GEOVANNA DA SILVA LIMA
IGOR DOMINGOS DA SILVA MOZETIC
KLEBER GONÇALVES DE SOUZA
PÂMELA VICTÓRIA DE SOUSA CARVALHO
VANESSA RODRIGUES CARDOSO

AVALIAÇÃO DE LÓGICA 2

Among Us

SÃO PAULO
2020

Sumário

1. JOGO: AMONG US 3

2. CLASSES 3

2.1. Classe Partida 3

2.2. Classe Sala 3

2.3. Classe Controle 4

2.4. Classe Personagem 5

2.5. Classe Impostor 5

2.6. Classe Sabotagem 6

2.7. Classe Tripulante 6

2.8. Classe Associativa Fila da Missão 6

2.9. Classe Missão 7

2.10. Classe Chat 7

3. RELAÇÃO ENTRE CLASSES 7

3.1. Relação entre Sala e Partida. 8

3.2. Relação entre Sala e Controle. 8

3.3. Relação entre Controle e Personagem 9

3.4. Herança entre Personagem, Impostor e Tripulante 9

3.5. Relação entre Impostor e Sabotagem 10

3.6. Relação entre Tripulante e Missões 11

3.7. Relação entre Personagem e Chat 11

4. DIAGRAMA DE CLASSES 12

1. JOGO: AMONG US

Among Us é um jogo eletrônico casual de multijogadores desenvolvido e publicado pelo estúdio de jogos estadunidense InnerSloth. Foi lançado em 15 de junho de 2018 para Android e iOS; e em 17 de agosto de 2018 para Microsoft Windows.

O jogo reúne uma equipe com até 10 pessoas dentro de uma nave, na qual até três integrantes podem ser impostores. Os jogadores precisam realizar a manutenção dessa nave, enquanto os impostores possuem o objetivo de sabotar e assassinar os demais participantes. Ao encontrar um corpo, os jogadores discutem sobre quem foi o responsável no chat e iniciam uma votação de emergência.

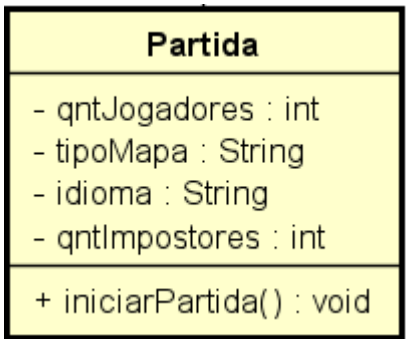
A partida termina quando os impostores são descobertos, todos os objetivos da nave são concluídos ou quando o impostor consegue eliminar todos os adversários.

2. CLASSES

Este tópico visa apresentar as classes que estão presentes no diagrama de classes produzido pelo respectivo grupo.

2.1. Classe Partida

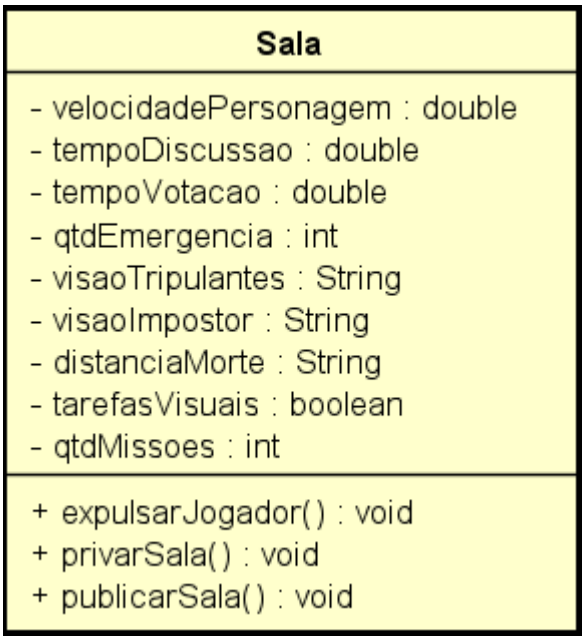
A Classe Partida tem os atributos: “qntJogadores” do tipo Inteiro, “tipoMapa” do tipo String, “idioma” do tipo String e “qntImpostores” do tipo Inteiro. E os determinados métodos: “iniciarPartida” do tipo Void.



2.2. Classe Sala

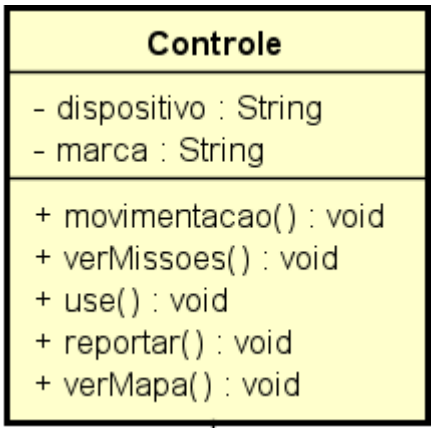
A Classe Sala tem os atributos: “velocidadePersonagem” do tipo Double, “tempoDiscussao” do tipo Double, “tempoVotacao” do tipo Double, “qtdEmergencia” do tipo Inteiro, “visaoTripulantes” do tipo String, “visaoImpostor” do tipo String, “distanciaMorte” do tipo String, “tarefasVisuais” do tipo Boolean (mostra quantas

missões vão aparecer para o jogador), “qtdMissoes” do tipo Inteiro. E os determinados métodos: “expulsarJogador” do tipo Void, “privarSala” do tipo Void, “publicarSala” do tipo Void.



2.3. Classe Controle

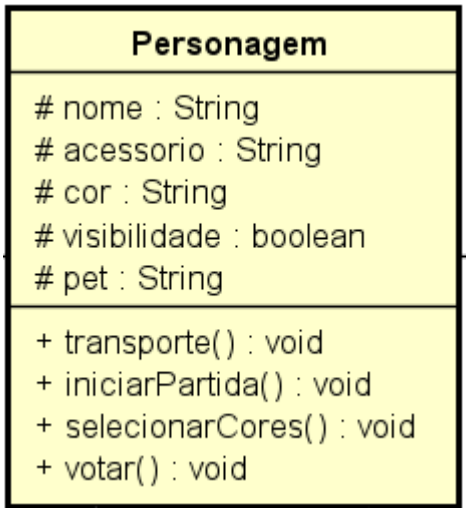
A Classe Controle tem os atributos: “dispositivo” do tipo String, “marca” do tipo String. E os determinados métodos: “movimentacao” do tipo Void, “verMissoes” do tipo Void, “use” do tipo Void (esse seria um botão que é utilizado quando o personagem faz suas missões ou sabotagens), “reportar” do tipo Void e “verMapa” do tipo Void.



2.4. Classe Personagem

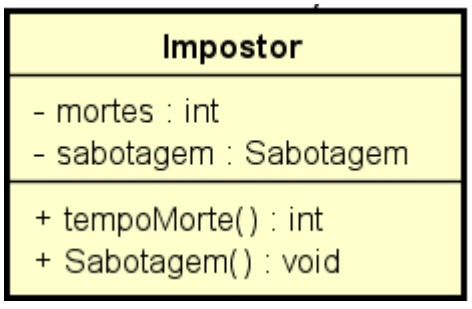
A Classe Personagem contém os atributos “mortes”, “acessorio”, “cor”, e “pet” do tipo String, e “visibilidade” do tipo boolean. Possui os métodos “transporte”, “iniciarPartida”, “selecionarCores”, e “votar”, sendo todos estes do tipo void.

Os métodos que possuem os nomes menos explícitos, são “transporte” e “votar”, onde o primeiro é o método encarregado de permitir que o usuário se mova pelo mapa e o outro permite ao jogador que vote em quem acredite ser o Impostor, com a finalidade de eliminá-lo.



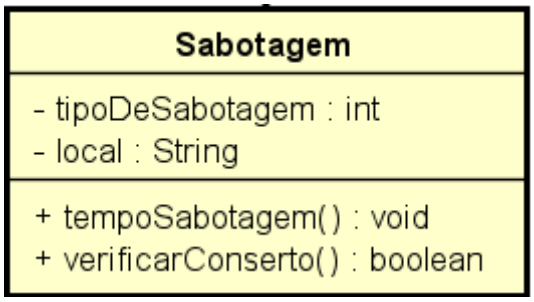
2.5. Classe Impostor

A Classe Impostor apresenta o atributo “mortes” do tipo Inteiro e o atributo “sabotagem” do tipo Sabotagem, além de possuir o método “tempoMorte”, para estabelecer quando o jogador poderá voltar a realizar as mortes, contém também o método “Sabotagem” que representa a sabotagem que o tripulante poderá fazer, respectivamente, os métodos são do tipo Inteiro e do tipo Void.



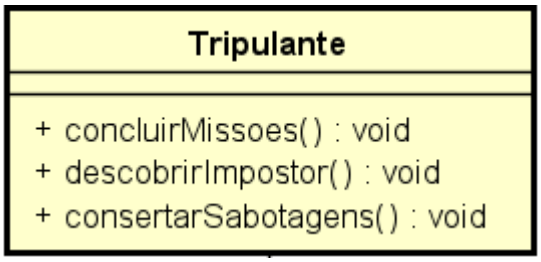
2.6. Classe Sabotagem

A Classe Sabotagem tem os atributos: “tipoDeSabotagem” do tipo Inteiro, “local” do tipo String. E os determinados métodos: “tempoSabotagem” do tipo Void e “verificarConserto” do tipo Boolean (esse método traz “True” ou “False” dependendo do estado da missão (completa ou incompleta)).



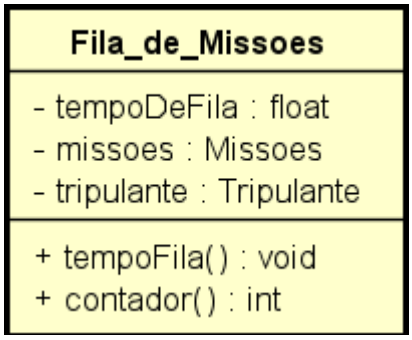
2.7. Classe Tripulante

A classe Tripulante, não dispõe de atributos, pois os mesmos estão contidos na classe Personagem. Todavia, detém os métodos “concluirMissoes”, “descobrirImpostor” e “consertarSabotagens”, ambos do tipo Void.



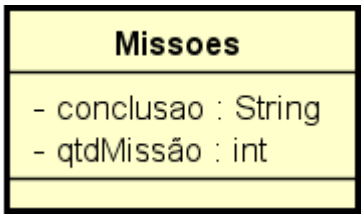
2.8. Classe Associativa Fila da Missão

A classe Fila_de_Missoes possui os atributos “tempoDeFila”, do tipo Float, “missoes” do tipo Missoes, e o atributo “tripulante” do tipo Tripulante. Para mais, inclui os métodos “tempoFila”, que representa a contagem de tempo estabelecida após o tripulante realizar todas as missões, do tipo Void; além do método contador, que irá contar quando o jogador efetuar as missões, do tipo Inteiro.



2.9. Classe Missão

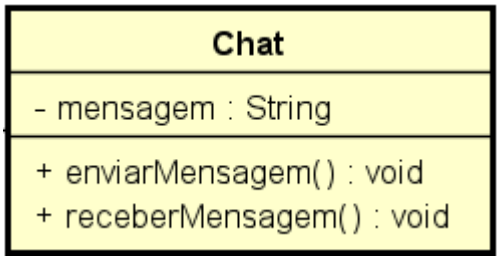
A classe Missoes, conta com os atributos “conclusao” e “qtdMissao”, respectivamente, do tipo String e do tipo Inteiro. Ambos são atributos private.



2.10. Classe Chat

A classe Chat, conta com o atributo “mensagem”, do tipo String.

Possui os métodos “enviarMensagem” e “receberMensagem”, que são incumbidos de enviar e receber as mensagens, ambos são do tipo Void.



3. RELAÇÃO ENTRE CLASSES

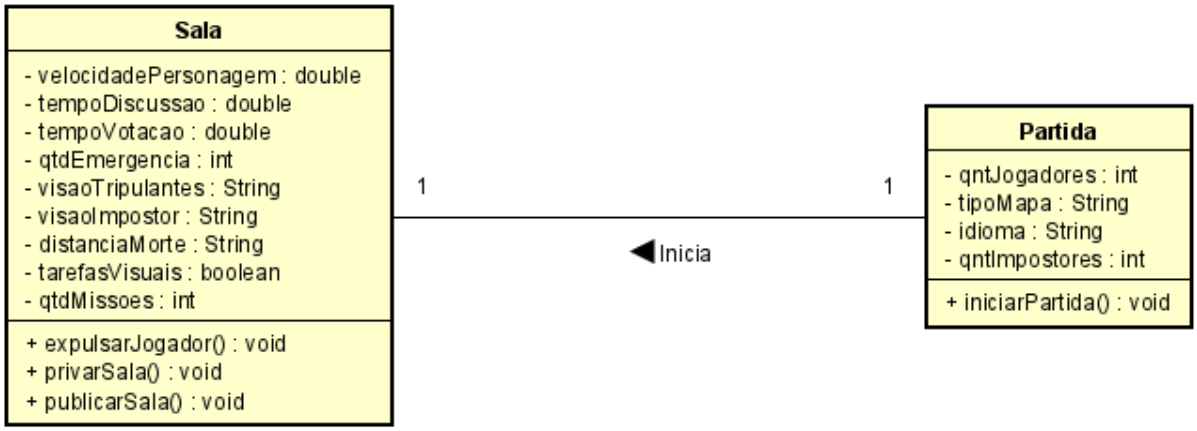
Este tópico visa apresentar as respectivas relações entre as classes presentes no diagrama de classes.

3.1. Relação entre Sala e Partida.

Esta relação apresenta a Classe Sala e a Classe Partida, que apresentam a multiplicidade 1 para 1, representado pelo número um ao lado de cada classe.

Nesse caso, a Partida inicia uma Sala, e uma Sala é iniciada por uma Partida, graças a sua relação de associação, descrita pela linha disposta entre as duas classes.

A palavra “inicia” representa o sentido de leitura da relação.

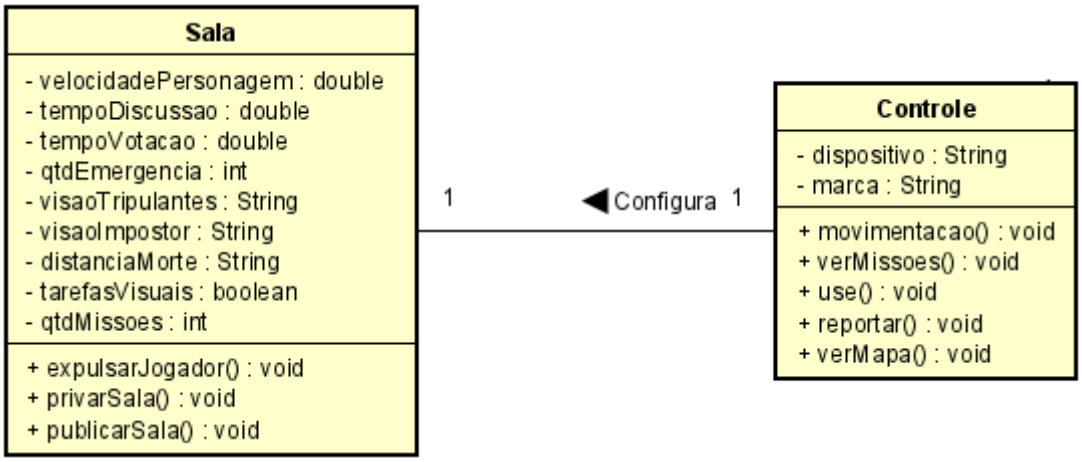


3.2. Relação entre Sala e Controle.

Esta relação apresenta a Classe Sala e a Classe Controle, que possuem a multiplicidade 1 para 1, representada graficamente pelo número 1 ao lado de ambas as classes.

Nessa relação em questão, o Controle configura a Sala, logo, a Sala é configurada pelo Controle. A relação é de associação, representada pela linha sólida entre as classes.

A palavra “configura” indica o sentido de leitura da relação.

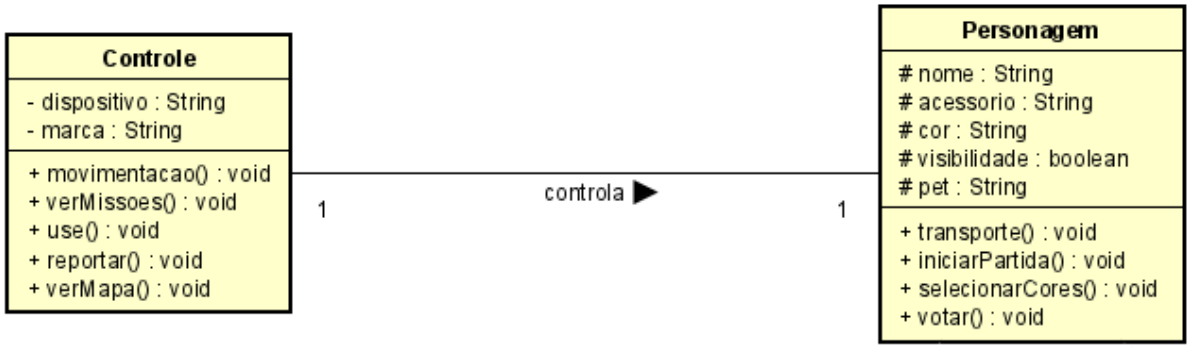


3.3. Relação entre Controle e Personagem

Esta relação apresenta a Classe Controle e a Classe Personagem que apresentam a multiplicidade 1 para 1, representado pelo número um ao lado de cada classe.

Nesse caso, o Controle controla um Personagem, e um Personagem é controlado por um Controle, graças a sua relação de associação, descrita pela linha disposta entre as duas classes.

A palavra “controlar” representa o sentido de leitura da relação.

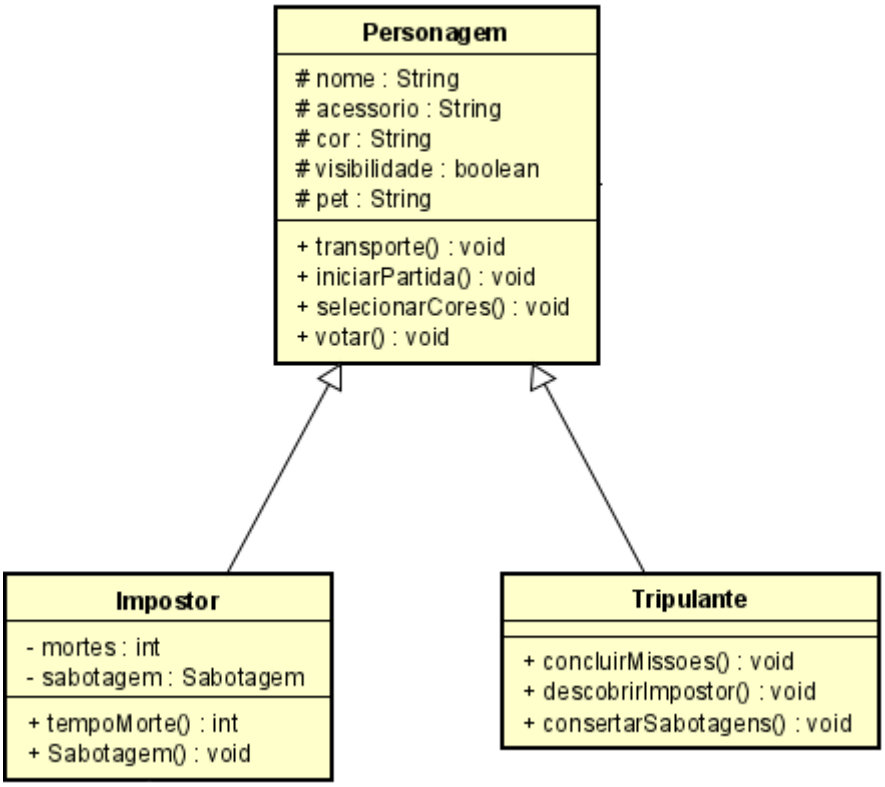


3.4. Herança entre Personagem, Impostor e Tripulante

Esta é uma relação de herança que apresenta a Classe Personagem que se segmenta nas classes Impostor e Tripulante.

Os atributos do Personagem são do tipo protect pois as subclasses (Impostor e o Tripulante), que são especializações, precisam ter acesso aos atributos da superclasse (Personagem), a classe generalista.

Nesse caso, as classes filhas podem acessar todos os atributos da classe Personagem.



3.5. Relação entre Impostor e Sabotagem

Esta relação apresenta a Classe Impostor e a Classe Sabotagem que contem, respectivamente, as multiplicidades 1..3 e 0..*, representadas por estes símbolos ao lado de cada classe.

Nesse caso, o Impostor realiza zero ou muitas Sabotagens, enquanto zero ou muitas Sabotagens são realizadas por um, dois ou três Impostores.

A relação existente entre ambas as classes é a de associação por agregação, descrita pela linha disposta entre as duas classes e seu losango sem preenchimento em uma das pontas.

A palavra “realiza” representa o sentido de leitura da relação.

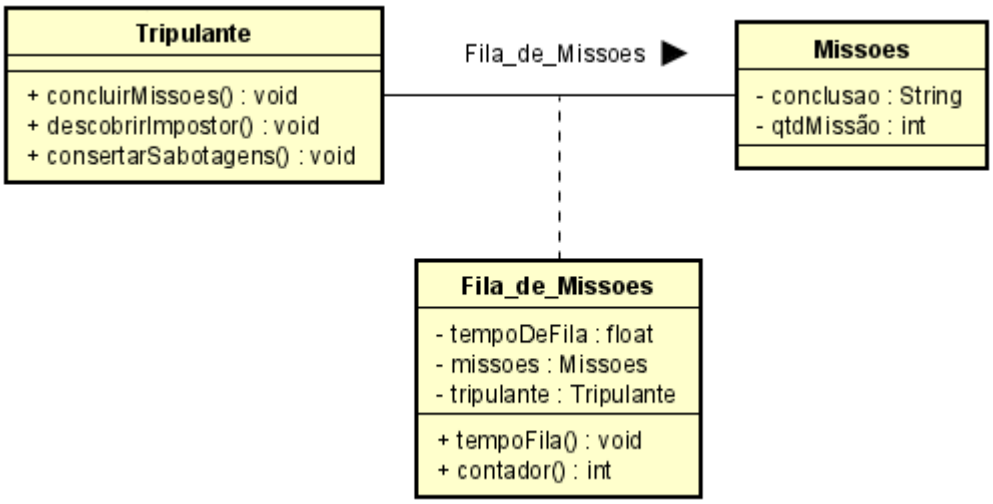
3.6. Relação entre Tripulante e Missões

Esta relação apresenta a Classe Tripulante e a Classe Missões que inclui, uma multiplicidade de muitos para muitos, por esse motivo foi criada uma classe associativa chamada Fila_de_Missoes, a mesma tem como objetivo determinar a sequência de missões que o tripulante deve exercer, quando essas acabam, uma contagem de tempo se inicia, e ao fim dela, chegam novas missões.

Um ou muitos Tripulantes fazem zero ou muitas Missões, e zero ou muitas Missões são feitas por um ou muitos Tripulantes.

A Classe Associativa traz atributos do tipo Tripulante e do tipo Missões, e um novo atributo que determina o tempo de espera de fila.

A palavra “Fila_de_Missoes” representa o sentido de leitura da relação.

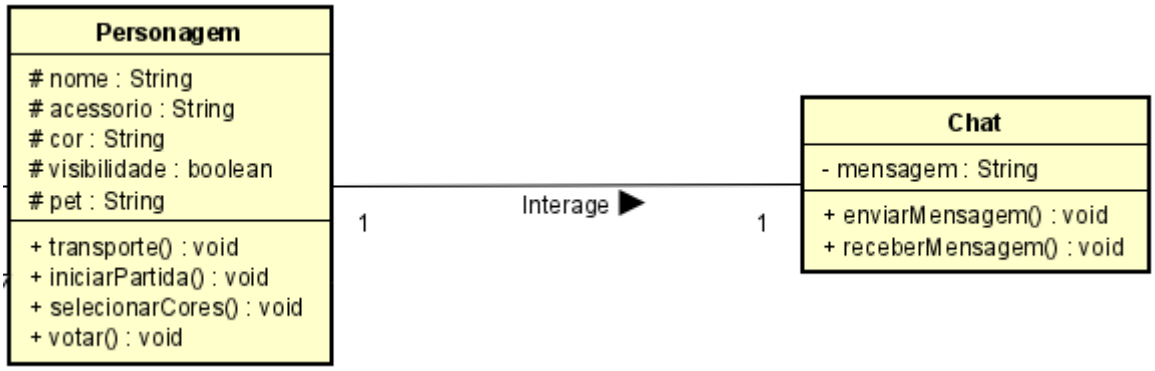


3.7. Relação entre Personagem e Chat

Esta relação apresenta a Classe Personagem e a Classe Chat que apresentam a multiplicidade 1, representado pelo número um ao lado de cada classe.

Nesse caso, o Personagem interage com um Chat, e Chat possui Interação por um Personagem, graças a sua relação de associação, descrita pela linha disposta entre as duas classes.

A palavra “Interage” representa o sentido de leitura da relação.



4. DIAGRAMA DE CLASSES