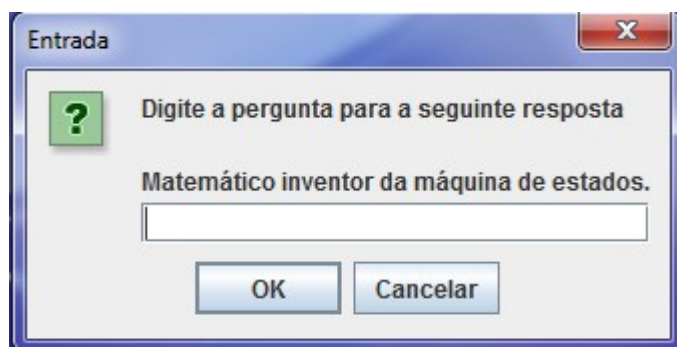
**7. Tecnologia de interface gráfica para usuário:** A interface gráfica deve ser implementada utilizando a biblioteca *swing* do java.

**8. Tecnologia de informação distribuída:** O sistema será implementado utilizando-se o recurso NetGames, permitindo que o usuário conecte-se a um servidor e interaja com outro usuário que também esteja conectado no servidor.

#### 4. Esboço da interface



*Quadro de pistas com temas. Cada ocorrência de tema possui valor diferente, ordenados da esquerda para a direita em ordem crescente de dificuldade e valor.*



*Janela de diálogo para texto de pistas e inserção de respostas. Surge após uma pista ser selecionada. A mesma janela é usada para pista com dica, acrescentando-se apenas o texto da dica.*



*Esboço de tabuleiro com disposição dos jogadores.*