UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Aline Cristina Meyer
Camila Prim
Fabio Fernandes da Silva Junior

Especificação de Requisitos do jogo Kalah

Florianópolis, 2025.

SUMÁRIO

Versão	Autor(es)	Data	Ação
1.0	Identificados na Capa	16/04/2025	Especificação dos requisitos

Versão	Autor(es)	Data	Ação
2.0	Identificados na Capa	12/05/2025	Especificação arquitetural

1. Objetivo:

Desenvolver um sistema digital do jogo Kalah, pertencente à família Mancala, que permita a dois jogadores competirem entre si através de uma interface gráfica intuitiva. O software implementará todas as regras do jogo, incluindo distribuição de sementes, capturas e turnos extras, garantindo uma experiência fiel à versão tradicional do jogo e promovendo o desenvolvimento de habilidades estratégicas.

1.2. Definições e abreviaturas:

- **Kalah:** Variante ocidental do jogo Mancala, jogado em um tabuleiro com 12 casas e 2 armazéns.
- **Mancala:** Família de jogos de tabuleiro de origem africana baseados na semeadura e captura.
- **Casa:** Receptáculo pequeno no tabuleiro onde ficam as sementes.
- **Armazém/Kalah:** Receptáculo maior nas extremidades do tabuleiro onde os jogadores acumulam pontos.
- **Semente/Esfera:** Peças do jogo distribuídas durante as jogadas.
- **RF:** Requisito Funcional.
- **RNF:** Requisito Não-Funcional.
- **Captura:** Ação de obter sementes do lado adversário.
- **Turno extra:** Jogada adicional concedida ao jogador.

1.3. Referências:

1. Russ, L. (2000). The Complete Mancala Games Book: How to Play the World's Oldest Board Games. Marlowe & Company.

2. Visão Geral

2.1. Arquitetura do programa:

Programa orientado a objetos, cliente-servidor distribuído com DOG e o cliente será baseado em MVC.

2.2. Premissas de desenvolvimento:

1. **Tecnologia específica:** Desenvolvimento em Python 3.10+ e framework Tkinter para a interface gráfica.
2. **Modelagem UML:** Todo o sistema deve ser modelado previamente utilizando diagramas UML (classes, casos de uso, sequência e atividades) com o software Visual Paradigm.
3. **DOG:** Utilização do DOG como suporte para execução distribuída, permitindo comunicação e sincronização entre os jogadores.
4. **MVC**: Adoção do padrão arquitetural MVC na camada cliente.

3. Requisitos de Software

3.1. Requisitos Funcionais:

RF01 – Iniciar Partida: O cliente deve conectar-se ao servidor DOG e, após dois jogadores estarem conectados, um deles aciona “Iniciar Partida”.Deve ser criada

a instância de jogo no servidor DOG, distribuir automaticamente 48 (4 para casa casa) sementes em 12 casas (6 por jogador), zerar ambos os armazéns e sortear aleatoriamente quem começa, exibindo na interface.

RF02 – Sincronização de Estado Inicial: A cada cliente deve ser enviado, pelo DOG, o estado inicial completo do tabuleiro (posição das sementes e vez do jogador), garantindo consistência entre todas as instâncias.

RF03 – Controle de Acesso por Turno: Apenas o jogador com vez ativa pode interagir com o tabuleiro. Clientes sem vez aguardam notificação do DOG antes de habilitar ações.

RF04 – Validar Jogadas Ao selecionar uma casa, o cliente solicita ao controlador a validade da jogada (casa não vazia, pertencente ao jogador). Se inválida, exibe mensagem clara explicando o motivo.

RF05 – Realizar Jogada (Distribuição de Sementes): Validada a jogada, o cliente envia a ação ao DOG, que devolve o novo estado do tabuleiro.A distribuição de sementes ocorre no sentido anti-horário, percorrendo casas de ambos os jogadores e o armazém do atual, pulando o armazém adversário.

RF06 – Conceder Turno Extra: Se a última semente cair no armazém próprio, o DOG mantém a vez do mesmo jogador e notifica o cliente para indicar o turno extra.

RF07 – Captura de Sementes: Caso a última semente caia em casa vazia do jogador atual e a casa oposta possua sementes, o jogador captura todas elas para o armazém do atual e transmite o estado atualizado.

RF08 – Verificar Fim de Partida: Após cada jogada, o DOG verifica se todas as casas de um lado estão vazias. Se sim, move as sementes restantes do adversário para o armazém correspondente, encerra a partida e envia o estado final.

RF09 – Exibir Vencedor: Ao receber a notificação de fim de jogo, o cliente compara as sementes em cada armazém e exibe quem venceu ou se houve empate, com mensagem clara.

RF10 – Reiniciar Partida: Disponibilizar opção “Reiniciar” que solicite ao DOG nova instância de jogo, restaurando o estado inicial (4 sementes por casa, armazéns vazios) sem fechar a aplicação.

RF11 – Encerrar Aplicação: Permitir que o jogador feche o cliente a qualquer momento via botão/menu “Desistir”, encerrando graciosamente a conexão com o servidor DOG.

3.2. Requisitos Não Funcionais:

RNF01 – Interface Gráfica: A interface gráfica do jogo deve ser desenvolvida utilizando o framework Tkinter, nativo da linguagem Python.

RNF02 – Responsividade Visual: O sistema deve atualizar a interface gráfica em tempo real sempre que houver movimentação de sementes, captura ou troca de turno, proporcionando feedback visual claro ao usuário.

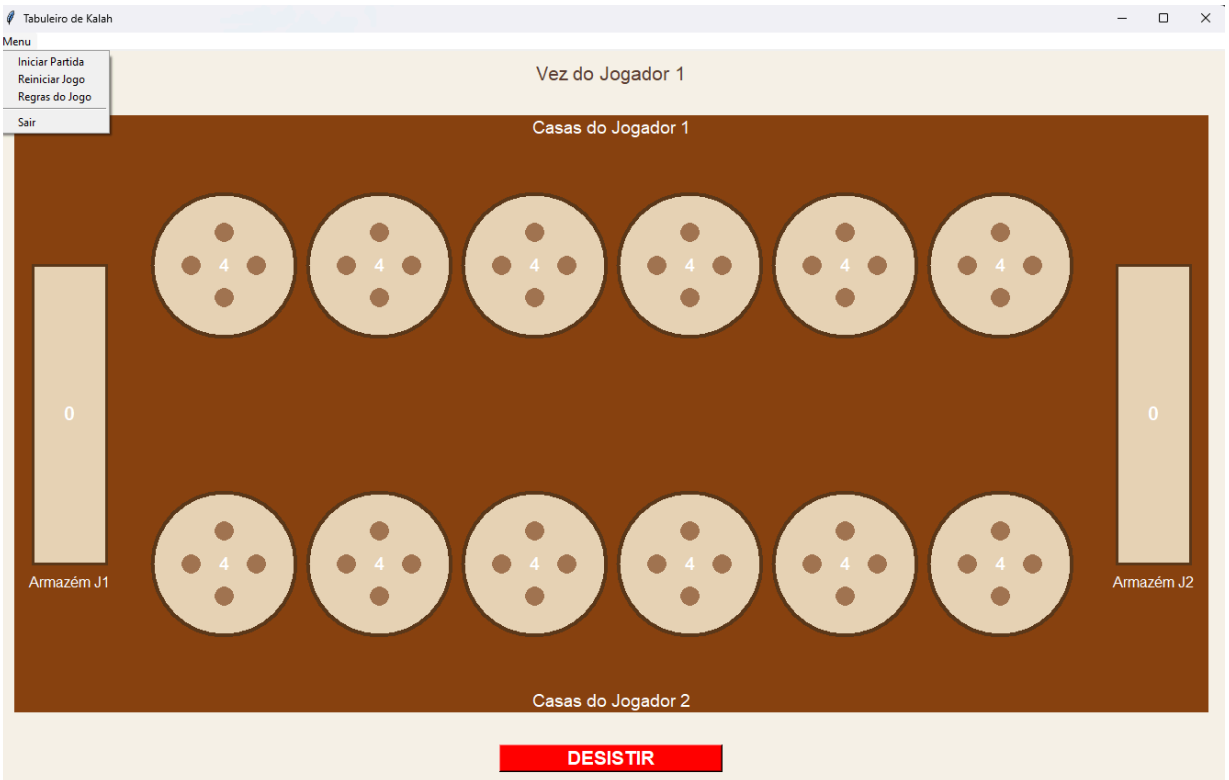
RNF03 – Indicação Visual do Jogador Ativo: A interface do jogo deve destacar visualmente qual jogador possui o turno ativo através de elementos gráficos como mudança de cor, destaque do nome do jogador ou indicadores direcionais, garantindo que ambos os jogadores possam identificar facilmente de quem é a vez de jogar.

RNF04 – Linguagem de Programação: A aplicação deve ser desenvolvida utilizando a linguagem Python 3.10 ou superior.

RNF05 – Persistência Temporária: Não é necessário persistência em banco de dados. Os dados da partida serão mantidos apenas em tempo de execução.

RNF06 – Ferramenta de Modelagem: As modelagens UML (diagrama de classes, casos de uso, sequência e atividades) devem ser realizadas utilizando o software Visual Paradigm.

RNF07 – Inspiração Visual: A imagem [1] apresentada a seguir, serve como referência para a criação da interface gráfica.



[1]: Interface Gráfica

5. Apêndice

4.1. Objetivo do jogo:

O objetivo do jogo Kalah é capturar o maior número possível de sementes em seu próprio armazém até o final da partida. Os jogadores se revezam realizando

jogadas que envolvem a coleta e distribuição de sementes a partir das casas do seu lado do tabuleiro. Ao final do jogo, vence o jogador que tiver acumulado mais sementes em seu armazém.

4.2. Regras:

Configuração Inicial: O tabuleiro contém 12 casas (6 para cada jogador) e 2 armazéns (um de cada lado). Cada casa começa com 4 sementes. Os armazéns começam vazios.

Distribuição: O jogador da vez escolhe uma de suas casas, recolhe todas as sementes dela e distribui, uma a uma, no sentido anti-horário, pulando o armazém do adversário.

Turno Extra: Se a última semente cair no próprio armazém, o jogador ganha um turno extra.

Captura: Se a última semente cair em uma casa vazia do lado do jogador e a casa oposta (do adversário) estiver com sementes, todas as sementes da casa oposta (incluindo a última jogada) são capturadas e movidas para o armazém do jogador.

Fim de Jogo: A partida termina quando todas as casas de um dos lados do tabuleiro estiverem vazias. As sementes restantes do lado do adversário são movidas para o respectivo armazém.

Vencedor: Vence o jogador com mais sementes em seu armazém. Em caso de empate, declara-se empate.