

**DOCUMENTO DE REQUISITOS**  
**SISTEMA DE COMPRA DE JOGOS**

## Sumário

<b>1</b>	<b>MODELAGEM DO DOMÍNIO DO PROBLEMA.</b>	<b>3</b>
1.1	REFERÊNCIAS.....	3
1.2	Linguagem.....	3
1.3	DESCRIÇÃO DO SISTEMA.....	3
<b>2</b>	<b>REQUISITOS.....</b>	<b>5</b>
2.1	REQUISITOS FUNCIONAIS.....	5
2.2	REQUISITOS NÃO-FUNCIONAIS.....	5
2.3	REGRAS DE NEGÓCIO.....	7
<b>3</b>	<b>DIAGRAMAS DE ATIVIDADE UML.....</b>	<b>8</b>
<b>4</b>	<b>DIAGRAMAS DE CASO DE USO.....</b>	<b>15</b>
<b>5</b>	<b>DIAGRAMA DE CLASSE DE DOMÍNIO.....</b>	<b>16</b>
<b>6</b>	<b>DIAGRAMA DE CLASSE DE PROJETO.....</b>	<b>17</b>
6.1	DESCRIÇÃO DE MÉTODOS.....	17
6.2	REPRESENTAÇÃO GRÁFICA.....	18

# **1 MODELAGEM DO DOMÍNIO DO PROBLEMA**

## **1.1 REFERÊNCIAS**

Steam, Epic Games Store e Xbox Store

## **1.2 Linguagem**

Esse documento será apresentado em uma linguagem de modo informal, devido ao público alvo do sistema se tratar de pessoas que consomem videogames, que podem ser adolescentes assim facilitando o entendimento.

## **1.3 DESCRIÇÃO DO SISTEMA**

Este sistema é um ambiente de compra de jogos online que visa proporcionar uma experiência amigável e eficiente para os usuários. Ele oferece uma variedade de recursos, desde a busca e compra de jogos até a interação com outros jogadores por meio de avaliações e comentários. Além disso, mantém um registro detalhado de todas as transações realizadas para garantir transparência e segurança.

Para garantir uma experiência de usuário satisfatória, a interface é projetada para ser intuitiva e fácil de usar, permitindo uma navegação suave. O sistema também se preocupa com o desempenho, fazendo uso eficiente de cache e seguindo padrões de desenvolvimento para garantir eficiência e consistência.

Além disso, o sistema é projetado para ser interoperável com outras plataformas de jogos populares, como Steam e Epic Games Store, ampliando as opções de jogos disponíveis para os usuários. No geral, o sistema busca oferecer uma plataforma de compra de jogos online que seja fácil de usar, transparente, segura e ética, promovendo uma experiência positiva para os usuários e respeitando as melhores práticas da indústria de jogos.

## 2 REQUISITOS

### 2.1 REQUISITOS FUNCIONAIS

**[RF1]** O sistema deve possuir a função de comprar jogos.

**[RF2]** O sistema deve possuir um catálogo de jogos que estão à venda. Esse requisito deve estar de acordo com a RN1 e RN3.

**[RF3]** O sistema deve manter um registro de todas as transações executadas. Esse requisito deve estar de acordo com a RN2.

**[RF4]** Os usuários devem poder avaliar e comentar sobre os jogos comprados. Esse requisito deve estar de acordo com a RN4.

**[RF5]** O sistema deve notificar os usuários sobre confirmações de pedidos, atualizações de status de entrega e ofertas especiais.

**[RF6]** O sistema deve fornecer recursos de autoatendimento, como perguntas frequentes (FAQs) e tutoriais. Esse requisito deve estar de acordo com a RN5.

**[RF7]** O sistema deve possuir a função de pesquisar jogos no catálogo.

### 2.2 REQUISITOS NÃO-FUNCIONAIS

**[RNF01]** A interface do sistema deve ser intuitiva e fácil de entender, permitindo fácil navegação. (Requisito de facilidade de uso).

**[RNF02]** O sistema deve fazer uso eficiente de cache para armazenar recursos estáticos, consultas de bancos de dados e outras informações relevantes. (Requisito de desempenho).

**[RNF03]** Toda a instalação do sistema não pode requerer mais que 50 GB de armazenamento. (Requisito de espaço)

**[RNF04]** Toda última segunda-feira do mês os desenvolvedores devem apresentar uma nova versão atualizada do sistema (Requisito de entrega).

**[RNF05]** O sistema utilizará do Git como processo de controle de versão. (Requisito de implementação).

**[RNF06]** O sistema irá seguir padrões da metodologia de cascata, do processo de desenvolvimento de software em V. Que é uma melhoria da cascata. Visando interpolar etapas de planejamento e construção com testes organizados para garantir a consistência do produto. (Requisito de padrões).

**[RNF07]** O sistema deve interoperar com a Steam e Epic Games Store. (Requisito de interoperabilidade).

**[RNF08]** O sistema deve ser transparente sobre práticas de monetização, como microtransações, loot boxes e anúncios in-game, fornecendo informações claras sobre os custos adicionais e que os usuários estejam cientes sobre seus gastos. (Requisito ético).

**[RNF09]** O sistema deve respeitar os direitos autorais e propriedade intelectual de terceiros, garantindo que apenas conteúdo legalmente adquirido seja distribuído. (Requisito legal).

**[RNF10]** Fornecer educação e conscientização aos usuários sobre questões de privacidade, e como controlar suas próprias informações pessoais dentro do sistema. (Requisito de privacidade).

**[RNF11]** O sistema deve utilizar criptografia para proteger dados sensíveis, como informações de pagamentos. (Requisito de segurança).

## 2.3 REGRAS DE NEGÓCIO

**[RN1]** O catálogo de jogos deve ser constantemente atualizado para refletir os produtos em estoque.

**[RN2]** Todas as transações devem ser registradas pelo sistema, incluindo detalhes como data de transação, itens adquiridos, método de pagamento e valor total.

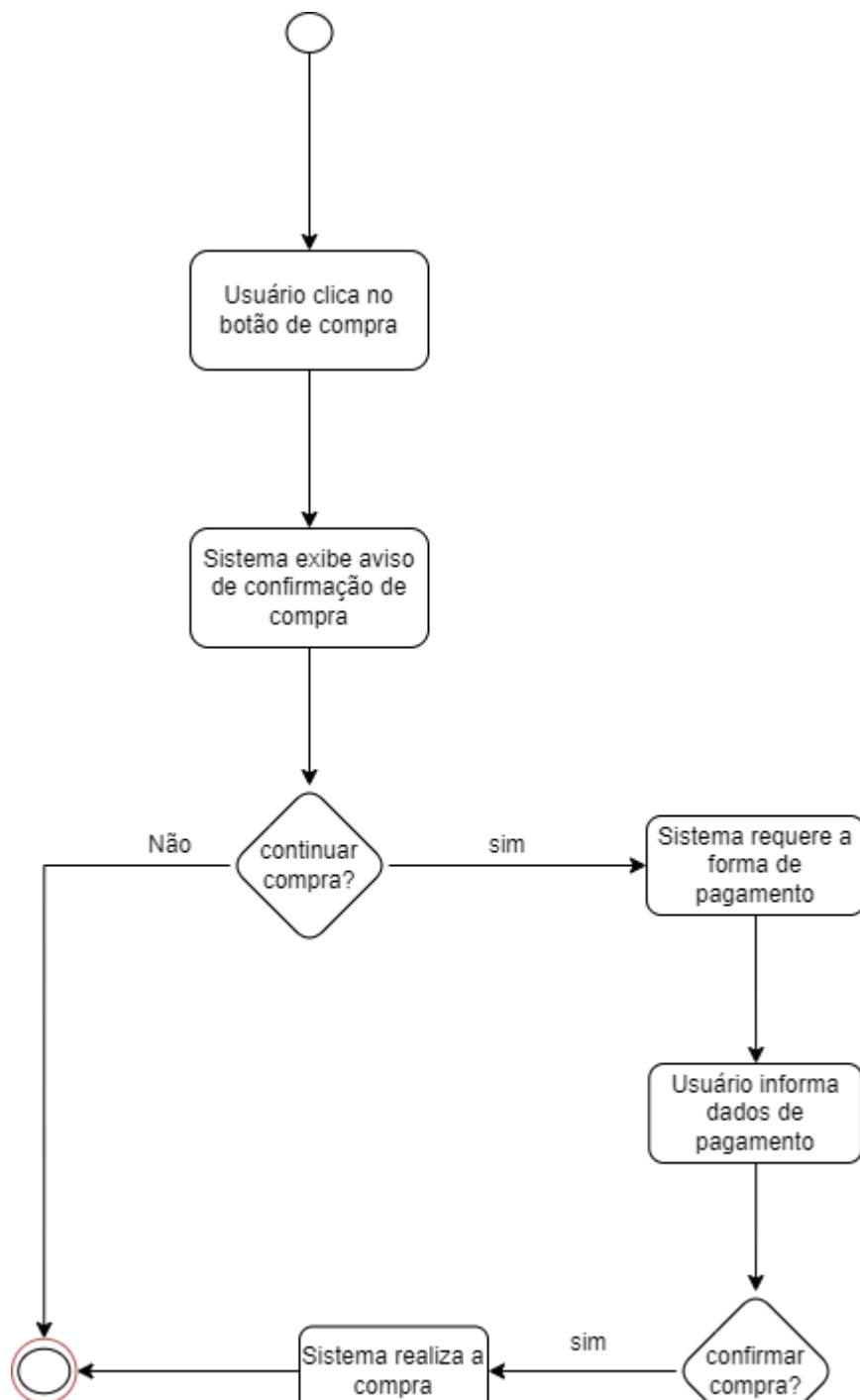
**[RN3]** Jogos removidos pelos desenvolvedores devem ser excluídos do catálogo.

**[RN4]** A aba de avaliação deve ser supervisionada por um bot para evitar possíveis comentários prejudiciais a obra ou a comunidade.

**[RN5]** As abas de FAQs devem possuir funções onde os usuários escolham sobre o que precisam, como jogos, hardwares ou o próprio cadastro.

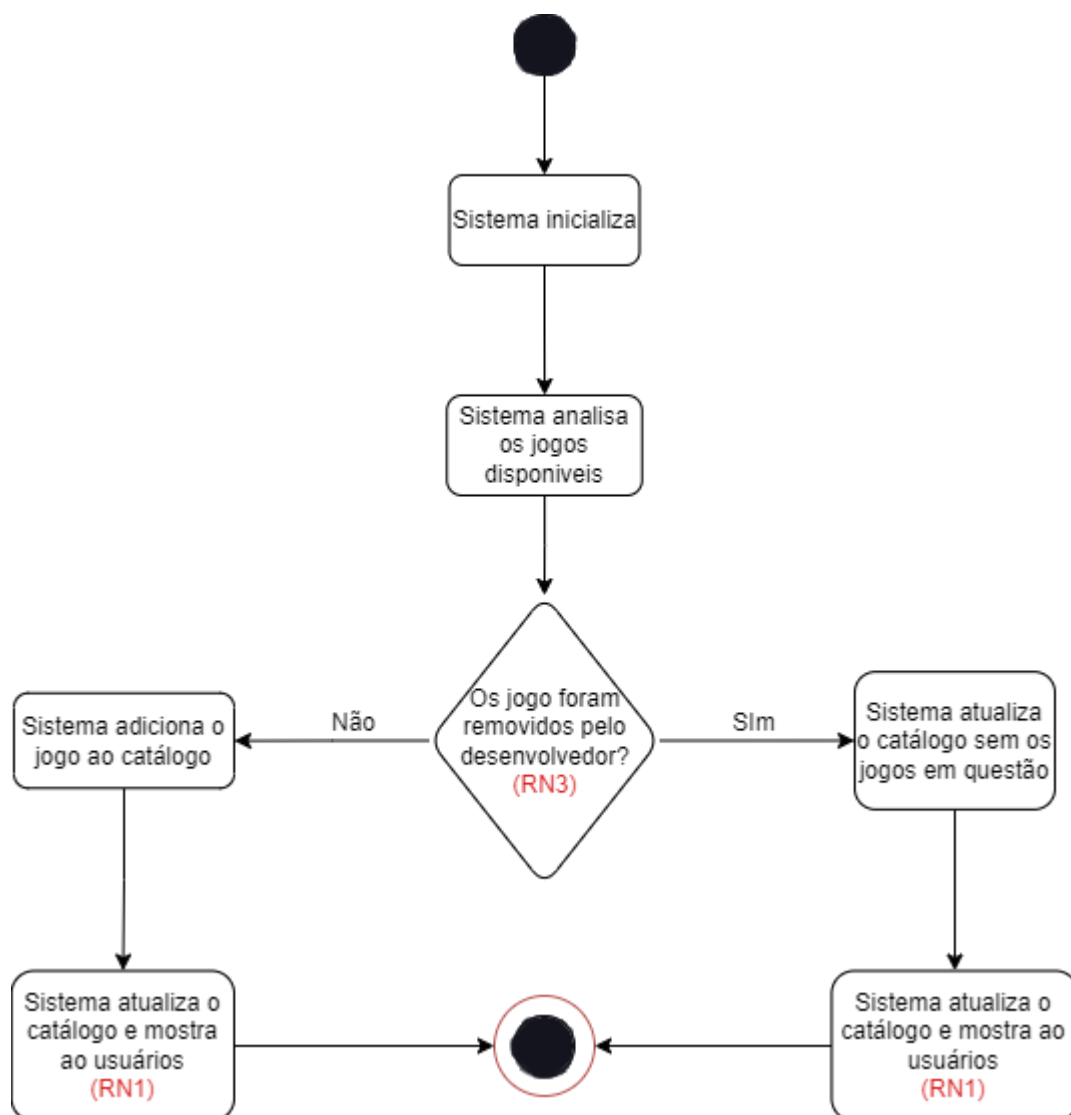
### 3 DIAGRAMAS DE ATIVIDADE UML

**[RF1]** O sistema deve possuir a função de comprar jogos.

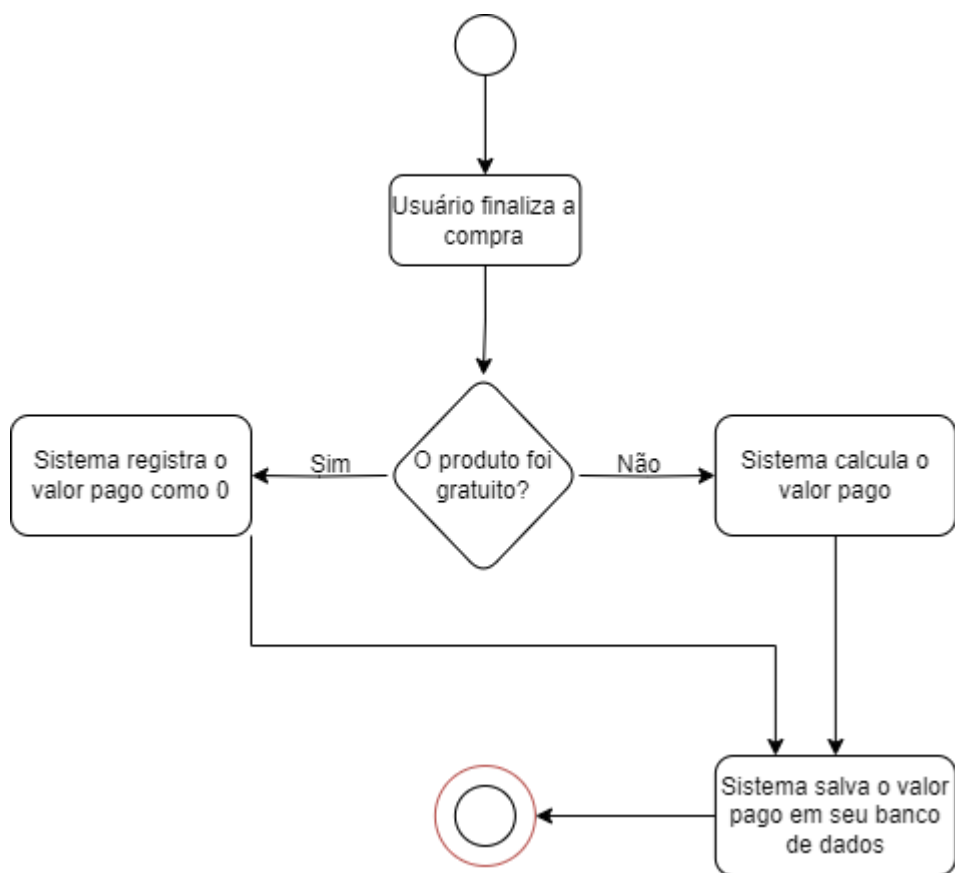




**[RF2]** O sistema deve possuir um catálogo de jogos que estão à venda. Esse requisito deve estar de acordo com a RN1 e RN3.

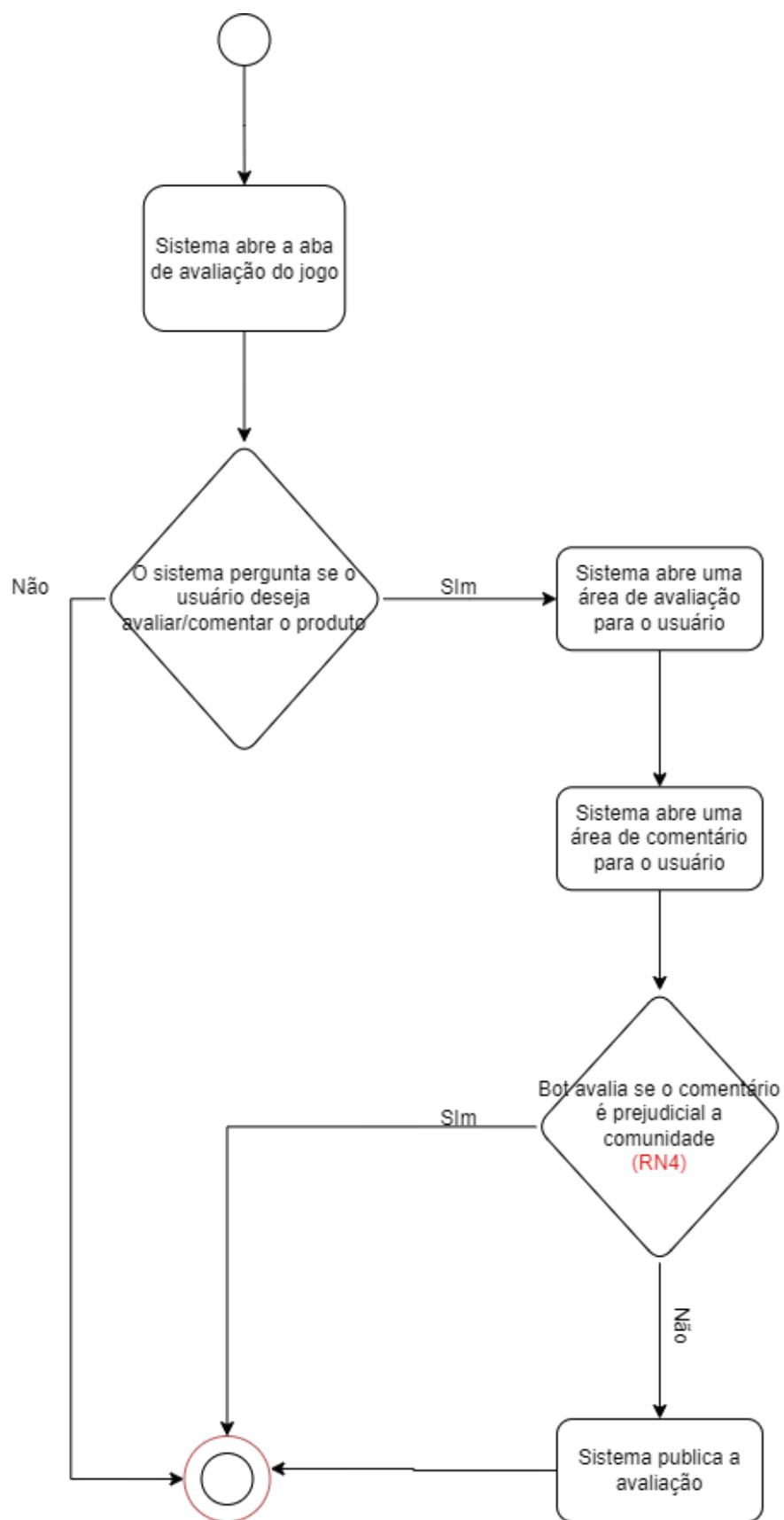


**[RF3]** O sistema deve manter um registro de todas as transações executadas. Esse requisito deve estar de acordo com a RN2.

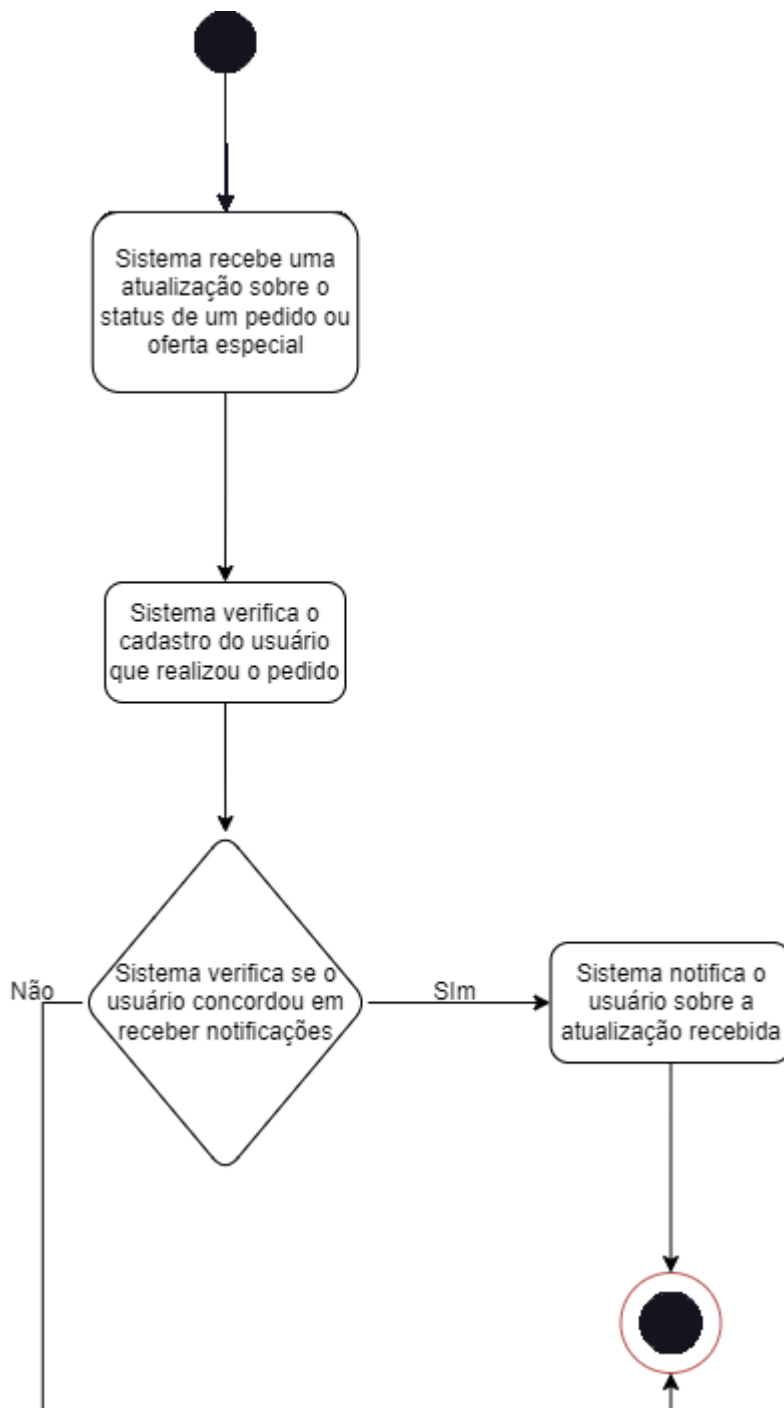


**[RF4]** Os usuários devem poder avaliar e comentar sobre os jogos comprados. Esse requisito deve estar de acordo com a RN4.

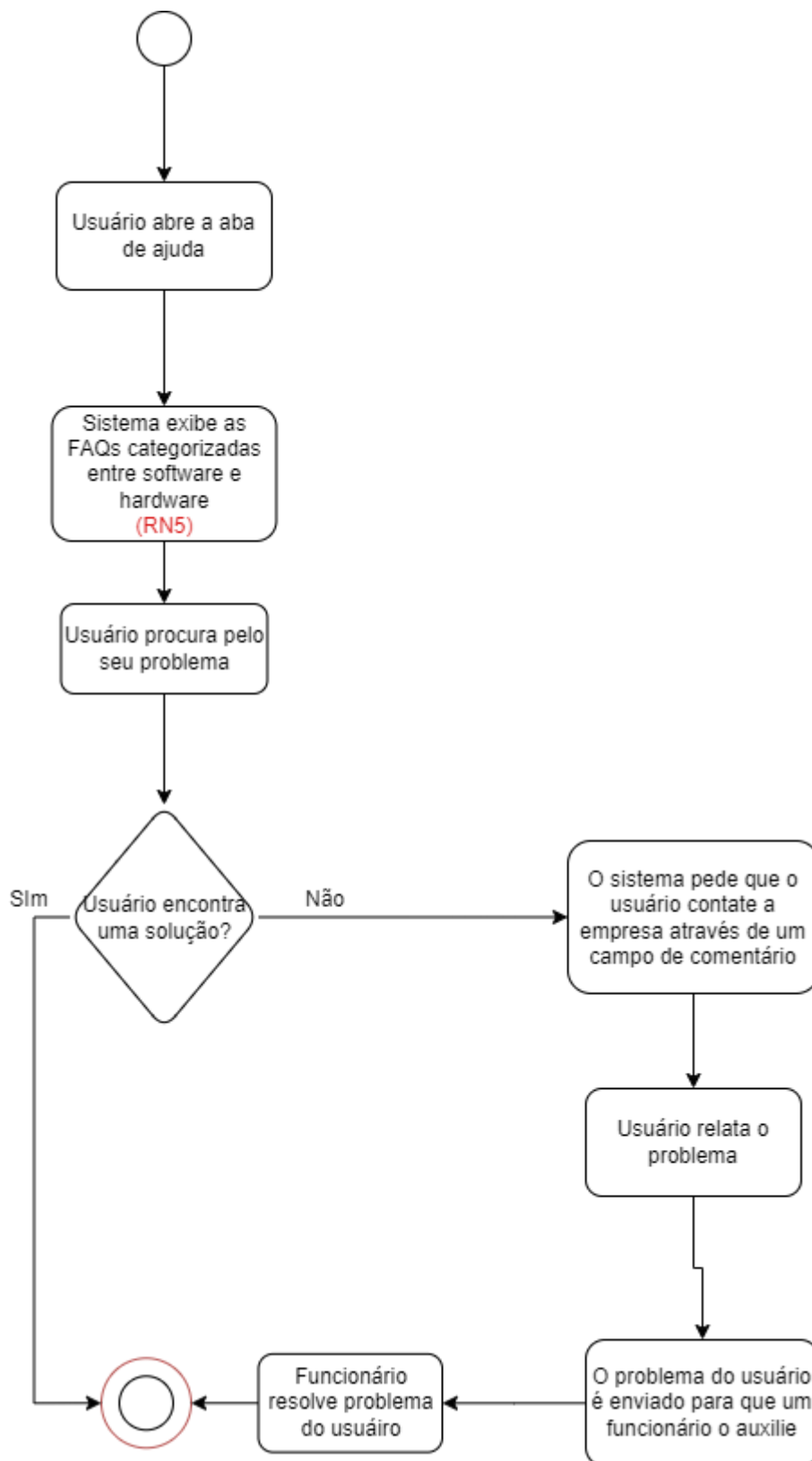




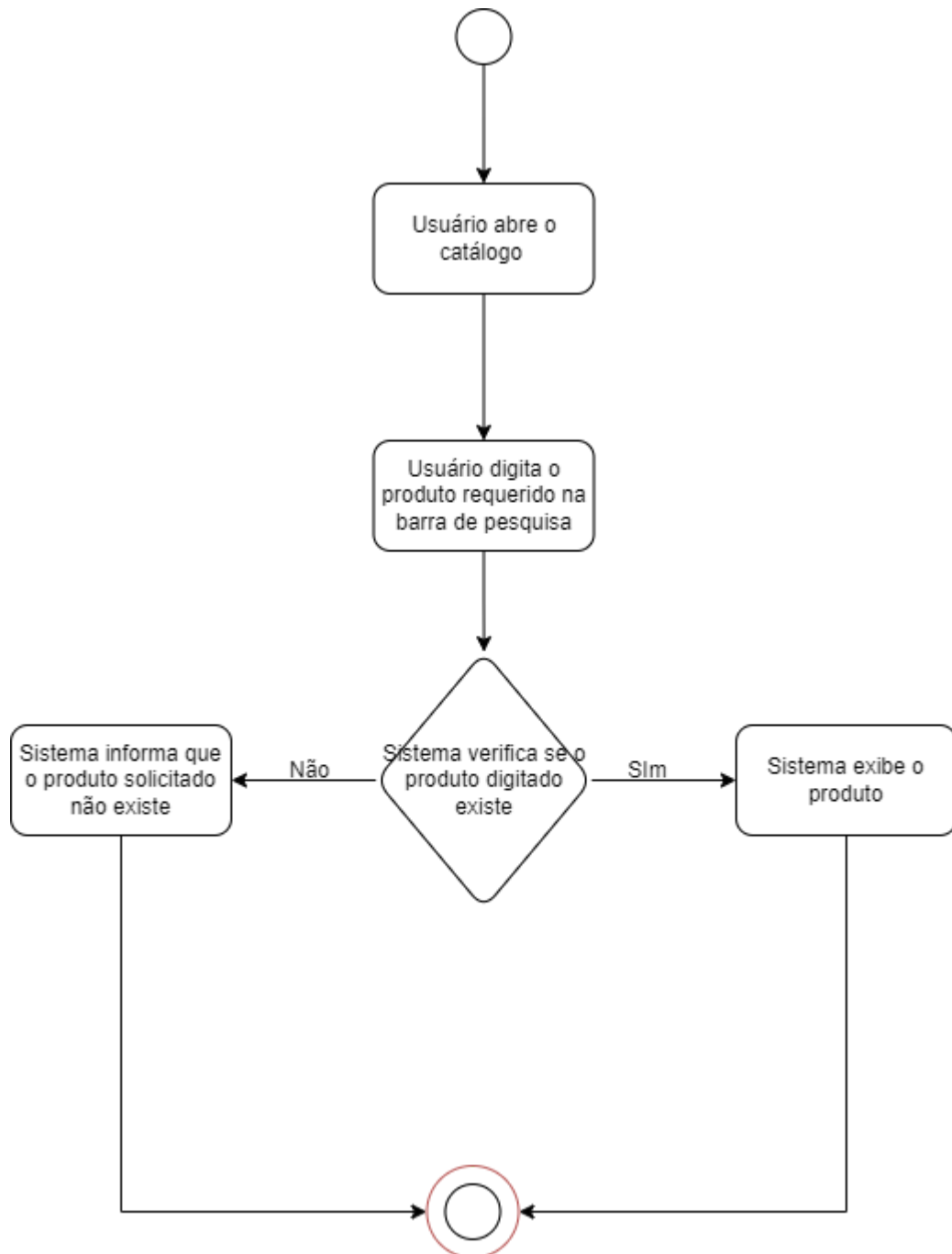
**[RF5]** O sistema deve notificar os usuários sobre confirmações de pedidos, atualizações de status de entrega e ofertas especiais.



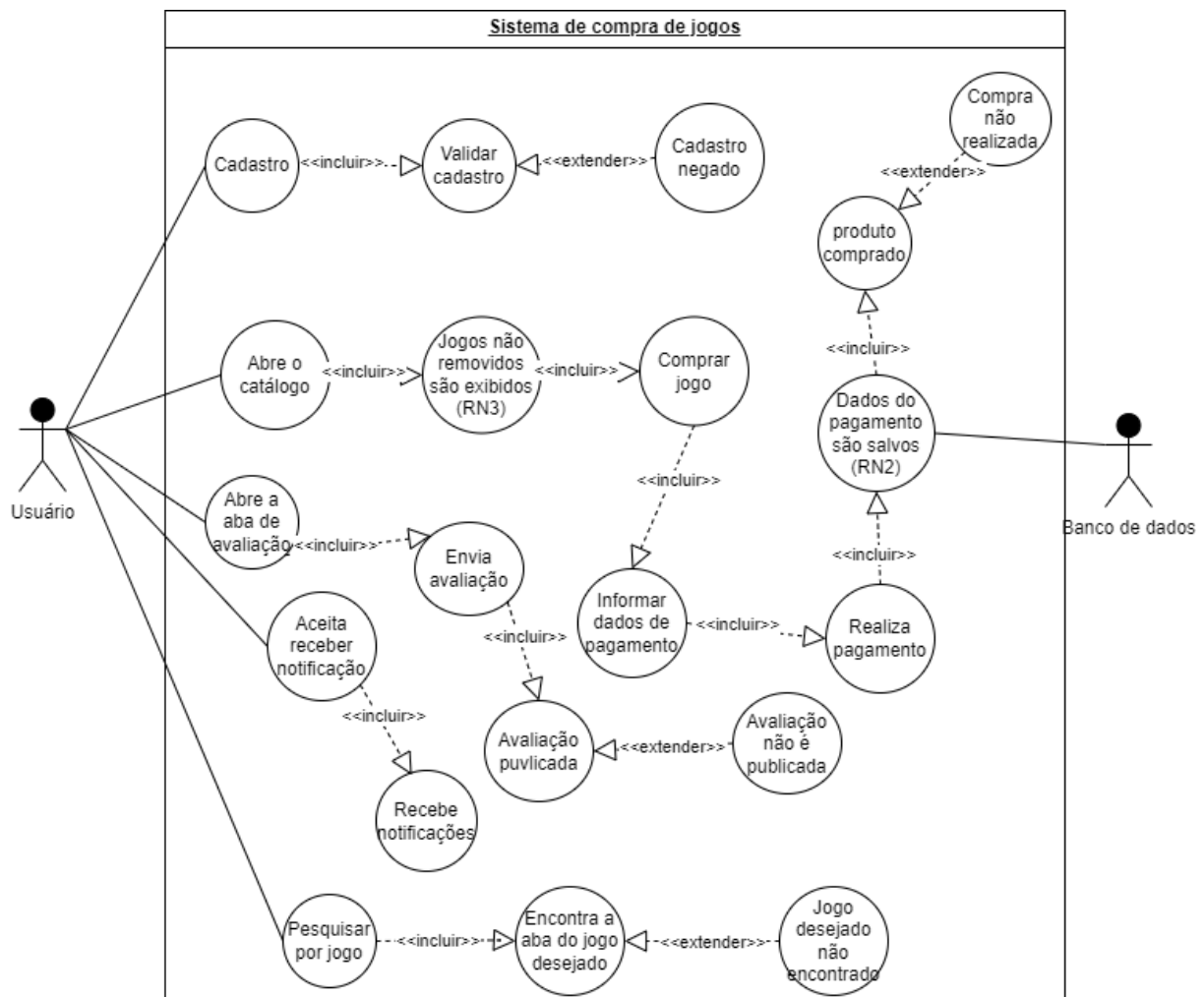
**[RF6]** O sistema deve fornecer recursos de autoatendimento, como perguntas frequentes (FAQs) e tutoriais. Esse requisito deve estar de acordo com a RN5.



**[RF7]** O sistema deve possuir a função de pesquisar jogos no catálogo.

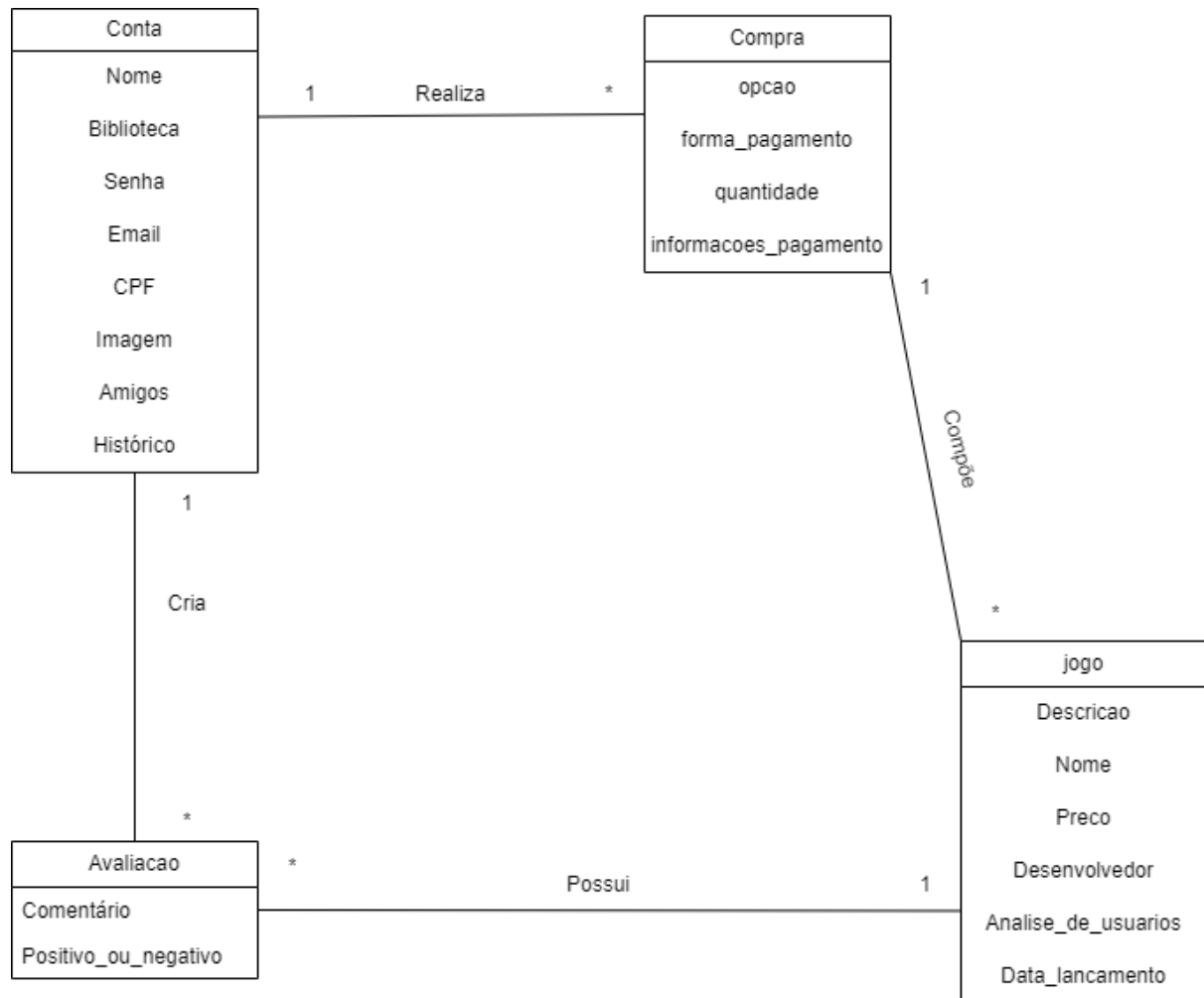


## 4 DIAGRAMAS DE CASO DE USO





## 5 DIAGRAMA DE CLASSE DE DOMÍNIO



## 6 DIAGRAMA DE CLASSE DE PROJETO

### 6.1 DESCRIÇÃO DE MÉTODOS

Métodos da classe Conta:

- AcessarBiblioteca(): Abre a biblioteca da conta atual.
- AcessarHistorico(): Mostra o histórico da conta atual.
- ExibirAmigos(): Mostra os amigos adicionados.
- FazerLogin(nome, senha): Permite acessar uma conta já criada através dos parâmetros nome e senha.
- AdicionarAvaliacao: Permite o usuário realizar uma avaliação através do parâmetro ava.

Métodos da classe Compra:

- AdicionarJogo(Jogo jogo): Adiciona um novo jogo à compra atual.
- RemoverJogo(Jogo jogo): Remove um jogo da compra atual.
- RealizarCompra(): Finaliza a compra.
- CalcularTotal(): Calcula o valor a ser pago pela compra.
- AdicionaraBiblioteca(): Adiciona os jogos comprados à biblioteca do usuário.

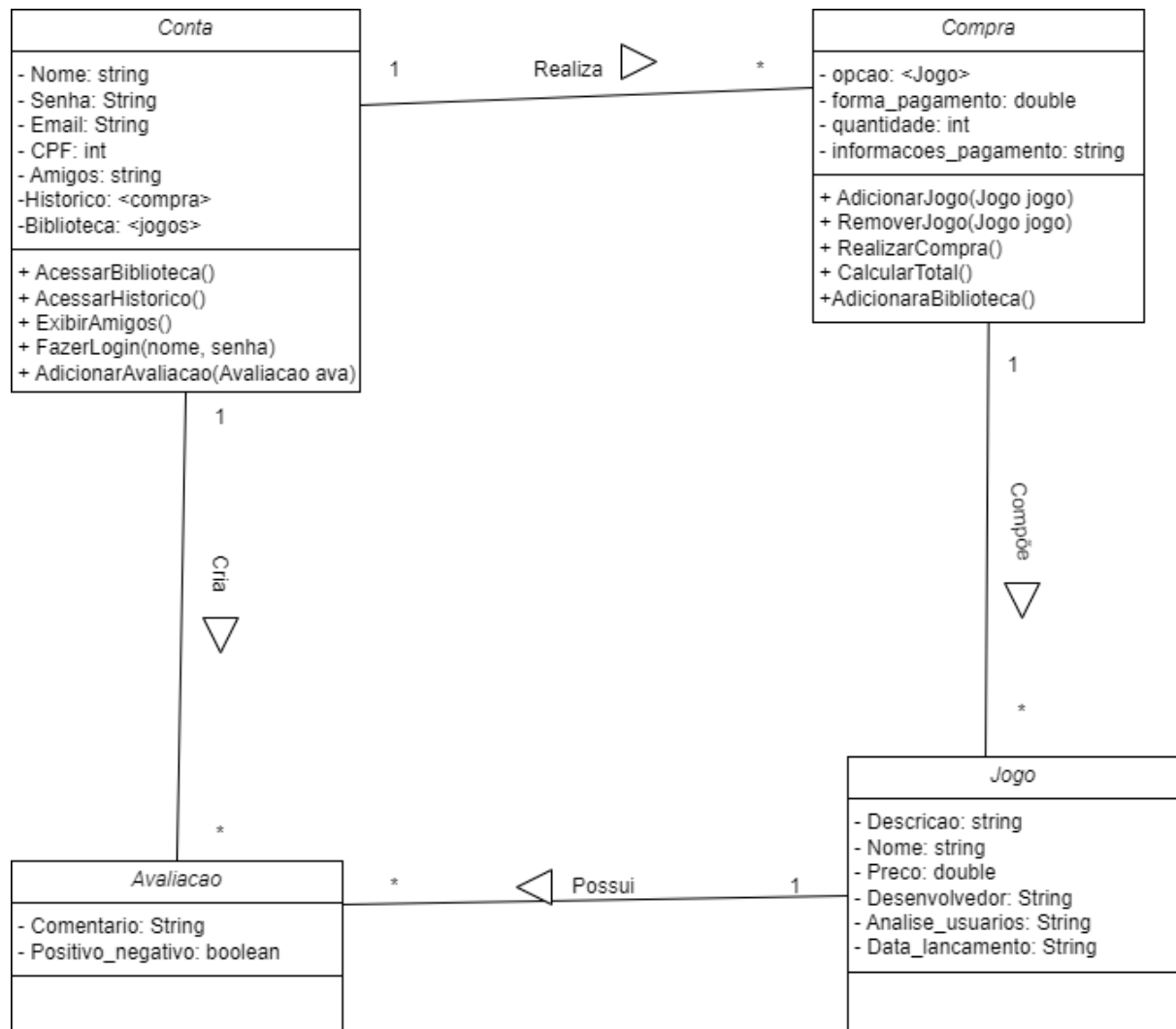
Métodos da classe Avaliação:

Essa classe não possui métodos.

Métodos da classe Jogo:

Essa classe não possui métodos.

## 6.2 REPRESENTAÇÃO GRÁFICA



## 7 DIAGRAMA DE OBJETOS

