



DEPARTAMENTO DE INFORMÁTICA ESTATÍSTICA  
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
ENGENHARIA DE SOFTWARE I

ANDRE VAZ DE ASSIS  
DÉCIO MORITZ JUNIOR  
ILÊ CAIAN GUMS  
LUIZ HENRIQUE DE LORENZI CANCELLIER

**REQUISITOS E ANÁLISE DE INTERAÇÃO 1**

Professora: Dra. Patrícia Vilain

Florianópolis

2014

## **1. Visão**

### **1.1. Objetivo**

O objetivo do projeto é desenvolver um jogo de cartas temático. A complexidade deste projeto pode ser facilmente moldada pelas regras do jogo.

### **1.2. Características do Jogo**

É um jogo de cartas, onde existem dois jogadores que se enfrentam e usam seus baralhos para confronto. Um baralho válido deve conter entre 40 e 60 cartas e não pode haver mais que dois exemplares de cada carta nele. Nestes baralhos eles podem escolher entre várias cartas que possuem ações diferentes, e habilidades diferentes.

Quanto aos tipos de cartas, existem:

- Monstros: São criaturas que lutam por seus jogadores atacando e defendendo
- Magias: São poderes que podem ser usados no adversário ou em si mesmo
- Encantamentos: São poderes que podem ser usados em monstros de ambos os lados
- Tributos: São cartas que serão descartadas. No lugar dela é jogado uma carta mais forte.

Ambos os jogadores possuem um marcador de vida iniciado em 30 pontos, quando este marcador chega ao fim ou as cartas do jogador acabam, o jogador perde o jogo.

No começo da partida um jogador é sorteado para iniciar o jogo. Cada jogador retira seis cartas de seu baralho e estas formam sua mão de cartas inicial.

O jogador sorteado deve começar o primeiro turno, sendo que não é permitido atacar neste momento.

Quando um jogador está jogando em seu turno, o outro está esperando e reagindo a suas ações.

Cada turno é dividido em quatro fases onde ocorrem os seguintes eventos:

- O jogador compra uma carta.
- O jogador pode adicionar uma carta a mesa.
- O jogador pode realizar uma ação, mas esta pode ser interrompida pelo seu oponente.
- O jogador pode adicionar mais uma carta a mesa antes de seu turno acabar.

Existe um tabuleiro que permite o uso de um número limitado de cartas em jogo.

As cartas de monstros ou magia são classificadas em elementos, sendo que um elemento possui vantagens e desvantagens contra outros.

### **1.3. Características do Sistema**

O sistema fará a conexão entre os dois usuários, bem como a parte de armazenamento de dados, histórico dos usuários e cadastros dos mesmos.

Também terá uma relação com todas as cartas e baralhos que um usuário pode usar para criar seu baralho antes de entrar na partida.

## **1.4. Usuários do Sistema**

### **1.4.1 Caracterização dos Usuários**

Pessoas que gostam de jogos de cartas.

### **1.4.2 Utilização do Sistema**

O sistema funcionará via conexão entre dois jogadores distintos que serão os usuários. O sistema iniciará a sessão assim que o usuário fizer o login com suas informações (nome de usuário e senha) e para se conectar a uma partida é necessário inserir os dados de IP e porta.

O sistema aguardará para que ambos os jogadores estejam conectados e com seus baralhos. Após a conexão, o jogo se inicia e ambos os usuários se confrontam em uma partida.

Ao terminar a partida os dados são salvos, e ambos os dois são desconectados mostrando a cada um sua pontuação e sua situação: Vitória ou Derrota.

Caso o usuário queira jogar novamente, ele pode informar ao sistema que deseja encontrar outra partida, e assim aguardar para que o processo se repita, ou pode decidir sair do jogo.

## **2. Casos de uso**

1. Conectar a um oponente: Sendo que uma partida foi criada, um oponente que possua o IP e a porta da partida pode se conectar a ela para jogar.

2. Visualizar acervo de cartas do jogo: Jogador pode, sem ter criado conta alguma, visualizar as cartas existentes no jogo, bem como tomar conhecimento de todas as suas características.

3. Filtrar acervo de cartas: Pesquisar as cartas utilizando algum filtro de pesquisa, como por exemplo pesquisar todas as cartas de um determinado tipo.

4. Montar Baralho: O jogador pode montar seu próprio baralho utilizando as cartas disponíveis no acervo para jogar com outros jogadores.

5. Validar baralho: O sistema deve, sempre que um baralho for criado ou editado, validá-lo para assegurar-se que este respeita as regras do jogo.

6. Jogar: Um jogador pode jogar com outros jogadores.

7. Criar Partida: O Usuário que deseja criar uma partida entra com os dados necessários(IP:Porta) para conexão e a partida é criada.
8. Logar no sistema: O usuário deve possuir uma conta válida, deve inserir os dados, e irá ser logado no sistema.
9. Criar conta: O usuário que deseja abrir uma conta no jogo escolhe um login e senha para que seja associado a sua conta. O login deve ser único, e a conta é criada com a confirmação do usuário.
10. Editar baralho: Tendo um baralho criado, é possível adicionar ou remover cartas deste.
11. Remover carta do baralho: Tendo um baralho criado, é possível remover cartas, desde que este não tenha um número de cartas menor que o permitido pelas regras do jogo.
12. Adicionar carta ao baralho: Tendo um baralho criado, é possível adicionar novas cartas, desde que este não tenha um número de cartas maior que o permitido pelas regras do jogo.

## **2.1 Casos de uso completo**

### **2.1.1. Editar Baralho**

**Escopo:** Sistema

**Nível:** Meta de usuário

**Ator primário:** Usuário

**Stakeholders:** Usuário: deseja editar seu baralho.

**Pré-condições:** É necessário que o usuário esteja logado(Logar no Sistema) e que tenha criado um baralho(Montar Baralho).

**Fluxo básico:**

1. Sistema mostra todas as cartas do baralho.
2. Usuário efetua uma alteração.  
*Repete 1-2 até usuário finalizar o baralho.*
3. Usuário finaliza baralho.
4. Sistema tenta validar o baralho. Inclui Validar Baralho.
5. Baralho é salvo.

**Extensões:**

- 2.a Usuário remove uma carta.
  1. Carta é removida. Inclui Remover Carta do Baralho.
- 2.b Usuário adiciona uma carta.
  1. Carta é adicionada. Inclui Adicionar Carta ao Baralho.

**Frequência de ocorrência:** Pouco frequente.

### 2.1.2. Adicionar Carta ao Baralho

**Escopo:** Sistema

**Nível:** Sub-função: parte do caso de uso Editar Baralho.

**Ator primário:** Usuário

**Stakeholders:** Usuário: deseja adicionar carta ao Baralho.

**Fluxo básico:**

1. Sistema exibe o acervo de cartas Inclui Visualizar acervo de cartas.
2. Usuário escolhe uma carta do acervo
3. Sistema adiciona carta escolhida ao baralho.

**Extensões:**

- 2.a Usuário escolhe carta inválida.
  1. Carta não é adicionada ao baralho.

**Frequência de ocorrência:** Sempre que usuário fizer alterações em seu baralho.

### 2.1.3. Remover carta do Baralho

**Escopo:** Sistema

**Nível:** Sub-função: parte do caso de uso Editar Baralho.

**Ator primário:** Usuário

**Stakeholders:** Usuário deseja remover carta do Baralho.

**Fluxo básico:**

1. Usuário escolhe uma carta do baralho.
2. Sistema remove carta escolhida do baralho.

**Frequência de ocorrência:** Sempre que usuário fizer alterações em seu baralho.

### 2.1.4. Logar no Sistema

**Escopo:** Sistema

**Nível:** Meta de usuário

**Ator primário:** Usuário

**Stakeholders:** Usuário: Deseja entrar no jogo.

**Pré-condições:** Usuário deverá possuir login e senha válidos (Criar Conta).

**Fluxo básico:**

1. Usuário acessa portal do jogo.
2. Usuário entra com login e senha.
3. Usuário é logado no sistema.

**Extensões:**

- 2.a. Login e senha não são válidos.
  1. Sistema pede que usuário digite novamente.

**Tecnologia:** É necessário que o usuário tenha uma conexão com a internet.

**Frequência de ocorrência:** Sempre que usuário utilizar o sistema.

### 2.1.5. Criar Conta

**Escopo:** Sistema

**Nível:** Meta de usuário

**Ator primário:** Usuário

**Stakeholders:** Usuário: Deseja abrir uma conta no jogo.

**Pós-condições:** Conta válida é criada.

**Fluxo básico:**

1. Usuário acessa portal do jogo.
2. Entra com login e senha desejados.
3. Conta é criada.
4. Login e senha são mostrados ao usuário novamente para confirmação.

**Extensões:**

\*2. Login já existe.

1. Sistema pede que usuário que digite um login diferente.

**Tecnologia:** É necessário que o usuário esteja conectado à internet.

**Frequência de ocorrência:** Sempre que um novo usuário utilizar o sistema.

### **2.1.6. Criar partida.**

**Escopo:** Jogo de cartas.

**Nível:** Meta de usuário.

**Ator primário:** Usuário.

**Stakeholders:** Usuário: Deseja criar uma partida.

**Pré-condições:** Usuário deverá estar logado. (Logar no Sistema).

**Fluxo básico:**

1. Usuário acessa portal do jogo.
2. Usuário entra com número de IP e porta.
3. Partida é criada.

**Extensões:**

\*a. Usuário possui problema com a conexão.

1. A partida não é criada.
2. Sistema pede que usuário tente novamente.

**Tecnologia:** É necessário que o usuário esteja conectado à internet.

**Frequência de ocorrência:** Frequente.

### **2.1.7. Conectar a um oponente.**

**Escopo:** Jogo de cartas.

**Nível:** Meta de usuário.

**Ator primário:** Usuário.

**Stakeholders e interesses:**

Jogador: Deseja se conectar a uma partida com os dados fornecidos pelo oponente.

Oponente: Deseja que um jogador se conecte a sua partida.

**Pré-condições:** É necessário que jogador e oponente estejam logados (Logar no Sistema), e que o oponente tenha criado uma partida: (Criar Partida).

**Pós-condições:** Jogador e oponente estão aptos a iniciar a partida.

**Fluxo básico:**

1. O usuário insere endereço IP e porta da partida para realizar a conexão com seu oponente.

3. O sistema conecta o usuário à partida do oponente.

**Extensões:**

\*.a Caso a conexão de internet falhe para o jogador ou para o oponente.

1. O sistema informa ao jogador e oponente que houve uma falha de conexão.
2. O sistema pede que ambos tentem novamente.

1.a Caso o sistema não encontre a partida.

1. O usuário é notificado que a partida não foi encontrada.

**Tecnologia:**

O usuário e o oponente devem se conectar via internet (IP:Porta).

Teclado para digitar as informações e realizar a conexão.

**Frequência de ocorrência:** Sempre que usuário utilizar o sistema.

### **2.1.8 Visualizar acervo de cartas do jogo.**

**Escopo:** Jogo de cartas

**Nível:** Meta de usuário

**Ator primário:** Usuário

**Stakeholders e interesses:**

Usuário: deseja ver todas as cartas disponíveis no jogo.

**Fluxo básico:**

1. O jogador abre o acervo de cartas.
2. O sistema apresenta todas as cartas disponíveis no jogo.
3. O jogador seleciona uma carta.
4. Uma representação ampliada da carta é mostrada junto com seu tipo, nome e descrição. Cada carta também pode possuir características especiais que dependem de seu tipo e deverão constar nas regras de domínio.

*Os passos 3-4 se repetem quantas vezes o jogador quiser.*

5. O jogador fecha o acervo de cartas.

**Requisitos especiais:** As cartas devem ser apresentadas em forma de lista, e as principais características de cada carta deve ser apresentada de forma visível ao jogador.

**Tecnologia:**

O sistema usará a engine Unity para apresentar a interface gráfica.

Mouse para selecionar as cartas.

**Frequência de ocorrência:** Ocorre com pouca frequência.

### **2.1.9: Filtrar acervo de cartas**

**Escopo:** Jogo de cartas

**Nível:** Meta de usuário

**Ator primário:** Usuário

**Stakeholders e interesses:**

Usuário: deseja ver todas as cartas disponíveis no jogo com uma determinada característica.

**Pré-condições:** Estar dentro do acervo de cartas.

**Pós-condições:** Apresentar resultados coerentes com a pesquisa.

**Fluxo básico:**

1. O jogador seleciona um ou mais filtros pré-determinados com uma das características das cartas (elemento, letra inicial, habilidade especial, tipo de carta).

2. O sistema busca todas as cartas com a característica selecionada.

3. O sistema apresenta todas as cartas que possuem a característica filtrada.

**Requisitos especiais:** As cartas devem ser apresentadas da mesma forma que no acervo de cartas.

**Tecnologia:** O sistema usará a engine Unity para apresentar a interface gráfica. Mouse para selecionar as cartas e monitor para visualizar as imagens.

**Frequência de ocorrência:** Ocorre com pouca frequência

### **2.1.10: Montar Baralho**

**Escopo:** Jogo de cartas

**Nível:** Meta de usuário

**Ator primário:** Usuário

**Stakeholders e interesses:**

1. Usuário: deseja montar um baralho pra jogar.

**Pré-condições:** Jogador deverá estar logado. (Logar no Sistema).

**Fluxo básico:**

1. O usuário acessa o sistema para montar cartas.

2. O sistema exibe todas as cartas disponíveis no acervo.

3. O Jogador escolhe uma carta do acervo.

4. O Jogador adiciona a carta do acervo ao baralho.

Os passos 3-4 se repetem enquanto o sistema permitir ou baralho não tiver tamanho mínimo especificado nas regras de domínio.

5. O jogador finaliza o baralho.

6. O baralho é validado pelo sistema: Inclui Validar Baralho.

7. O jogador salva o baralho.

**Extensões:**

3.a O jogador escolhe uma carta do seu baralho.

1. A carta escolhida pelo jogador é removida do baralho.

4.a A quantidade máxima foi atingida.

1. O sistema recusa a adição da carta.

2. O usuário é notificado que seu baralho está completo.

5-6a. O sistema invalida o baralho.

1. O jogador é notificado sobre o motivo da invalidação.

2. O jogador modifica seu baralho.

7.a O jogador escolhe não salvar o baralho.

1. Baralho não é salvo.

**Requisitos especiais:** As cartas do acervo devem ser apresentadas de forma que suas principais características sejam visíveis ao usuário.

**Tecnologia:**

O sistema usará a engine Unity para tratar controle do usuário via teclado e mouse e apresentar a interface gráfica e som.

Mouse para selecionar as cartas.



**Frequência de ocorrência:** Ocorre com pouca frequência. Normalmente depois de alguma partida.

#### **2.1.11: Validar baralho**

**Escopo:** Sistema

**Nível:** Sub-função. Parte do caso de uso Montar Baralho.

**Ator primário:** Sistema.

**Stakeholders e interesses:**

Sistema: Deve conferir se o baralho criado é válido.

**Pré-condições:** Usuário tentou criar baralho.

**Pós-condições:** O baralho é validado ou invalidado.

**Fluxo básico:**

1. O sistema verifica a quantidade máxima de cartas.
2. O sistema verifica se não existem muitas cartas repetidas.
3. O sistema valida o baralho.

**Extensões:**

- 1.a A quantidade de cartas no baralho é insuficiente.
  1. O sistema invalida o baralho.
  2. O sistema notifica o jogador de que a quantidade de cartas é insuficiente.
  3. O sistema pede que jogador tente criar um baralho diferente.
- 2.a Há muitas cartas repetidas.
  1. O sistema invalida o baralho.
  2. O sistema notifica o jogador de que há muitas cartas repetidas no baralho.
  3. O sistema pede que jogador tente criar um baralho diferente.

**Frequência de ocorrência:** Sempre que um baralho for criado.

#### **2.1.12: Jogar Partida**

**Escopo:** Jogo de cartas

**Nível:** Meta de usuário

**Ator primário:** Jogador

**Stakeholders e interesses:**

Jogador: deseja jogar contra o oponente.

Oponente: deseja jogar contra o jogador.

**Pré-condições:** Jogador e oponente devem estar conectados à mesma partida. (Conectar a um oponente), (Inclui Montar Baralho).

**Pós-condições:** Existe um vencedor.

**Fluxo básico:**

1. Sistema decide qual jogador iniciará a partida.
2. Sistema informa ambos jogadores qual jogador iniciará a partida.
3. Sistema automaticamente distribui as cartas para cada jogador, sendo que as cartas provêm de seus respectivos baralhos.
4. Sistema exibe na tela pontos de vida e quantidade de cartas na mão de ambos os jogadores.
5. O sistema automaticamente compra uma carta para o jogador no início de seu turno.
6. O jogador escolhe uma carta de sua Mão de Cartas.

7. Jogador usa sua carta.
8. Sistema remove carta da Mão de Cartas do jogador e a coloca no Cemitério.
9. Sistema verifica impactos da carta jogada em ambos os jogadores.
10. Sistema atualiza todas as informações alteradas pela carta.
- Repete 5-9 até jogador finalizar sua jogada ou passar a vez.*
11. Jogador finaliza sua jogada.
12. Turno do jogador é encerrado.
13. Turno é cedido ao oponente.
- Os passos 3-12 são repetidos até que haja um vencedor.*
14. A partida termina.
15. Sistema informa vencedor da partida.

### **Extensões:**

\*.a Um dos jogadores desiste

1. A partida termina.
2. O jogador que não desistiu vence a partida.

\*.b Ocorre uma falha na conexão.

1. A partida termina.
2. O jogador que não teve a falha na conexão vence a partida.

\*.c Um dos jogadores encerra o sistema.

1. A partida termina.
2. O jogador que não encerrou o sistema vence a partida.

5.a. Jogador não tem mais cartas no baralho.

1. Vitória do oponente.

7.a. O jogador tenta jogar uma carta inválida.

1. O sistema informa que a carta é inválida.
2. O jogador escolhe outra carta.

5-10.a. O jogador escolhe passar a vez.

1. Jogador não realizará mais nenhuma ação neste turno.
2. O turno é cedido ao oponente.

5-10.b. Oponente contra-ataca.

1. Sistema interrompe turno do jogador sem que este tenha finalizado sua jogada.
2. Sistema remove carta da Mão de Cartas do oponente e a coloca no Cemitério.
3. Sistema verifica impactos da carta jogada em ambos os jogadores.
4. Turno é devolvido ao jogador para que este finalize sua jogada.

### **Requisitos especiais:**

A partida deve ocorrer de acordo com as regras do domínio.

### **Tecnologia:**

O sistema usará a engine Unity para tratar controle do usuário via teclado e mouse e apresentar a interface gráfica e som.

Monitor para visualizar a interface gráfica e desejável alto-falantes para ouvir o som.

Teclado e mouse.

**Frequência de ocorrência:** Com grande frequência.

## **2.2 Iteração I**

2. Visualizar acervo de cartas do jogo.
4. Montar Baralho.
5. Validar baralho.
8. Logar no sistema.
9. Criar conta.
10. Editar baralho.
11. Remover carta do baralho.
12. Adicionar carta ao baralho.

## **2.3 Iteração II**

1. Conectar a um oponente.
3. Filtrar acervo de cartas.
6. Jogar.
7. Criar Partida.

## **3. Requisitos não funcionais**

### **3.1 Usabilidade:**

- Ações de usuário deverão contar com feedback sonoro.
- O usuário deve aprender a jogar de imediato.

### **3.2 Security:**

- O uso do sistema requer autenticação de usuário.

### **3.3 Interface:**

- Toda a interface com o usuário deverá se dar pela engine Unity.
- É necessário ter componentes como monitor, teclado e mouse para poder interagir com o jogo.

### **3.4 Software:**

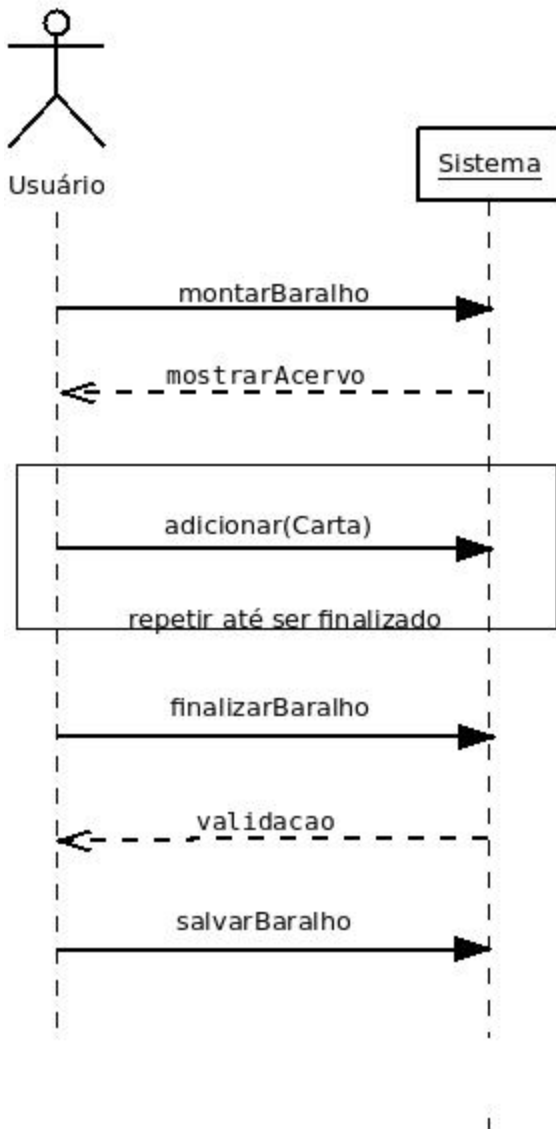
- O sistema deve ser desenvolvido utilizando a linguagem Java
- A interface gráfica será construída usando a engine Unity.
- O sistema deverá funcionar em Linux.

## **4. Diagrama de casos de uso**

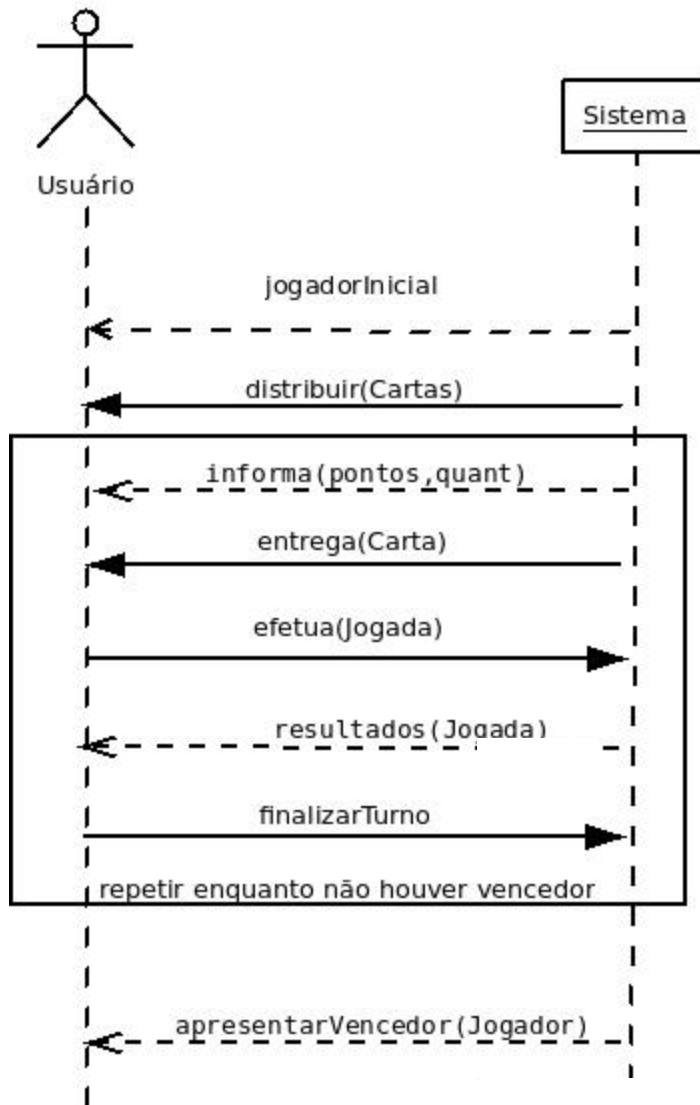


## 5. Diagramas de sequência do sistema

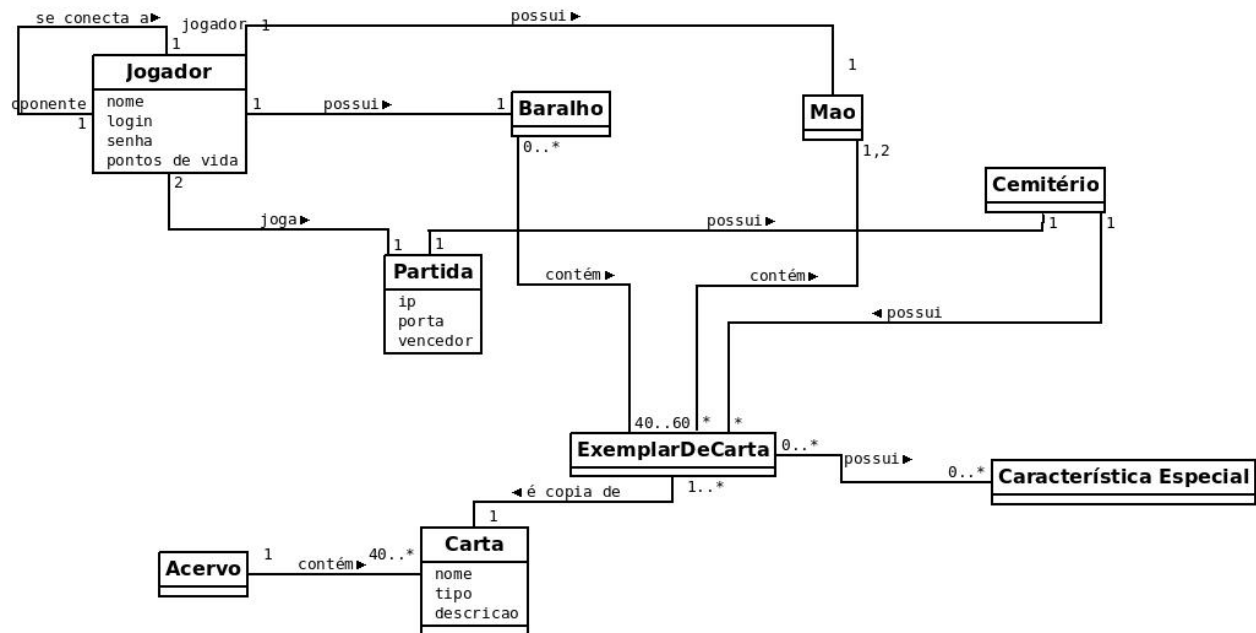
### 5.1 Caso de uso Montar baralho



## 5.2 Caso de uso Jogar



## 6. Diagrama de classes



## 7. Contratos

**Operação:** adicionar(Carta).

**Referências Cruzadas:** Caso de uso Adicionar Carta ao Baralho, diagrama de sequência MontarBaralho.

**Pré-condições :** Nenhuma.

**Pós-condições:** Carta é adicionada ao baralho.

## 8. Glossário

- *Acervo*: Coleção de todas as cartas existentes no jogo.
- *Baralho*: Coleção de todas as cartas que um jogador poderá vir a comprar durante uma partida.
- *Cemitério*: Local do tabuleiro onde ficam as cartas jogadas/descartadas.
- *Comprar uma carta*: Remover a carta do topo do baralho e colocar na mão de cartas.
- *Contra-atacar*: Realizar uma ação sem que seja o turno de um jogador, como resposta a um ataque.
- *IP*: é o endereço do computador de um Jogador.
- *Jogada*: Decisões/ações que um jogador pode tomar em seu turno.
- *Jogador*: um usuário conectado a uma partida.
- *Logado*: usuário conectado ao sistema.
- *Mão de cartas*: Cartas disponíveis para o jogador jogar num dado turno.
- *Mesa*: tabuleiro onde estão as cartas jogadas pelos jogadores.
- *Oponente*: o jogador adversário. Pode ser visto como segundo jogador. Nos casos de uso e diagramas será visto, para fins de entendimento, como um agente externo que usa o sistema apenas para se conectar a uma partida criada pelo usuário. Porém, na prática o oponente será um outro usuário do sistema.

- *Passar a vez:* Abrir mão de seu turno para o oponente.
- *Pontos de vida:* Valor definido pelas regras do jogo. Se zerar o jogador perde o jogo.
- *Porta:* É a porta de conexão, que o Jogador deixou aberta em seu sistema para ser conectado ou a qual o Jogador vai conectar ao Oponente.
- *Turno:* A vez de um jogador jogar.
- *Unity:* Uma ferramenta utilizada para auxiliar o desenvolvimento de jogos.