

# Projeto Pyon

## Especificação de Requisitos de Software

Versão 1.0

23/03/2023

Versão	Autores	Data	Ação
1.0	Gabriel Fornaza da Silveira  Arthur Alexandre Nascimento  Luiz Gabriel Slongo	03/04/2024	Estabelecimento dos requisitos

# **Tabela de Conteúdos**

## **1. Introdução**

- 1.1. Objetivo
- 1.2. Definições, abreviaturas
- 1.3. Regras do Jogo
- 1.4. Referências

## **2. Visão Geral**

- 2.1. Arquitetura do programa
- 2.2. Premissas de desenvolvimento

## **3. Requisitos de Software**

- 3.1. Requisitos Funcionais
- 3.2. Requisitos Não Funcionais

# 1. Introdução

## 1.1. Objetivo

Desenvolvimento de um programa distribuído para efetuar partidas do jogo Pyon em formato usuário contra usuário.

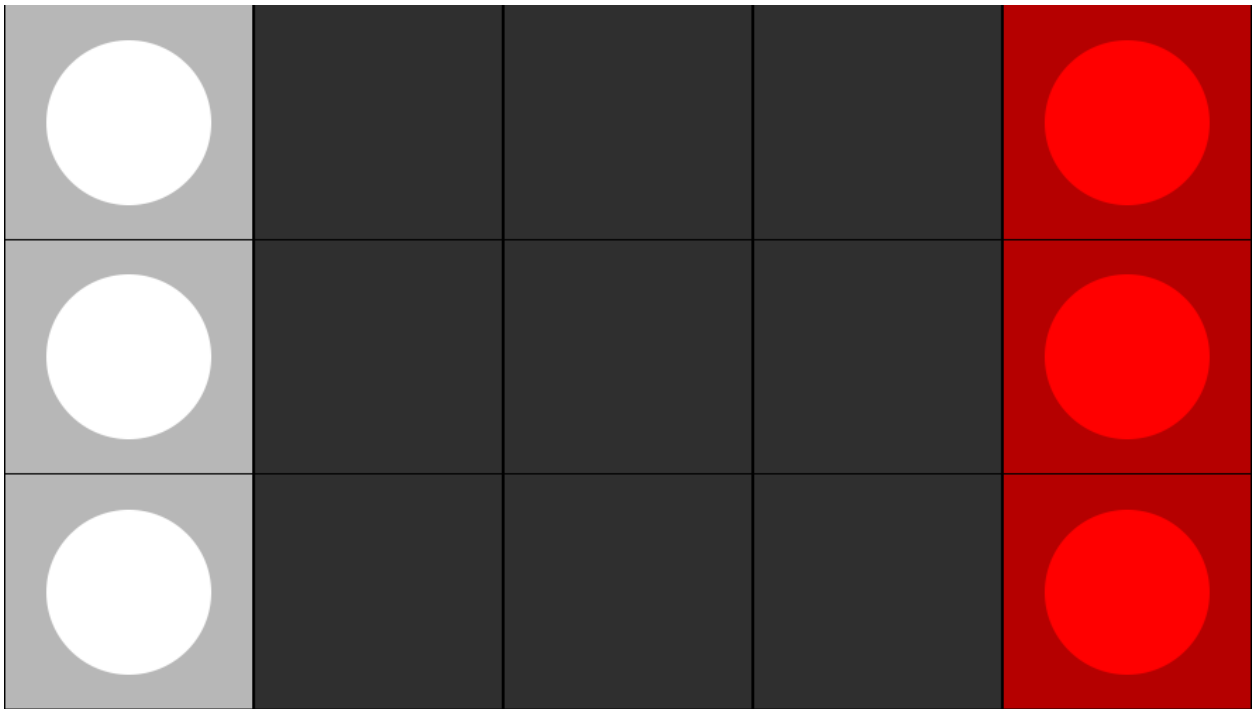
## 1.2. Definições, abreviaturas

**RF:** Requisito Funcional.

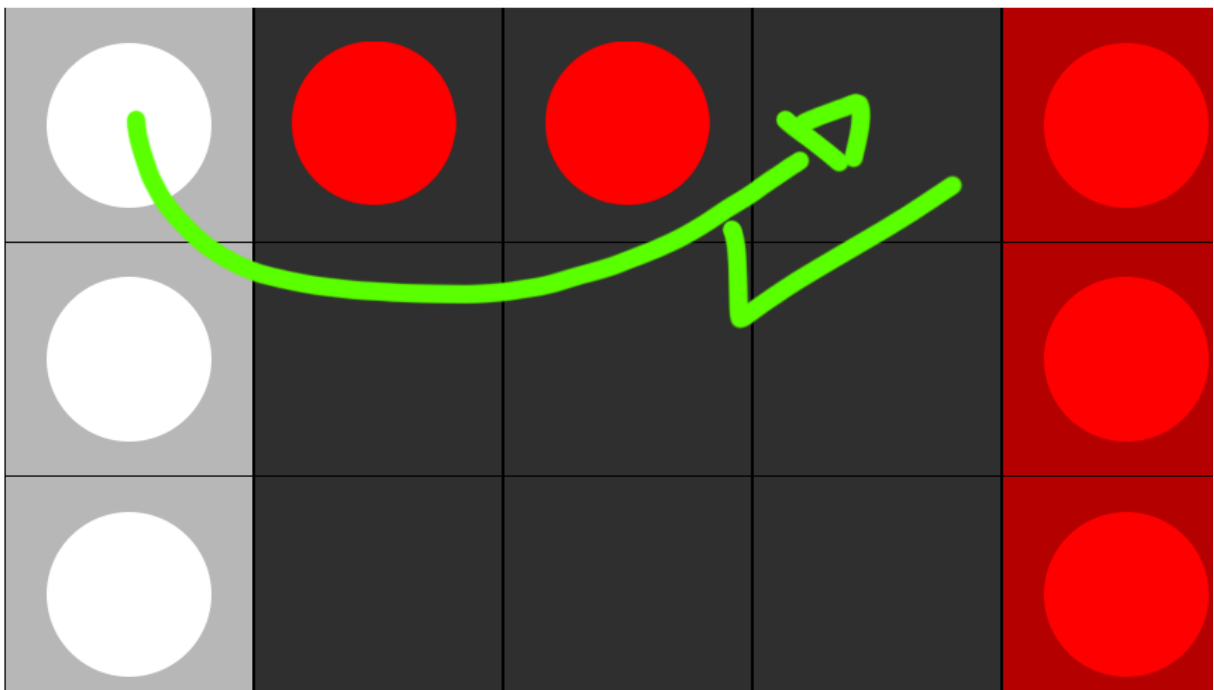
**RNF:** Requisito Não Funcional.

## 1.3. Regras do Jogo

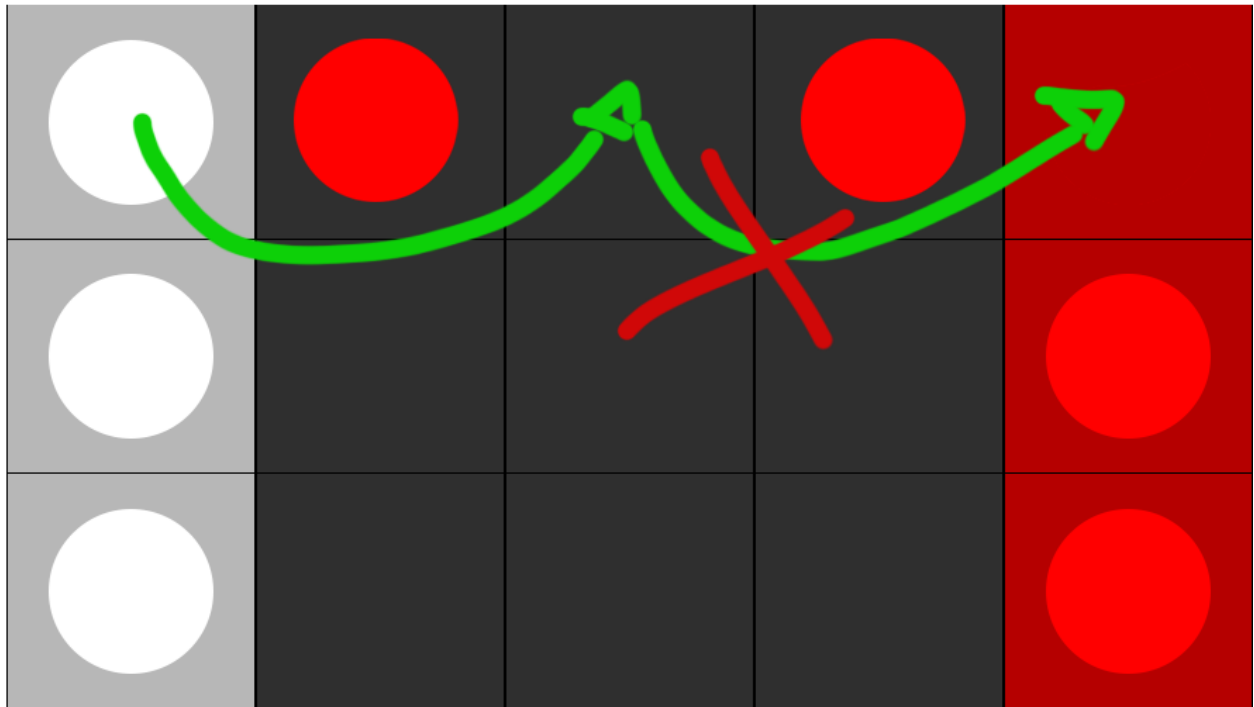
Os dois jogadores começam com 3 peças (os círculos na imagem), posicionadas em suas respectivas cores, brancas começam nos quadrados brancos e vermelhas nos quadrados vermelhos. O objetivo do jogo é colocar as 3 peças nos 3 quadrados adversários, o primeiro jogador que fizer isso é o ganhador. As peças podem se movimentar em todas as direções, vertical, horizontal e diagonal, as peças também podem saltar sobre outras, não havendo limite do tamanho do salto, podendo saltar sobre suas próprias peças, desde que as peças estejam adjacentes e na diagonal. Não é possível fazer pulos em sequência como em damas. Se o jogador saltar sobre uma peça adversária ele pode colocar essa peça adversária em qualquer posição. Abaixo é uma imagem do tabuleiro:



Exemplos de captura de peça: O exemplo abaixo é uma jogada válida



O Exemplo abaixo não é uma jogada válida



## 1.4. Referências

Apresentação do jogo (vídeo do canal Cidade Curiosa):

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

## 2. Visão Geral

### 2.1. Arquitetura do programa

Cliente-servidor distribuído.

### 2.2. Premissas de desenvolvimento

- O programa deverá ser escrito em Python.
- O programa deve usar a biblioteca Pynetgames como plataforma para execução distribuída.

## 3. Requisitos de Software

### 3.1. Requisitos Funcionais

**RF 01 - Iniciar programa:** Ao iniciar o programa a interface deve ser apresentada com o tabuleiro e peças no estado inicial (**RF 03**). Após isso o programa deve conectar com o servidor Pynetgames, mostrando ao usuário o resultado desta conexão, caso sucedido a opção para Iniciar partida (**RF 02**) estará desbloqueada, caso contrário o programa deve ser reiniciado para tentar uma nova conexão.

**RF 02 - Iniciar partida:** O programa deve possuir uma opção “iniciar partida” (**RNF 03**) que deve mandar uma solicitação de início de partida para o servidor Pynetgames, que irá estabelecer uma conexão entre dois jogadores, caso há sucesso na conexão o jogador selecionado pelo servidor começa a partida, se não houver sucesso o cliente recebe uma notificação do fracasso.

**RF 03 - Restaurar ao estado inicial:** Quando o programa começar (**RF 01**) ou receber o início da partida (**RF 07**) o tabuleiro deve retornar ao estado inicial, com as peças nos lugares descritos nas regras do jogo.

**RF 04 - Selecionar Peça:** O programa deve habilitar o jogador selecionar uma peça e destacar qual peça foi selecionada por uma caixa amarela e os quadrados em que pode se movimentar com círculos verdes, caso o jogador tente selecionar uma peça que não deveria conforme as regras do jogo, o programa deve notificar o jogador que é uma jogada irregular e devendo esperar uma nova seleção do jogador.

**RF 05 - Movimentar Peça:** O programa deve habilitar o jogador a movimentar uma peça que começa ao selecionar ela (**RF 04**) e após deve selecionar uma posição de destino da peça selecionada, as posições válidas estarão marcadas com um círculo verde (**RF 04**). Caso o jogador selecione um destino inválido segundo as regras do jogo ele será notificado e deverá selecionar a peça novamente e selecionar um novo destino. Se for uma posição válida essa jogada será transmitida para o servidor pynetgames e recebido pelo adversário(**RF 06**), após isso deve ocorrer deve ser avaliado o fim da partida caso tenha um vencedor essa jogada.

**RF 06 - Receber jogada:** O programa deve receber uma jogada do adversário enviada pelo servidor Pynetgames quando for a vez do jogador local. A peça deve ser movimentada de acordo com as informações recebidas pelo servidor, após isso deverá ocorrer a avaliação do

estado do jogo, determinando se a partida tem um vencedor com essa jogada, caso sim deve haver uma notificação do ganhador, caso contrário o jogador local faz sua jogada começando com a seleção da peça (**RF 04**).

**RF 07 - Receber início da partida:** O programa deve possibilitar o jogador a receber um início de partida quando estiver conectado ao servidor e receber do servidor um início de partida, ocorrendo isso o tabuleiro será restaurado para seu estado inicial (**RF 03**) e a partida começará.

**RF 08 - Desconectar da partida:** O programa deve ser capaz de enviar uma solicitação de desconexão que será enviada do servidor Pynetgames. A partida deve ser encerrada.

**RF 09 - Recebimento de desconexão:** O programa deve ser capaz de receber uma notificação de desconexão do oponente enviada do servidor Pynetgames. A partida deve ser encerrada.

**RF 10 - Receber Erro:** O programa deve ser capaz de receber erros do servidor, caso for recebido um erro o jogo será colocado no estado inicial (**RF 03**).

## 3.2. Requisitos Não Funcionais

**RNF 01 - Tecnologia de interface gráfica:** A interface deve ser feita com a biblioteca Tkinter.

**RNF 02 - Suporte para a especificação de projeto:** A especificação de projeto deve ser produzida com a ferramenta Visual Paradigm.

**RNF 03 - Interface do programa:** A interface do programa deve usar a imagem abaixo como o modelo:

