

# **Especificação de Requisitos do jogo Marvel vs DC Carding Game**

## **Projeto Marvel vs DC Carding Game**

Especificações de requisito de  
Software: Versão 1.0

Versão	Autores	Data	Ação
2.0	José Victor Feijó, Ion Neto e Matteus Legat	14/11/2015	Estabelecimento de Requisitos

## **Sumário:**

1. Introdução
2. Visão Geral
3. Requisitos de Software

## 1. Introdução

Desenvolver um software que propõe um jogo de cartas estratégico baseado em turnos, na qual cada jogador tem uma quantidade de cartas e são feitos duelos entre elas. O jogo terá regras, descritas abaixo, baseadas em jogos tradicionais de combate de cartas, porém modificadas por razões de originalidade, e a tematica baseada nos heróis dos quadrinhos das produtoras Marvel e DC Comics.

### Rêferencias:

<http://www.dccomics.com/characters> (Página oficial dos personagens da DC Comics)

<http://marvel.com/comics/characters> (Página oficial dos personagens da DC Comics)

## 2. Visão Geral

### Arquitetura do programa:

- ❑ O Programa será desenvolvido na arquitetura orientada-a-objetos, com algumas técnicas de desenho de projeto como MVC (Model-View-Controller) e o uso do padrão Observer.

### Premissas de Desenvolvimento:

- ❑ O programa vai apresentar uma interface gráfica bidimensional, mais conhecido no mundo dos jogos como 2D.
- ❑ O programa deve ser implementado em Java, podendo executar em qualquer máquina que contém uma JVM (Máquina Virtual Java).
- ❑ O programa utilizará a biblioteca para distribuição em rede NetGames.

### Regras de Funcionamento do Jogo:

- **Aspecto Geral:** O jogo é um confronto entre dois jogadores, um com um baralho formado por 20 heróis da Marvel e outro por 20 heróis da DC. Cada jogador tem pontos de vida, e ganha quem zerar os pontos do adversário primeiro. Para causar dano a vida do adversário, é necessário realizar duelos. Ao iniciar o jogo cada jogador pega 5 cartas para sua mão e o campo é inicializado sem nenhuma carta, podendo ter no máximo 5 cartas no campo de cada jogador.

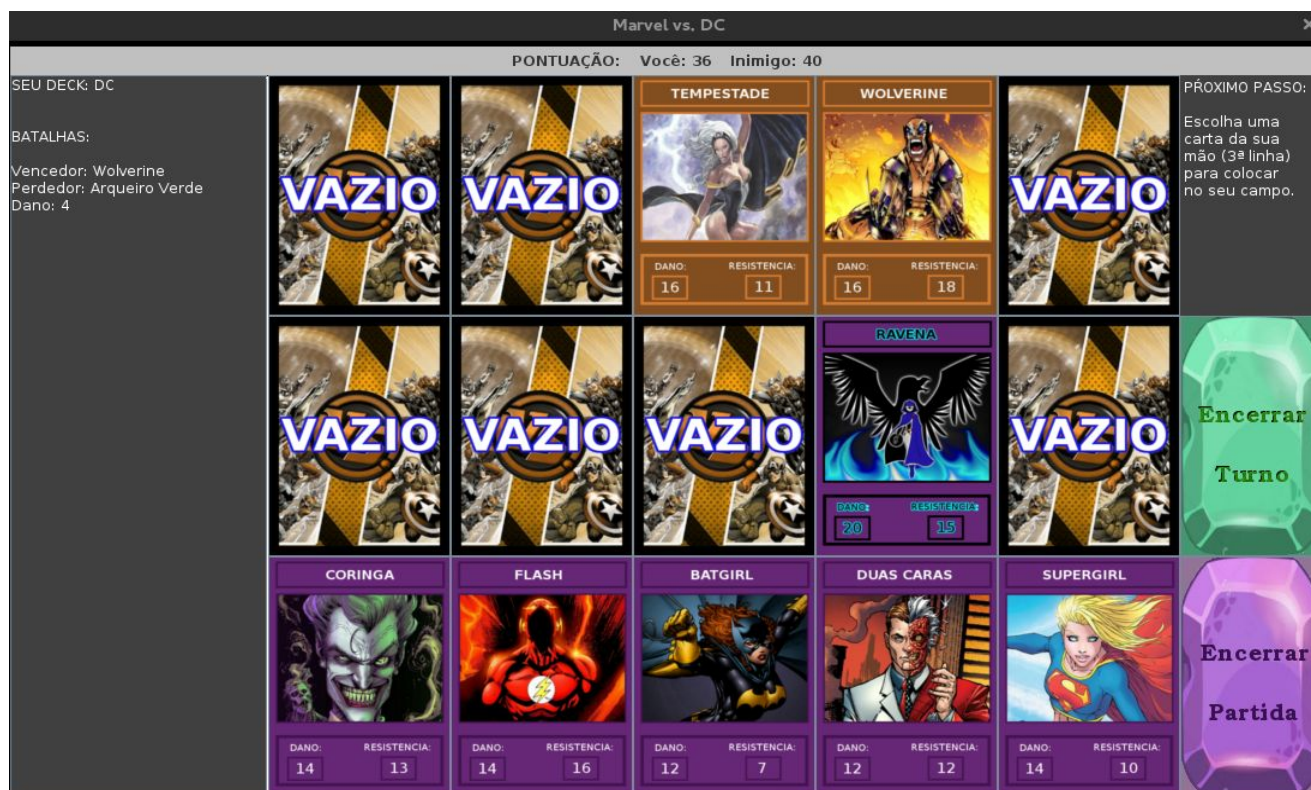
- **Os Turnos:** O jogo é baseado em turnos, onde cada jogador realiza seu turno e aguarda o outro jogador fazer o mesmo.

Cada turno é constituído de três partes:

- Comprar carta: Caso o jogador tenha entre 0 e 4 cartas em sua mão ele deverá obrigatoriamente, no começo de cada turno seu, comprar uma carta de seu baralho.
- Baixar carta: O jogador após comprar uma carta pode posicionar uma carta de sua mão em um dos cinco espaços do campo, desde que ele tenha menos 1 espaço livre em seu campo. O jogador pode abrir mão desta etapa sempre que quiser.
- Atacar: Após “Baixar Carta” o jogador deve escolher se atacar o inimigo ou não, para atacar o jogador deve escolher um de seus heróis e um herói do inimigo para duelar, se o inimigo não tiver nenhum herói o ataque pode ser feito diretamente ao inimigo. O jogador pode atacar com todos os heróis no seu campo em um turno, só não podendo repetir heróis.

- **Atributos das Cartas:** Cada carta de herói tem três atributos: nome, poder e resistencia

- **Duelos:** Os duelos entre heróis são feitos comparando o atributo poder do atacante com o atributo resistencia do defensor, quem tiver o menor número é removido de campo e o dono da carta recebe a diferença deste valor como dano aos seus pontos de vida. Caso ambos tenham o mesmo número, ambas as cartas são removidas de campo. A diferença absoluta do ataque com a defesa é diretamente removido da vida do jogador perdedor.



Montagem da Interface Gráfica do Jogo.



Montagem de uma Carta do Jogo.

## 3. Requisitos de Software

### 3.1 Requisitos Funcionais:

**Requisito Funcional 1 – Iniciar:** O programa deve apresentar na sua interface inicial, um botão iniciar que dara inicio ao jogo. Ao apertar este botão os jogadores serão aleatoriamente pareados com um dos dois baralhos, os baralhos terão sua ordem de cartas embaralhadas e cinco cartas serão levadas para a mão de cada jogador.

**Requisito Funcional 2 - Comprar carta:** o procedimento de comprar carta é feito no início do turno de cada jogador. Cada jogador inicia com 5 cartas em sua mão, mas após jogar uma carta em campo fica apenas com quatro. No início do seu próximo turno, se ele tiver um número de cartas na mão entre 0 e 4 ele recebe uma carta automaticamente no início do seu turno.

**Requisito Funcional 3 – Posicionar carta:** o programa deve permitir que para colocar uma carta em campo, você clique com o *mouse* na carta e depois em qual “slot” do campo deseja coloca-la, podendo colocar uma carta por rodada. Além disso deve haver também um botão de “Encerrar Turno”, permitindo que o jogador não posicione nenhum herói novo nesse turno.

**Requisito Funcional 4 – Procedimento de Duelo:** após passada a etapa de posicionamento de carta, o jogador deve ser capaz de atacar os heróis adversários ou o inimigo diretamente selecionando um herói seu e um alvo. O software deve imprimir um relatório dos danos sofridos após o duelo.

**Requisito Funcional 5 - Procedimento de Encerrar Turno:** após ser feito um duelo o jogador deve clicar no botão encerrar turno para passar para o outro jogador o novo estado do campo e os duelos que foram feitos.

**Requisito Funcional 6 – Fim do jogo:** quando um dos jogadores chegar a zero pontos de vida, o programa deve encerrar a partida e contabilizar o vencedor e mostrar a lista de todas as batalhas feitas e então voltar para o menu inicial, ele deve verificar isso ao fim de cada turno.

### 3.2. Requisitos não funcionais:

**Requisito não funcional 1 – Especificação de projeto:** além do código em java, deve ser produzido diagramas de classe e casos de uso na especificação UML.

**Requisito não funcional 2 – Tecnologia da interface gráfica:** a interface gráfica deve ser projetada utilizando a biblioteca java.swing, ela deve ser implementada de forma mais genérica possível com apenas uma interface para os dois jogadores.

**Requisito não funcional 3 – Banco de dados:** deve existir um banco de dados onde as cartas estão criadas e armazenadas lá, ao início de cada partida uma query é feita para esse banco de dados fazendo um requerimento das 20 cartas da DC para um jogador, e as 20 cartas da Marvel para o outro. Elas deveram vir em ordem aleatória.

**Requisito não funcional 4 – Documentação:** todos os métodos estejam documentados utilizando o javaDOC, para facilitar o entendimento e manutenção do

código.

**Requisito não funcional 5 – NetGames:** o jogo deve utilizar a biblioteca para distribuição em rede NetGames, onde é criada uma instância de jogador para que possa ser jogado em rede.