



**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**CENTRO TECNOLÓGICO**  
**DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

**PROJETO: MARIO KART**  
**ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE**  
**VERSÃO 1.1**

**Gabriel Saldanha da Silva Athayde**  
**Monique Bertan**

**FLORIANÓPOLIS**

**Julho de 2019**

## HISTÓRICO DE VERSÃO

<b>Versão</b>	<b>Autor(es)</b>	<b>Data</b>	<b>Ação</b>
1.0	Gabriel Saldanha da Silva Athayde e Monique Bertan	15/05/2019	Confecção da primeira versão da especificação de requisitos
1.1	Gabriel Saldanha da Silva Athayde e Monique Bertan	05/07/2019	Refinamento das especificações de requisitos

# Sumário

<b>Introdução</b>	<b>4</b>
Objetivo:	4
Definições:	4
Regras do Jogo	4
Referências	5
<b>Visão Geral</b>	<b>5</b>
<b>Requisitos de Software:</b>	<b>5</b>
Requisitos funcionais:	5
Requisitos não funcionais	6

# 1.Introdução

## 1.1 Objetivo:

Desenvolver um jogo de tabuleiro, que reproduza de maneira simplificada, a interação e diversão proporcionadas pelo clássico jogo da **Nintendo: Super Mario Kart**.

## 1.2 Definições:

**Ataque** - Ação feita pelo jogador com a intenção de atingir o jogador adversário fazendo-o retroceder casas.

**Barra de Energia** - Barra que indica a energia do personagem. Ela será usada para calcular o movimento do personagem no tabuleiro;

**Checkpoint** - Trecho da pista que libera para o jogador a opção de realizar um ataque;

**Força do ataque** - Número sorteado no momento em que for realizado um ataque. Ele determinará se um ataque será bem sucedido ou não.

**Força de defesa** - Número sorteado no momento em que o jogador for atacado. Ele corresponderá a possibilidade do jogador defender-se de um ataque.

**Piloto** - Personagem escolhido pelo jogador para ser usado para os movimentos no tabuleiro;

### 1.2.2 Regras do Jogo

O jogo consiste em uma corrida onde o jogador tem a opção de movimentar-se, atacar o adversário ou defender-se de ataques;

Ao iniciar a partida os jogadores escolhem o piloto que será usado na corrida. Será sorteado quem iniciará o jogo e os personagens são posicionados no tabuleiro.

### **Movimento no Tabuleiro**

Enquanto não cruzar o primeiro checkpoint, os jogadores tem disponível apenas a opção de movimentar o personagem.

O jogador lança o dado. O movimento será determinado pelo número do dado + o número da Barra de energia;

### **Lançar um ataque**

Ao cruzar um checkPoint, o jogador ganha o direito de usar um ataque. A partir da próxima rodada, o jogador escolhe se ataca o adversário ou movimenta o seu personagem.

- Se escolher movimentar seu personagem, o jogador joga o dado e faz o movimento normalmente. O ataque fica armazenado esperando a escolha do jogador. Não é possível acumular mais de um ataque.
- Se o jogador escolher atacar o adversário, ele joga o dado, o número sorteado será a força do ataque. O adversário também lança o dado. O número sorteado por ele será a força de defesa.
  - Se a força de ataque for maior que a força de defesa, o número sorteado reduzirá a barra de energia do adversário. O jogador que realizou o ataque lança o dado e movimenta-se normalmente;
  - Se a força de ataque for menor que a força de defesa, o jogador atacante movimenta-se o número correspondente a sua barra de energia e a sua rodada encerra.

Vence quem completar o circuito primeiro.

## **1.3 Referências**

Pereira e Silva, Ricardo. Apostila Engenharia de Requisitos - Produção de Especificação de Requisitos : [link](#)

Leite, J. C. S. P. Engenharia de Requisitos – notas de aula. Rio de Janeiro, 1994

Benitti, F.B.V. Engenharia de Requisitos – apostila. Florianópolis, 2015

## **2. Visão Geral**

**Arquitetura do Programa:** Programa Orientado a objetos, utilizando arquitetura cliente-servidor distribuído.

#### **Premissas de Desenvolvimento:**

- O programa será desenvolvido utilizando a linguagem Java;
- As especificações do projeto serão feitas utilizando a linguagem UML na versão 2.0;

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

### **3.1 Requisitos funcionais:**

#### **RF.01:**

Início do Jogo: Para o jogo ser iniciado é necessário no mínimo e máximo 2 jogadores, só então a partida é iniciada e a tela do jogo aparece

#### **RF.02:**

Iniciar: O jogo deve permitir que o jogador inicie uma nova partida. Neste momento o jogador entra com o seu nome, escolhe seu avatar entre uma lista de avatares.

#### **RF.03:**

Ações: Os botões de ações ficarão desabilitados, sendo que só serão habilitados caso for a sua vez na rodada ou esteja apto a usá-los.

#### **RF.04:**

Movimentar: O jogador tem o direito de executar a ação de movimentar quando for sua vez, sendo que caso esta ação for escolhida mais nenhuma outra poderá ser selecionada.

#### **RF.04:**

Atacar: O jogador poderá atacar o oponente caso chegue no checkpoint na rodada anterior.

#### **RF.05:**

Pista: A pista é linear onde o jogador se movimenta em direção horizontal até a sua chegada com movimentos.

#### **RF.06:**

Vitória: O jogador será declarado vitorioso somente se chegar até o ponto de chegada da pista.

#### **RF.07:**

Sorte: Para que cada ação seja executada, será lançado na sorte através de um dado de seis faces de valor fracionado dependendo da situação para determinar o peso da ação

## 3.2 Requisitos não funcionais

### Requisito não funcional 1:

A interface gráfica será desenvolvida em Java-Swing, utilizando o software IntelliJ;

### Requisito não funcional 2:

O desenvolvimento do software utilizará o repositório Github para armazenamento do código produzido, com constantes releases para versionamento conforme as entregas.

### Requisito não funcional 3:

Caso algum jogador desconecte o jogo é encerrado no mesmo instante

### Requisito não funcional 4:

As jogadas e seus dados serão passadas completamente para cada jogador que estiver no turno de em formato de programação distribuída