

---

# **ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE**

**para**

**JOGO DE TABULEIRO “ALAPO”**

**Versão 1.0**

**Preparado por:**

- 1. Pedro Santi Binotto (20200634)**
- 2. Gabriel Lemos da Silva (18200628)**
- 3. Petterson Pereira da Rosa (20105097)**

**Professor: Ricardo Pereira e Silva**

**16 de setembro de 2024**

# Sumário

<b>Lista de Definições</b>	<b>3</b>
<b>1 Introdução</b>	<b>4</b>
Propósito . . . . .	4
Regras do jogo . . . . .	4
Sobre a movimentação . . . . .	5
Configuração inicial do tabuleiro . . . . .	5
Progressão da partida . . . . .	5
<b>2 Visão Geral</b>	<b>7</b>
Arquitetura do programa . . . . .	7
Premissas de desenvolvimento . . . . .	7
<b>3 Requisitos de Software</b>	<b>8</b>
Requisitos Funcionais . . . . .	8
Requisitos Não-Funcionais . . . . .	8
<b>Bibliografia</b>	<b>9</b>

# Lista de Definições

- **Peça Circular Grande**; pode ser referenciada de forma geral como “●” ou “**Peça Circular Grande**”; ou especificamente por “●” ou “○” quando contexto de jogador é especificado. . 4, 5
- **Peça Quadrada Grande**; pode ser referenciada de forma geral como “■” ou “**Peça Quadrada Grande**”; ou especificamente por “■” ou “□” quando contexto de jogador é especificado. . 4, 5
- ▲ **Peça Triangular Grande**; pode ser referenciada de forma geral como “▲” ou “**Peça Triangular Grande**”; ou especificamente por “▲” ou “△” quando contexto de jogador é especificado. . 4, 5
- **Peça Circular Pequena**; pode ser referenciada de forma geral como “●” ou “**Peça Circular Pequena**”; ou especificamente por “●” ou “○” quando contexto de jogador é especificado. . 5
- **Peça Quadrada Pequena**; pode ser referenciada de forma geral como “■” ou “**Peça Quadrada Pequena**”; ou especificamente por “■” ou “□” quando contexto de jogador é especificado. . 4, 5
- ▲ **Peça Triangular Pequena**; pode ser referenciada de forma geral como “▲” ou “**Peça Triangular Pequena**”; ou especificamente por “▲” ou “△” quando contexto de jogador é especificado. . 4, 5
  
- Alapo** Um jogo de estratégia semelhante ao *xadrez*, projetado para dois jogadores. Desenvolvido por Johannes Tranelis em 1982, na Alemanha, e publicado pela *Edition Perlhuhn*, que oferece o produto até o dia de hoje[3]. . 4
  
- DOG** DOG é uma solução que possibilita o desenvolvimento de jogos como programas distribuídos a desenvolvedores de software capacitados apenas para desenvolvimento centralizado[2]. . 7
  
- UML** UML, abreviação de *Unified Modeling Language*, é uma linguagem de modelagem padronizada que consiste em um conjunto integrado de diagramas, desenvolvida para ajudar desenvolvedores de sistemas e software a especificar, visualizar, construir e documentar os artefatos de sistemas de software, bem como para modelagem de negócios e outros sistemas não-software[1]. . 7

# 1 Introdução

## Propósito

Desenvolvimento de um programa distribuído que possibilite a realização de disputas do jogo “Alapo”[3] na modalidade *usuário vs. usuário*.

## Regras do jogo

O jogo deve ser jogado em um tabuleiro quadrado de 36 (6 × 6) posições; cada jogador deve iniciar uma partida com 12 peças para si, para um total de 24 peças no tabuleiro no momento que a partida é iniciada.

As peças do jogo são de seis (6) tipos diferentes, e cada jogador deve receber duas peças de cada tipo, sendo que as peças devem ser identificadas por cor, para que se possa fazer a distinção entre quais peças no tabuleiro pertencem a cada jogador:

### Peça Triangular Grande:

Sinalizada neste documento pelo símbolo “▲”, o movimento dessa peça é oblíquo, ou seja, pode movimentar-se diagonalmente no tabuleiro, e pode fazê-lo por quantas posições o jogador desejar; ***duas peças para cada jogador***;

### Peça Quadrada Grande:

Sinalizada neste documento pelo símbolo “■”, o movimento dessa peça é ortogonal, ou seja, pode movimentar-se horizontalmente como verticalmente no tabuleiro, e pode fazê-lo por quantas posições o jogador desejar; ***duas peças para cada jogador***;

### Peça Circular Grande:

Sinalizada neste documento pelo símbolo “●”; o movimento dessa peça abrange manobras diagonais assim como ortogonais, podendo ser movida em todas as direções, por quantas posições o jogador desejar; ***duas peças para cada jogador***;

### Peça Triangular Pequena:

Sinalizada neste documento pelo símbolo “▲”; o movimento dessa peça é oblíquo, ou seja, pode movimentar-se diagonalmente no tabuleiro, porém apenas uma por posição por vez; ***duas peças para cada jogador***;

### Peça Quadrada Pequena:

Sinalizada neste documento pelo símbolo “■”; o movimento dessa peça é ortogonal, ou seja, pode movimentar-se horizontalmente como verticalmente no tabuleiro, porém apenas uma por posição por vez; ***duas peças para cada jogador***;

### Peça Circular Pequena:

Sinalizada neste documento pelo símbolo “●”; o movimento dessa peça abrange manobras diagonais assim como ortogonais, podendo ser movida em todas as direções, porém apenas uma por posição por vez; **duas peças para cada jogador**;

### Sobre a movimentação

- A movimentação do jogo **não permite** que peças “pulem” por cima umas das outras, ou seja, as peças só podem transitar por um trecho que está inobstruído do início ao fim;
- O jogo é jogado em turnos; cada jogador deve realizar **apenas um movimento por turno**;
- A *captura* é realizada ao passar uma peça para a posição onde uma peça do oponente está ocupando (análogo à forma que essa mecânica se apresenta no *xadrez* ou na *dama*);

### Configuração inicial do tabuleiro

Para começar uma partida as peças devem ser posicionadas, em extremidades do tabuleiro, por cada jogador, na seguinte disposição:

- As peças **maiores** (▲, ■, ●) devem ser posicionadas ao longo da **primeira linha** relativa ao jogador (linha “1” ou linha “6”, dependendo da perspectiva);
- As peças **menores** (▲, ■, ●) devem ser posicionadas ao longo da **segunda linha** relativa ao jogador (linha “2” ou linha “5”, dependendo da perspectiva);
- As peças **quadradas** (■, ■) devem ser posicionadas nas posições pertencentes às colunas mais externas do tabuleiro (colunas “A” e “F”);
- As peças **triangulares** (▲, ▲) devem ser posicionadas nas posições imediatamente internas às peças **quadradas** (colunas “B” e “E”);
- As peças **circulares** (●, ●) devem ser posicionadas nas posições pertencentes às colunas mais internas do tabuleiro (colunas “C” e “D”);

De maneira que o tabuleiro apresente a seguinte configuração, assim que todas as peças estejam devidamente posicionadas:

### Progressão da partida

Assim que as peças estiverem posicionadas, um dos jogadores (qual jogador iniciará a primeira jogada pode ser escolhido aleatoriamente) escolhe uma das suas peças e faz o primeiro movimento. Os jogadores tomam turnos realizando lances (movimentos) e capturando as peças um do outro, com o objetivo alcançar a extremidade oposta do tabuleiro com uma de suas peças, sem ser capturado.

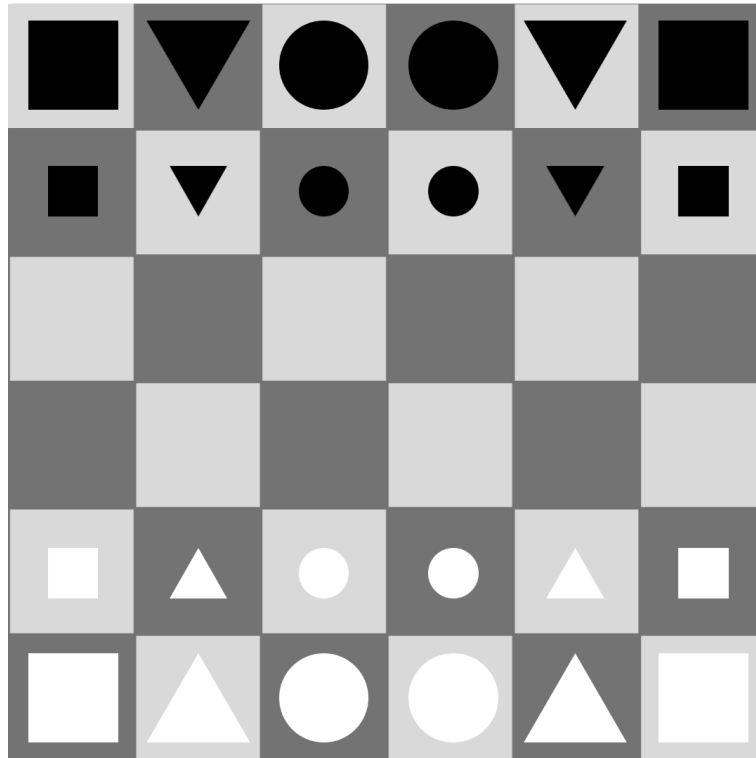


Figura 1.1: Configuração inicial do tabuleiro

- Uma **vitória** é configurada quando um jogador alcança o extremo oposto do tabuleiro *sem ter sua posição imediatamente contestada*, isto é, quando a posição ocupada pela peça não está a alcance de captura de nenhuma peça do oponente;
- Um **empate** é configurado quando a condição de vitória é alcançada por ambos os jogadores *no intervalo de um turno*;
- Caso um jogador alcance o extremo oposto do tabuleiro e a posição ocupada pela peça seja contestável pelo oponente, **a contestação por parte do oponente é obrigatória** e deve ser realizada no turno imediatamente após a ocupação da posição;

## 2 Visão Geral

### Arquitetura do programa

A arquitetura do programa é baseada em *cliente-servidor distribuído*.

### Premissas de desenvolvimento

- O programa deve ser implementado em Python;
- O programa deve usar **DOG** como suporte para execução distribuída;
- Além do código, deve ser produzida especificação de projeto baseada em **UML**, segunda versão.

## **3 Requisitos de Software**

**Requisitos Funcionais**

**Requisitos Não-Funcionais**



# Bibliografia

- [1] What is unified modeling language (uml)? <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>. Accessed: 2024-09-15.
- [2] Ricardo Pereira e Silva. Dog. <https://www.inf.ufsc.br/~ricardo.silva/dog/>. Accessed: 2024-09-15.
- [3] Johannes Tranelis. Alapo - edition perlhuhn. <https://editionperlhuhn.de/spiele/alapo>. Accessed: 2024-09-15.