

# Projeto Battle Element

Especificação de Requisitos de Software

Primeira especificação de requisitos

03/04/2023

Edivaldo Ferreira de Souza Júnior

Gabriel Ávila

Guilherme Cavalieri

## **1. Introdução**

### **1.1 Objetivo**

Desenvolvimento de um programa distribuído que suporte a criação de partidas e jogatina entre dois jogadores em computadores distintos.

### **1.2 Explicação do jogo**

Battle Element envolve uma batalha entre dois jogadores utilizando equipes de três personagens com forças de elementos. A luta é baseada em turnos e existem diferentes movimentos para usar na sua vez de utilizar o personagem. O objetivo do jogo é derrotar os três personagens do adversário.

## **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 usar DOG como suporte para execução distribuída
- A interface gráfica deve ser implementada usando a biblioteca Tkinter
- O programa deve ser modelado com base nas especificações da linguagem UML 2
- Para a modelagem, será utilizado o software Visual Paradigm

## **3. Requisitos de Software**

### **3.1 Requisitos Funcionais**

RF01 - O programa deve possibilitar que os jogadores nomeiem-se antes de iniciar uma partida. Após colocar o nome, o programa deve tentar conexão com o DOG Server e mostrar o resultado da conexão. Caso o resultado seja positivo, o programa deve levar o jogador para a fila de jogadores em espera.

RF02 - O programa iniciará uma nova partida quando dois jogadores estiverem disponíveis para jogar e obter resposta positiva do DOG Server

RF03 - O jogador pode montar uma equipe com três personagens escolhendo entre os personagens base e definindo sua classe e elemento que mudarão os atributos dos mesmos.

RF04 - O jogador, quando está no seu turno, deve selecionar uma das possíveis ações para sua jogada. As ações são escolher um alvo, ataque básico, ataque especial e ataque elemental. O programa enviará a ação para o DOG Server, e então para a interface do adversário.

RF05 - O programa deve, no final de cada rodada, checar se há um vencedor. Caso haja, deve enviar essa informação para o DOG Server, e então para a interface do adversário.

Após isso, deve permitir aos jogadores clicarem em um botão para entrar na fila de espera novamente.

RF06 - O programa deve poder receber uma notificação de abandono de partida por parte do adversário remoto, enviada por Dog Server. Neste caso, a partida deve ser considerada encerrada, o abandono notificado na interface e o jogador restante deve ir para a fila de espera novamente.

### **3.2 Requisitos Não-funcionais**

RNF01 - A linguagem de programação utilizada no projeto deve ser Python.

RNF02 - A interface gráfica deve ser implementada usando a biblioteca Tkinter do Python

RNF03 - A tecnologia DOG deve ser usada para permitir o suporte à execução distribuída

RNF04 - A interface gráfica deve ser compartilhada entre os dois jogadores com as mesmas informações durante todo o uso do programa.

### **3.3 Regras de negócio**

RN01 - Existem personagens pré-definidos, porém é possível definir uma classe e elemento, alterando assim seus atributos.

RN02 - Os personagens possuem pontos de vida, mana, ataque, inteligência e agilidade.

RN03 - Caso a vida dos personagens chegue a 0, ele é morto e não pode mais ter turnos até o final da batalha

RN04 - Um personagem não pode usar uma habilidade que custe mais mana do que ele atualmente possui

RN05 - O atributo de ataque define o dano base que os ataques básico, especial e elemental causam nos inimigos.

RN06 - O atributo de agilidade define a prioridade que o personagem possui em receber os turnos.

RN07- O atributo de inteligência define o tempo de recargas das habilidades especiais e elementais

RN08 - As habilidades especiais e elementais começam com números específicos de tempo de recarga.

RN09 - A condição para a vitória é que todos os três personagens do adversário tenham sido derrotados.