



UNIVERSIDADE FEDERAL DE SANTA CATARINA

INE5417 - ENGENHARIA DE SOFTWARE I

# Especificação de Requisitos de Software

*Membros do Grupo:*

Hélcio Valentim de Andrade Neto (23100476)

João Gabriel Feres (23100454)

Eduardo Wallner Giacomelli (23100752)

Florianópolis

13/04/2025

# HISTÓRICO DE VERSÕES:

| Versão | Autores   | Data       | Ação                                |
|--------|---|------------|-------------------------------------|
| 1.0    | Hélcio Valentim de Andrade Neto, João Gabriel Feres, Eduardo Wallner Giacomelli | 13/04/2025 | Estabelecimento dos Requisitos      |
| 2.0    | Hélcio Valentim de Andrade Neto, João Gabriel Feres, Eduardo Wallner Giacomelli | 12/05/2025 | Correções nos requisitos funcionais |

## Conteúdo

|   |  |          |
|---|--|----------|
| <b>1</b>  | <b>Introdução</b>                      | <b>2</b> |
| 1.1   | Objetivo . . . . .                     | 2        |
| 1.2   | Definições e Abreviaturas . . . . .    | 2        |
| 1.3   | Referências . . . . .                  | 2        |
| <b>2</b>  | <b>Visão Geral</b>                     | <b>3</b> |
| 2.1   | Arquitetura do Programa . . . . .      | 3        |
| 2.2   | Premissas de desenvolvimento . . . . . | 3        |
| <b>3</b>  | <b>Requisitos de Software</b>          | <b>3</b> |
| 3.1   | Requisitos Funcionais . . . . .        | 3        |
| 3.2   | Requisitos Não-funcionais: . . . . .   | 7        |
| <b>Apêndice A - Qyshinsu : Informações sobre o Jogo</b> |  | <b>9</b> |

# 1 Introdução

## 1.1 Objetivo

O projeto almeja desenvolver um programa distribuído, que possibilite a disputa de partidas do jogo Qyshinsu entre dois usuários.

## 1.2 Definições e Abreviaturas

- Regras do jogo: ver apêndice A.
- Inventário : Área que contém as pedras do jogador. Cada jogador possui seu próprio inventário.
- Tema: Configuração estética para representar os elementos visuais do jogo.

## 1.3 Referências

Apresentação das regras do jogo (vídeo do canal Vem Ka Jogar):

<https://www.youtube.com/watch?v=UTdt1gGWls0>

## 2 Visão Geral

### 2.1 Arquitetura do Programa

Cliente-servidor distribuído.

### 2.2 Premissas de desenvolvimento

- O programa deve ser implementado em Python;
- O programa deve utilizar a plataforma D.O.G. como suporte para execução distribuída;
- O projeto deve possuir documentação e especificações baseadas em UML (segunda versão).

## 3 Requisitos de Software

### 3.1 Requisitos Funcionais

#### I - Requisito Funcional 1 (Inicialização do Programa):

O programa deverá iniciar a sua execução em uma tela de 'Menu Principal', que possibilite que o usuário inicie uma nova partida ou finalize a execução do programa. Imediatamente após entrar nessa tela, o programa deverá solicitar o nome do jogador. Em seguida, o programa deverá tentar estabelecer uma conexão com a plataforma D.O.G. (servidor). Nesse momento, o programa deve estar preparado para lidar com duas situações: sucesso ou insucesso da tentativa de conexão.

- a-) *Conexão mal-sucedida:* em caso de uma falha na tentativa de conexão com o D.O.G. Server, o programa deve notificar o usuário sobre o ocorrido ('conexão falhou. Abortando...') e encerrar sua execução.
- b-) *Conexão bem-sucedida:* caso o programa obtenha êxito em sua tentativa de conexão com o D.O.G. Server, o usuário deverá ser notificado sobre o ocorrido ('Conectado a Dog Server.').

## **II - Requisito Funcional 2 (Iniciar uma Partida):**

O programa deverá apresentar, no Menu Principal, um botão que permita ao usuário iniciar uma partida. Após receber ordens do usuário que indiquem o desejo do mesmo de iniciar uma nova partida, deve ser enviada uma solicitação de início de partida ao D.O.G. Server. Dois casos devem ser tratados:

- a-) *Inicialização mal-sucedida:* em caso de resultado negativo da solicitação de início feita ao D.O.G. Server, o programa deve notificar o usuário sobre o ocorrido ('falha na inicialização.'), e retorná-lo para a interface de 'Menu Inicial'.
- b-) *Inicialização bem-sucedida:* caso o programa obtenha êxito em sua tentativa de inicialização da partida feita ao D.O.G. Server, o usuário deverá ser notificado sobre o ocorrido ('inicialização bem-sucedida!'). Em seguida, o usuário deve ser redirecionado para a interface principal da partida (tela de jogo), já podendo executar o seu lance (caso seja o seu turno). Esta funcionalidade só deve estar habilitada se o programa estiver em seu estado inicial, isto é, sem partida em andamento e com o tabuleiro em seu estado inicial;

**III - Requisito Funcional 3 (Retornar para o Menu Inicial):** O programa deve apresentar um botão com os dizeres "Voltar para o Menu Inicial", oferecendo ao usuário uma maneira de retornar à interface de início. Só deverá ser possível utilizar essa funcionalidade caso o programa tenha acabado de finalizar uma partida;

**IV - Requisito Funcional 4 (Selecionar uma pedra):** De acordo com as regras do jogo (ver Apêndice A), a primeira jogada sempre consiste na inserção de uma pedra no tabuleiro. Para que jogadas desse tipo possam ocorrer, deverá ser possível que o usuário selecione uma das pedras em seu inventário (ver seção 1.2), o que justifica a existência desse requisito funcional. O programa deve possibilitar ao usuário a opção de alterar a sua escolha de pedra, o que se dará através de um novo clique em seu inventário.

**V - Requisito Funcional 5 (Selecionar uma casa):** O programa deve possibilitar que o usuário escolha uma casa, interagindo com uma posição do tabuleiro através de um clique. Se a casa selecionada estiver ocupada por uma peça do adversário, a jogada deverá ser considerada inválida. Se a casa selecionada estiver vazia e o jogador local não possuir uma pedra selecionada, a jogada também deverá ser considerada inválida. Caso o jogador tenha selecionado uma jogada inválida, o mesmo deverá ser notificado pelo programa sobre o caráter irregular do lance. Em seguida, o programa deve permitir ao jogador que ele escolha uma nova posição no tabuleiro ou selecione uma pedra. Caso a jogada seja válida, existem duas opções a serem consideradas:

- a-) *Se a casa selecionada estiver ocupada por uma peça do jogador local:* deve ser efetuada a remoção de sua peça que se encontra na posição escolhida.
- b-) *Se a casa selecionada estiver vazia:* o programa deve efetuar a inserção da pedra selecionada anteriormente (vide R.F. 4) na casa selecionada no tabuleiro.

Após a seleção de uma casa válida e a subsequente efetuação da jogada adequada, o tabuleiro deve ser atualizado. Em seguida, o programa deve avaliar se a partida está acabada por meio da simulação da captura de casas válidas do adversário. Se, na simulação, houverem casas válidas, o programa deve enviar a jogada ao adversário por meio do D.O.G.. Caso contrário, além de enviar a jogada, o programa deve notificar o outro usuário acerca do encerramento da partida (e vitória do jogador local), via D.O.G.. A jogada a ser enviada deve consistir em: posição na qual a jogada foi executada e o tipo da jogada que foi efetuada. Caso a jogada efetuada seja do tipo “inserção”, deve ser enviado também um número referente ao valor da peça inserida. Após o envio das informações da jogada efetuada, o programa deve desabilitar o jogador local, aguardando uma sinalização do jogador adversário (conforme descrito no R.F. 7).

- VI - Requisito Funcional 6 (Receber determinação de início):** Após a execução dos procedimentos envolvidos no cumprimento do Requisito Funcional 2 por um dos jogadores, o programa deverá notificar o outro jogador a respeito do início de partida (via D.O.G. Server). Em seguida, o programa deverá atualizar a interface, a fim de representar graficamente o início da partida. Caso o jogador local seja o primeiro a jogar, o programa deverá habilitar a interface para que o mesmo possa realizar a sua jogada.
- VII - Requisito Funcional 7 (Receber Jogada):** Após o fim do turno (por parte de um dos jogadores), o programa deverá receber as informações referentes à jogada anterior (como descrito no R.F. 5) do jogador remoto, o que deve ocorrer através do D.O.G. Server. Em seguida, o programa deverá atualizar a interface gráfica, para que a posição exibida pelo tabuleiro reflita a última jogada realizada pelo jogador remoto. Caso as informações indiquem que o jogo deve ser encerrado, o programa deverá notificar o jogador local sobre desse fato, bem como acerca do nome de quem venceu a partida. Caso contrário, deve ser habilitado o jogador local, para que o mesmo possa realizar o seu próximo lance.
- VIII - Requisito Funcional 8 (Receber notificação de abandono de partida):** Caso seja recebida uma notificação de abandono de partida (via DOG Server) por parte de um dos jogadores, o programa deverá encerrar a partida e a interface gráfica deve informar o jogador remanescente acerca do abandono da partida.
- IX - Requisito Funcional 9 (Alterar tema do jogo):** Antes do início da partida (ou seja, no "Menu Principal"), o programa deverá proporcionar ao jogador local o acesso a uma opção que o possibilite alterar o tema (ver seção 1.2) das peças e do tabuleiro, enriquecendo a gama de opções estéticas oferecidas pelo jogo.

**X - Requisito Funcional 10 (Finalizar Programa):** O programa deverá proporcionar ao usuário uma maneira de finalizar a execução do programa, através do pressionamento de um botão. Essa funcionalidade deverá ser passível de execução apenas quando o usuário estiver no menu principal.

### **3.2 Requisitos Não-funcionais:**

**I - Requisito Não-funcional 1 (Tecnologia de interface gráfica para usuário):**

A interface gráfica deverá ser baseada na biblioteca TKinter.

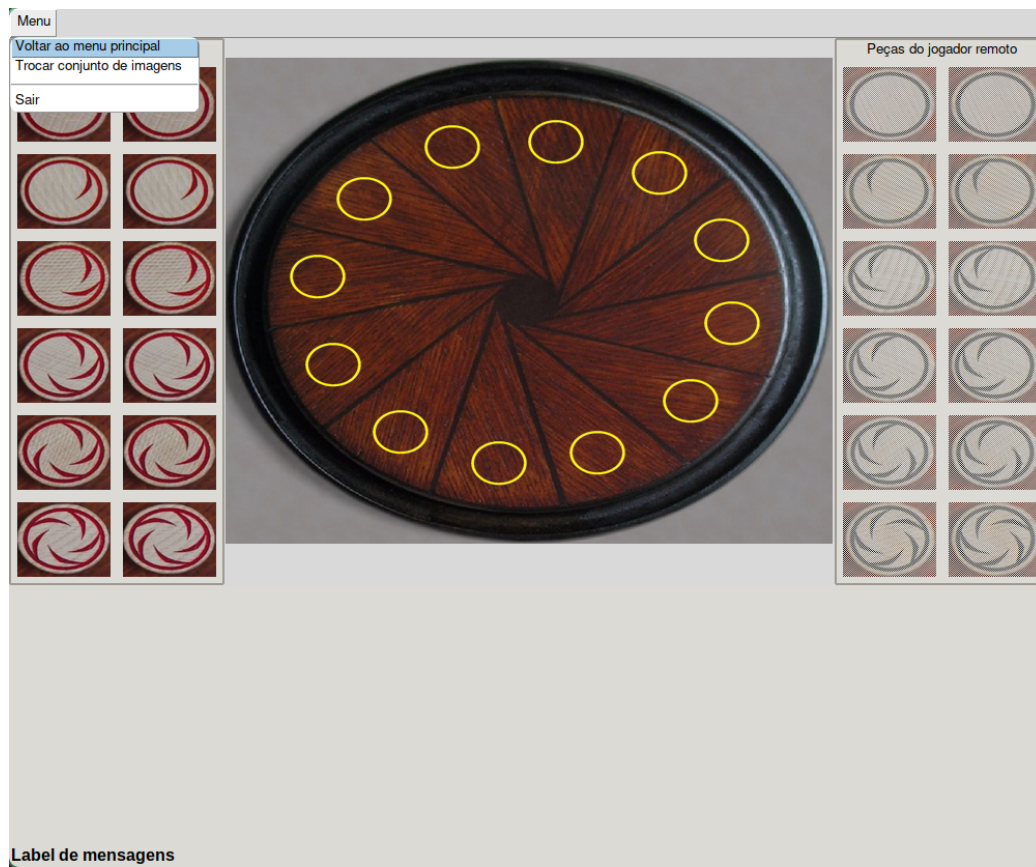
**II - Requisito Não-funcional 2 (Suporte para a especificação de projeto):**

A Especificação de Projeto deve ser produzida com apoio da ferramenta Visual Paradigm.

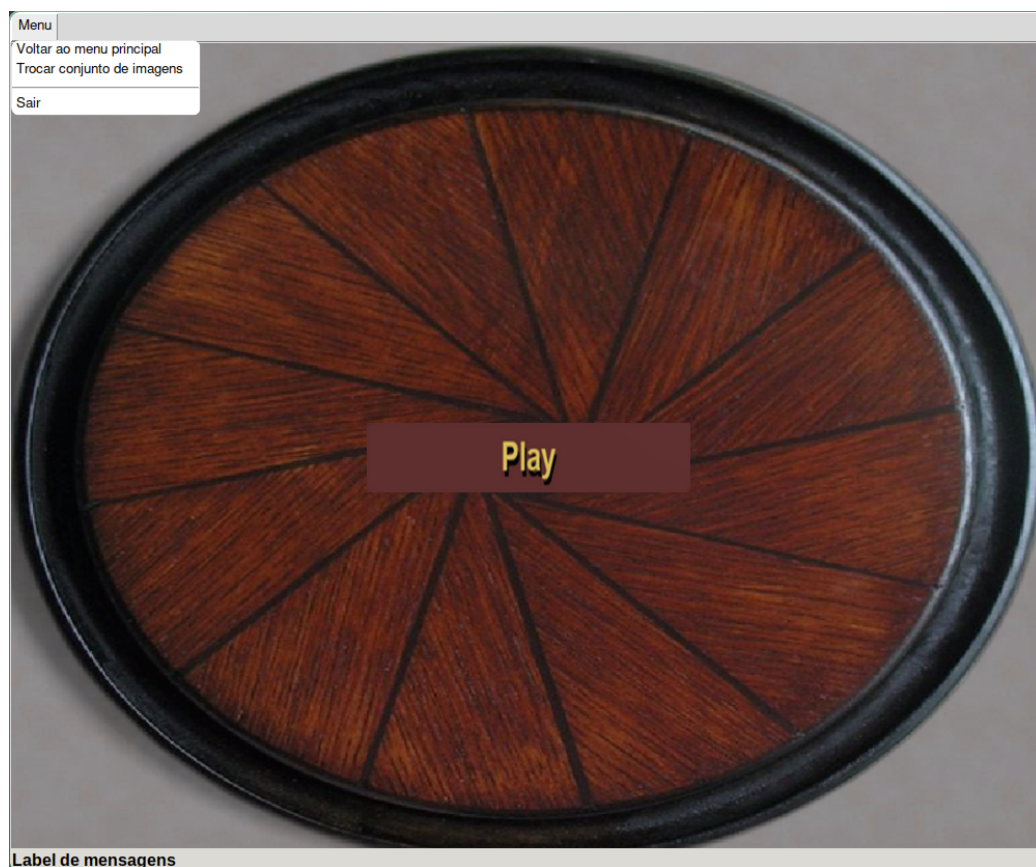
**III - Requisito Não-funcional 3 (Interfaces do Programa):** O programa deverá possuir tanto uma interface de representação visual da partida atual quanto uma interface de Menu Principal.



Essas interfaces deverão ser produzidas conforme os seguintes esboços (Figuras 1 e 2):



**Figura 1:** Interface Gráfica



**Figura 2:** Interface do Menu Principal

## **A - Qyshinsu : Informações sobre o Jogo**

- 1 - O “equipamento” (elementos do jogo):** O jogo é composto pelos seguintes componentes:

a-) **O tabuleiro:** O tabuleiro do jogo consiste em um círculo dividido em 12 casas triangulares, conforme ilustrado pela figura a seguir:



**Figura 3:** O Tabuleiro

- b-) **As pedras:** Cada jogador inicia o jogo possuindo dez pedras comuns e duas pedras especiais (contabilizando um total de 24 pedras, sendo 12 de cada jogador), conforme ilustrado pelas figuras 4 e 5:



**Figura 4:** Pedras Negras



**Figura 5:** Pedras Vermelhas

Cada pedra é nomeada de acordo com o número de marcações presentes em suas faces, exceto pela Pedra Antiga, que é uma pedra especial - cujas faces não apresentam nenhuma marcação. Desta forma, as demais pedras são batizadas como “1-pedra”, “2-pedra”, e assim em diante.

**2 - As Regras:** O jogador com as pedras negras deve iniciar o jogo colocando qualquer uma de suas pedras em uma casa arbitrária. Em seguida, seu turno termina, e o seu adversário deve jogar, de acordo com a estrutura básica definida abaixo:

- O jogador ativo deve realizar um lance, que pode consistir na inserção ou remoção de uma de suas pedras no tabuleiro;
- Se o lance anterior envolveu uma “*n-pedra*” - isto é, não envolveu uma Pedra Antiga, o jogador ativo deve inserir ou remover uma de suas pedras no tabuleiro, em uma casa que diste exatamente “*n*” espaços relativamente à casa envolvida no último lance;

- Caso o lance anterior tenha envolvido a *inserção* de uma Pedra Antiga, o jogador ativo deve *inserir* uma de suas pedras na casa vazia mais próxima da casa envolvida no último lance, podendo escolher entre as duas em situações em que haja equidistância entre elas e a casa envolvida no lance anterior;
- Caso o lance anterior tenha envolvido a *remoção* de uma Pedra Antiga, o jogador ativo deve *remover* uma de suas pedras da casa mais próxima daquela envolvida no último lance, podendo escolher entre as duas em situações em que haja equidistância entre as suas pedras mais próximas e a casa envolvida no lance anterior.

Há três regras adicionais, denominadas de “*preceitos*” pelo autor do jogo original:

**Preceito 1:** Cada jogador só pode mover as suas próprias pedras;

**Preceito 2:** Apenas duas pedras de cada número - não importando a cor das mesmas - podem estar presentes no tabuleiro simultaneamente (isso também vale para as Pedras Antigas);

**Preceito 3:** Se a jogada mais recente *do jogador ativo* envolveu a *remoção* de uma pedra, este mesmo jogador não poderá *inserir* uma pedra com o mesmo número e na mesma posição no seu turno atual.

Caso o jogador ativo não possua nenhum lance legal disponível, o jogo termina, com o seu adversário sendo declarado o vencedor.