

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO AMAZONAS**  
**IFAM CAMPUS MANAUS ZONA LESTE**

**DOCUMENTAÇÃO TÉCNICA DO SOFTWARE TIC-TAC-TOE**

*Versão 1.4*

**Professor:**

Gabriel Pinheiro Compto

**Autores:**

Antônio Hárliton Martins de Souza  
Caio Cesar Faneco Gonzaga  
Josué Alexandre Ferreira  
Raimundo Dione Álvaro de Araújo  
Nelson Thiago Da Silva Pinto

**MANAUS**  
**2023**

**Antônio Hárliton Martins de Souza**  
**Caio Cesar Faneco Gonzaga**  
**Josué Alexandre Ferreira**  
**Nelson Thiago Da Silva Pinto**  
**Raimundo Dione Álvaro de Araújo**

## **DOCUMENTAÇÃO TÉCNICA DO SOFTWARE JOGO TIC-TAC-TOE**

Trabalho de processo e paradigmas de linguagens de programação submetido ao curso de Engenharia de Software do Campus Zona Leste do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas com o propósito de corresponder aos critérios avaliativos propostos pelo professor no momento da solicitação do mesmo.

Prof. Gabriel Pinheiro Compto, Me.

**MANAUS**  
**2023**

## SUMÁRIO

<b>VERSÕES ANTERIORES.....</b>	<b>4</b>
<b>1. INTRODUÇÃO.....</b>	<b>5</b>
1.1. Organização do Projeto.....	5
1.2. Objetivos deste documento.....	5
1.3. Missão do produto.....	5
<b>2. REQUISITOS.....</b>	<b>6</b>
2.1. Requisitos Funcionais.....	6
2.2. Requisitos Não-Funcionais.....	7
2.3. Requisitos de Domínio (Regras de Negócio).....	8
<b>3. ATORES.....</b>	<b>9</b>
<b>4. DIAGRAMAS DE CASOS DE USO.....</b>	<b>10</b>
<b>5. ESPECIFICAÇÃO DE CASOS DE USO.....</b>	<b>11</b>
<b>6. DIAGRAMA DE CLASSE.....</b>	<b>17</b>
<b>7. INTERFACES.....</b>	<b>18</b>
<b>8. REFERÊNCIAS.....</b>	<b>20</b>

## VERSÕES ANTERIORES

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autores</b>
30/05/2023	0.1a	Definição da organização e elaboração da etapa introdutória do projeto.	Equipe
31/05/2023	0.2a	Elaboração dos requisitos funcionais, não funcionais e regras de domínio.	Equipe
06/06/2023	0.3a	Definição dos atores e início do desenvolvimento do software.	Equipe
15/06/2023	0.4a	Criação do primeiro protótipo e início do desenvolvimento dos diagramas.	Equipe
17/06/2023	0.5a	Correção de inconsistência no processo de transição de telas e lançamento de nova versão.	Equipe
20/06/2023	0.6b	Lançamento do 2º protótipo, finalização do diagrama e das descrições de caso de uso	Equipe
21/06/2023	0.7b	Finalização do diagrama de classe	Equipe
23/06/2023	0.8b	Implementação da funcionalidade de jogo contra a máquina e lançamento do último protótipo	Equipe
26/06/2023	0.9b	Finalização da documentação e correção da obtenção de pontos do modo PLAYER x COM.	Equipe
27/06/2023	1.0	Lançamento do Software	Equipe

*Tabela 1 - Registro de Alterações.*

## **1. INTRODUÇÃO**

### **1.1. Organização do Projeto**

Inicialmente será descrito o domínio do problema (Tópico 1) e as dificuldades enfrentadas que resultaram na necessidade de criação do software. Posteriormente são apresentados os requisitos Funcionais, Não-funcionais e de Domínio (Tópico 2), que descrevem as funções e seus parâmetros de execução, no (Tópico 3) serão apresentados os atores envolvidos no software e suas respectivas funções, no (Tópico 4) constará os diagramas de casos de uso do software, já no (Tópico 5) terá as especificações dos casos de uso e no (Tópico 6) os diagramas de classe, além disso, a documentação contará com imagens da interface final do software, servindo de referência para a utilização dos usuários (Tópico 7), e por fim será apresentado as referências para a elaboração desta documentação (Tópico 8).

### **1.2. Objetivos deste documento**

O presente documento tem o propósito de auxiliar o gerente de projetos e sua equipe na organização e direcionamento do desenvolvimento do jogo denominado Tic-Tac-Toe. A seguir será abordado informações sobre a aplicação, seus requerimentos e funções, além de apresentar as interações e dependências necessárias para o bom funcionamento do software.

### **1.3. Missão do produto**

O software apresentado busca promover aos usuários uma experiência de entretenimento clássica e construtiva, esta que tornou-se escassa diante da enorme quantidade de produtos no mercado que não trazem benefícios reais a seus usuários e ao invés disso, contribuem para uma utilização mecanizada e repetitiva, além disso, é comum visualizar tais jogos aliados a cenários de utilização desenfreada, mudança de humor e violência, situações que são extintas na solução proposta, vista que este promove uma competição saudável e crítica entre os jogadores, explorando a tomada de decisão lógica para a obtenção da vitória.

Desse modo, será realizado um software de jogo da velha além da elicitação de seus requisitos e levá-lo a um alto grau de abstração necessário para visualização dos problemas e soluções. Isso será possível através de técnicas e métodos baseados em programação orientada a objetos, linguagem Java, conhecimentos em banco de dados e por meio dos softwares NetBeans, Astah UML e HeidiSQL.

## 2. REQUISITOS

A seguir são apresentados os requisitos (Funcionais, Não-Funcionais e de Domínio) que foram identificados para atingir a missão do produto de forma eficiente e eficaz.

### 2.1. Requisitos Funcionais

Identificador	Descrição	Depende de
[RF01] - Iniciar partida Player vs Player	O sistema deve permitir ao jogador iniciar uma nova partida local, onde irá enfrentar um jogador que estará próximo para poder desafiar.	-
[RF02] - Iniciar partida Player vs Computador	O sistema deve permitir aos jogadores iniciar uma nova partida onde poderá jogar contra um oponente controlado pelo próprio jogo (Jogador Computador).	-
[RF03] - Visualizar Classificação	O software deve permitir que os usuários visualizem sua classificação em comparação aos outros usuários.	-
[RF04] - Visualizar Créditos	O software deve apresentar aos usuários uma página contendo os créditos pelo desenvolvimento do jogo.	-

[RF05] - Sair do jogo	O software deve permitir aos jogadores o encerramento da aplicação, assim saindo do jogo.	-
[RF06] - Revanche	O software deve permitir aos jogadores reiniciar partidas entre os mesmos usuários, mantendo os dados de matrículas.	[RF01], [RF02]

## 2.2. Requisitos Não-Funcionais

Identificador	Descrição	Categoria	Depende de
[RNF01]	O jogo deve ter mecanismos de segurança para prevenir trapaças, evitando que os jogadores manipulem o código ou explorem vulnerabilidades para ganhar vantagens injustas.	Segurança	-
[RNF02]	O sistema deve usar as linguagens Java e SQL.	Implementação	-

[RNF03]	O software deve ser desenvolvido de forma responsiva, executável no formato Desktop.	Portabilidade	-
[RNF04]	O banco de dados do software deverá possuir um uptime de 99.7% do tempo de terça a domingo.	Disponibilidade	-
[RNF05]	O software deve possuir interface intuitiva que proporcione ao usuário realizar seus processos sem necessitar de treinamento específico.	Usabilidade	-
[RNF06]	O sistema deverá atender às normas legais, tais como padrões, leis, etc.	Legalidade	-

### 2.3. Requisitos de Domínio (Regras de Negócio)

Identificador	Descrição	Depende de
[RD01]	Em caso de desconexão ou erro inesperado, o software irá realizar o descarte de qualquer informação, não atualizando o banco de dados com as informações da partida.	[RF01], [RF02]



[RD02]	Cada partida não terá limite de tempo, durando até algum jogador ganhar ou todas as posições do tabuleiro forem preenchidas.	[RF01], [RF02]
[RD03]	Somente os jogadores com as dez primeiras colocações poderão ser visualizados no ranking.	[RF01], [RF02]
[RD04]	Somente os usuários previamente cadastrados no banco de dados poderão ter acesso ao jogo.	-
[RD05]	O usuário poderá somente realizar uma ação por turno, respeitando a disponibilidade de “casas”.	[RF01], [RF02]

### 3. ATORES

Os atores identificados no contexto deste projeto estão descritos na Tabela 2. Eles irão realizar a utilização do software de acordo com sua alçada, realizando o download do jogo, inserindo sua matrícula e jogando.

Ator	Descrição
<b>Jogadores</b>	Atores responsáveis pela utilização do serviço do software e jogar o jogo.

*Tabela 2 – Atores.*

#### 4. DIAGRAMAS DE CASOS DE USO

A seguir será apresentado o diagrama de casos de uso dos requisitos anteriormente apontados.

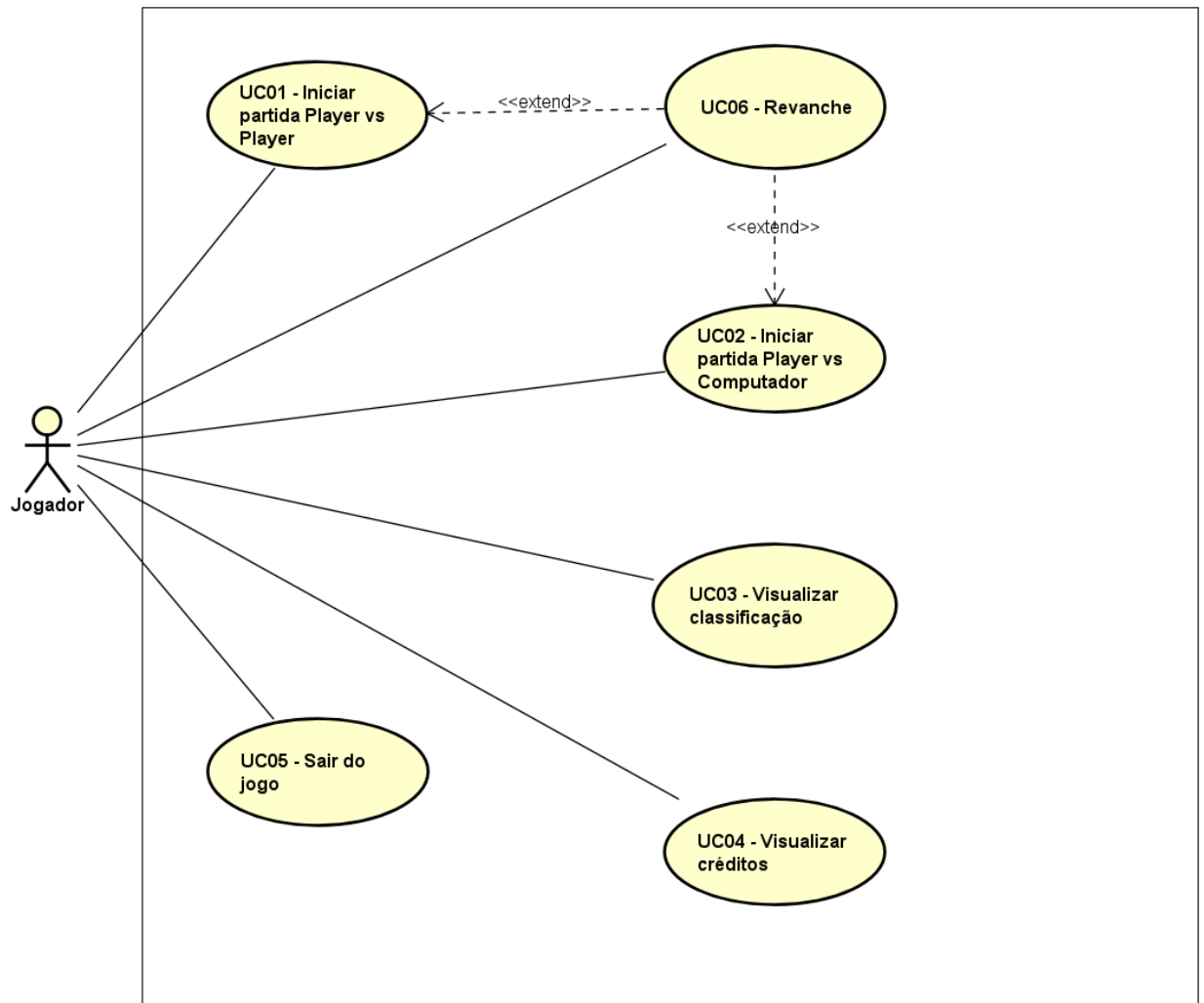


Figura 01 – Diagrama de Caso de Uso do software Jogo da Velha

## 5. ESPECIFICAÇÃO DE CASOS DE USO

Logo abaixo, estão a especificação dos casos de uso com o seus devidos fluxos (principal, alternativo e de exceção).

ID + Nome	UC01 - Iniciar partida PLAYER vs PLAYER
Descrição:	O requisito a seguir irá permitir aos jogadores realizarem todos os processos para começar uma nova partida contra outro jogador
Atores:	Jogadores
Requisitos:	Não há.
Pré-condições:	Não há.
Fluxo Principal:	<ol style="list-style-type: none"><li>1. O ator irá abrir o software.</li><li>2. O ator irá pressionar o botão "Player 1 vs Player 2".</li><li>3. O sistema irá pedir a matrícula do jogador 1.</li><li>4. O jogador 1 insere a matrícula e pressiona o botão "OK".</li><li>5. O sistema irá pedir a matrícula do jogador 2.</li><li>6. O jogador 2 insere a matrícula e pressiona o botão "OK".</li><li>7. O sistema inicia a partida.</li><li>8. Fluxo encerrado.</li></ol>
Fluxo Alternativo:	<p>FA01: Tela Inicial</p> <ol style="list-style-type: none"><li>1. O ator irá clicar no botão "Tela Inicial".</li><li>2. O sistema abrirá um pop-up com a pergunta "Deseja voltar à tela inicial?" e as opções "Yes" e "No".</li><li>3. O ator seleciona uma das opções (Caso a opção escolhida seja "Yes", o software irá direcioná-lo ao menu inicial do jogo; Caso a opção escolhida seja "No", irá manter os jogadores na tela de partida).</li><li>4. O sistema confirma a opção escolhida.</li><li>5. Fluxo Encerrado.</li></ol> <p>FA02: Reiniciar Jogo</p> <ol style="list-style-type: none"><li>1. O ator irá clicar no botão "Reiniciar Jogo".</li><li>2. O sistema abrirá um pop-up com a pergunta "Deseja reiniciar o jogo?" e as opções "Yes" e "No".</li><li>3. O ator seleciona uma das opções (Caso a opção escolhida seja "Yes", o software irá reiniciar a partida mantendo os dados de matrícula dos jogadores; Caso a opção escolhida seja "No", a partida continuará sem nenhuma alteração).</li><li>4. O sistema confirma a opção escolhida.</li><li>5. Fluxo Encerrado.</li></ol>

Fluxos de exceção:	<p>FE01: Os atores não informam suas matrículas e pressionam o botão "Ok".</p> <ol style="list-style-type: none"> <li>1. O sistema informará a seguinte mensagem: "Os campos das matrículas(s) não podem ser iguais ou vazios".</li> <li>2. O ator irá reiniciar o passo 4 do Fluxo Principal.</li> <li>3. Fluxo encerrado.</li> </ol> <p>FE02: Os atores informam de maneira incompleta/ incorreta suas matrículas e pressionam o botão "Ok"</p> <ol style="list-style-type: none"> <li>1. O sistema informará a seguinte mensagem: "Matrícula(s) não encontrada(s) na base de dados. Tente novamente".</li> <li>2. O ator irá reiniciar o passo 4 do Fluxo Principal.</li> <li>3. Fluxo encerrado.</li> </ol> <p>FE03: Os usuários informam a mesma matrícula em ambos os jogadores e pressionam o botão "Ok"</p> <ol style="list-style-type: none"> <li>1. O sistema informará a seguinte mensagem: "Os campos das matrículas(s) não podem ser iguais ou vazios".</li> <li>2. O ator irá reiniciar o passo 4 do Fluxo Principal.</li> <li>3. Fluxo encerrado.</li> </ol>
Pós condições:	Uma nova partida será iniciada.
Regra de negócio:	RD04

ID + Nome	UC02 - Iniciar partida PLAYER vs COMPUTADOR
Descrição:	O requisito a seguir irá permitir aos jogadores realizarem todos os processos para começar uma nova partida contra o computador.
Atores:	Jogadores
Requisitos:	Não há.
Pré-condições:	Não há.
Fluxo Principal:	<ol style="list-style-type: none"> <li>1. O ator irá abrir o software.</li> <li>2. O ator irá pressionar o botão "Player vs Computador".</li> <li>3. O sistema irá pedir a matrícula do jogador 1.</li> <li>4. O jogador 1 insere a matrícula e pressiona o botão "OK".</li> <li>5. O sistema inicia a partida.</li> <li>6. Fluxo encerrado.</li> </ol>
Fluxo Alternativo:	<p>FA01: Tela Inicial</p> <ol style="list-style-type: none"> <li>1. O ator irá clicar no botão "Tela Inicial".</li> </ol>

	<ol style="list-style-type: none"> <li>2. O sistema abrirá um pop-up com a pergunta “Deseja voltar à tela inicial?” e as opções “Yes” e “No”.</li> <li>3. O ator seleciona uma das opções (Caso a opção escolhida seja “Yes”, o software irá direcioná-lo ao menu inicial do jogo; Caso a opção escolhida seja “No”, irá manter os jogadores na tela de partida).</li> <li>4. O sistema confirma a opção escolhida.</li> <li>5. Fluxo Encerrado.</li> </ol> <p>FA02: Reiniciar Jogo</p> <ol style="list-style-type: none"> <li>1. O ator irá clicar no botão “Reiniciar Jogo”.</li> <li>2. O sistema abrirá um pop-up com a pergunta “Deseja reiniciar o jogo?” e as opções “Yes” e “No”.</li> <li>3. O ator seleciona uma das opções (Caso a opção escolhida seja “Yes”, o software irá reiniciar a partida mantendo os dados de matrícula dos jogadores; Caso a opção escolhida seja “No”, a partida continuará sem nenhuma alteração).</li> <li>4. O sistema confirma a opção escolhida.</li> <li>5. Fluxo Encerrado.</li> </ol>
Fluxos de exceção:	<p>FE01: Os atores não informam suas matrículas e pressionam o botão "Ok".</p> <ol style="list-style-type: none"> <li>1. O sistema informará a seguinte mensagem: "Os campos das matrículas(s) não podem ser iguais ou vazios”.</li> <li>2. O ator irá reiniciar o passo 4 do Fluxo Principal.</li> <li>3. Fluxo encerrado.</li> </ol> <p>FE02: Os atores informam de maneira incompleta/ incorreta suas matrículas e pressionam o botão “Ok”</p> <ol style="list-style-type: none"> <li>1. O sistema informará a seguinte mensagem: "Matrícula(s) não encontrada(s) na base de dados. Tente novamente”.</li> <li>2. O ator irá reiniciar o passo 4 do Fluxo Principal.</li> <li>3. Fluxo encerrado.</li> </ol> <p>FE03: Os usuários informam a mesma matrícula em ambos os jogadores e pressionam o botão “Ok”</p> <ol style="list-style-type: none"> <li>1. O sistema informará a seguinte mensagem: "Os campos das matrículas(s) não podem ser iguais ou vazios”.</li> <li>2. O ator irá reiniciar o passo 4 do Fluxo Principal.</li> <li>3. Fluxo encerrado.</li> </ol>
Pós condições:	Uma nova partida será iniciada.
Regra de negócio:	RD04

ID + Nome	UC03 - Visualizar Classificação
Descrição:	O requisito a seguir irá permitir aos jogadores a visualização da classificação atual contendo os dez melhores jogadores e também sua posição atual de vitórias.
Atores:	Jogadores.
Requisitos:	Não há.
Pré-condições:	Não há.
Fluxo Principal:	<ol style="list-style-type: none"> <li>1. O ator irá abrir o software.</li> <li>2. O ator irá pressionar o botão "Classificação".</li> <li>3. O sistema irá apresentar a posição dos 10 melhores jogadores e a pontuação referente a matrícula inserida anteriormente.</li> <li>4. Fluxo Encerrado.</li> </ol>
Fluxo Alternativo:	<p>FA01: Buscar usuário</p> <ol style="list-style-type: none"> <li>1. O ator irá inserir a matrícula que deseja consultar a classificação e irá apertar o botão "Buscar"</li> <li>2. O software irá apresentar a colocação do usuário que foi buscado.</li> <li>3. Fluxo Encerrado.</li> </ol> <p>FA02: Tela inicial</p> <ol style="list-style-type: none"> <li>1. O ator irá pressionar o botão "Tela inicial" com o propósito de voltar ao menu principal do software.</li> <li>2. O software irá processar a solicitação e o encaminha à página desejada.</li> <li>3. Fluxo Encerrado.</li> </ol>
Fluxos de exceção:	<p>FE01: O ator não informa a matrícula e pressiona o botão "Buscar".</p> <ol style="list-style-type: none"> <li>1. O sistema informará a seguinte mensagem: "Matrícula não encontrada, tente outra".</li> <li>2. O ator irá reiniciar o passo 1 do FA01..</li> <li>3. Fluxo encerrado.</li> </ol> <p>FE02: O ator informa de maneira incompleta/ incorreta sua matrícula.</p> <ol style="list-style-type: none"> <li>1. O sistema informará a seguinte mensagem: "Matrícula não encontrada, tente outra".</li> <li>2. O ator irá reiniciar o passo 1 do FA01..</li> <li>3. Fluxo encerrado.</li> </ol>

Pós condições:	Será apresentado o ranking dos melhores jogadores.
Regra de negócio:	RD03.

ID + Nome	UC04 - Visualizar Créditos
Descrição:	O requisito a seguir irá permitir que o ator acesse os nomes dos criadores do jogo.
Atores:	Jogadores.
Requisitos:	Não há.
Pré-condições:	Não há
Fluxo Principal:	<ol style="list-style-type: none"> <li>1. O ator irá abrir o software;</li> <li>2. O ator irá pressionar o botão "Créditos";</li> <li>3. O sistema irá exibir os nomes dos criadores do software;</li> <li>4. Fluxo encerrado.</li> </ol>
Fluxo Alternativo:	FA01: Tela inicial <ol style="list-style-type: none"> <li>1. O ator irá pressionar o botão “Tela inicial” com o propósito de voltar ao menu principal do software.</li> <li>2. O software irá processar a solicitação e o encaminha à página desejada.</li> <li>3. Fluxo Encerrado.</li> </ol>
Fluxos de exceção:	Não há.
Pós condições:	Os nomes dos criadores do software são exibidos.
Regra de negócio:	Não há.

ID + Nome	UC05 - Sair do jogo
Descrição:	O requisito a seguir irá permitir que o ator encerre a aplicação e feche o jogo.
Atores:	Jogadores.
Requisitos:	Não há.
Pré-condições:	Não há
Fluxo Principal:	<ol style="list-style-type: none"> <li>1. O ator irá abrir o software;</li> <li>2. O ator irá pressionar o botão "Sair do jogo";</li> <li>3. As atividades da aplicação são encerradas e o software é</li> </ol>

	<p>fechado;</p> <p>4. Fluxo encerrado.</p>
Fluxo Alternativo:	Não há
Fluxos de exceção:	Não há.
Pós condições:	O processo é encerrado e o jogo fecha.
Regra de negócio:	Nao há

ID + Nome	UC06 - Revanche
Descrição:	O requisito a seguir irá permitir aos atores recomeçarem a partida mantendo os dados de matrículas dos jogadores, desta forma, impedindo a inserção excessiva de suas matrículas
Atores:	Jogadores
Requisitos:	RF01
Pré-condições:	O UC01 necessita estar finalizado.
Fluxo Principal:	<ol style="list-style-type: none"> <li>1. O software irá apresentar aos jogadores a possibilidade de recomeçar a partida mantendo os dados de matrícula já inseridos, através da mensagem “O jogador x venceu o jogo!! Deseja fazer revanche?”</li> <li>2. O jogador irá selecionar a ação de acordo com sua necessidade (Caso pressione o botão “Yes” terá uma revanche com o adversário; Caso pressione “No”, será encaminhado ao menu principal).</li> <li>3. O software irá validar a opção selecionada.</li> <li>4. Fluxo encerrado.</li> </ol>
Fluxo Alternativo:	Não há.
Fluxos de exceção:	Não há.
Pós condições:	As falhas ou bugs serão reportadas.
Regra de negócio:	Não há.



## 6. DIAGRAMA DE CLASSE

Adiante, o diagrama de classe feito a partir dos casos de uso anteriormente.

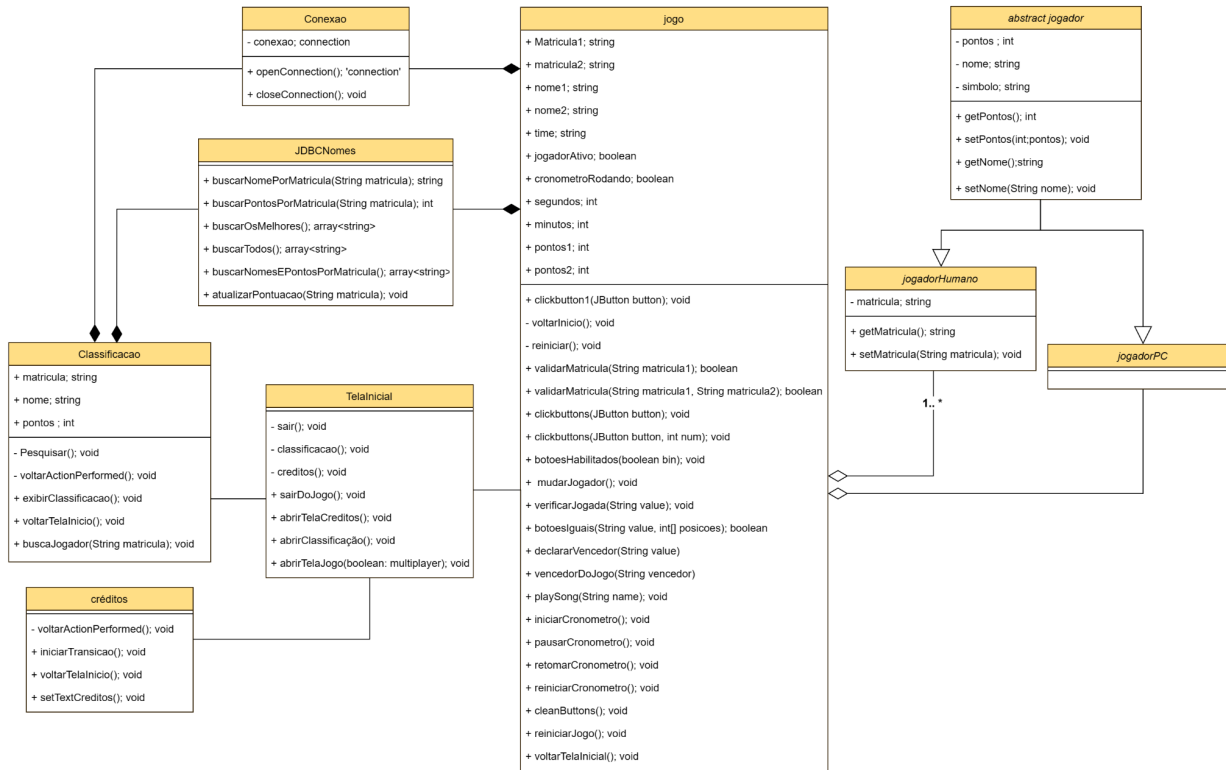


Figura 02 – Diagrama de Classe do software Jogo da Velha (versão atual)

## 7. INTERFACES

A seguir serão apresentadas as interfaces do software que tem o propósito documentar como a ferramenta irá se comportar e quais serão os passos para os procedimentos principais.



Figura 03 – Interface da tela inicial do jogo



Figura 04 – Interface da tela de uma partida do jogo



Figura 05 – Interface da tela de classificação do jogo

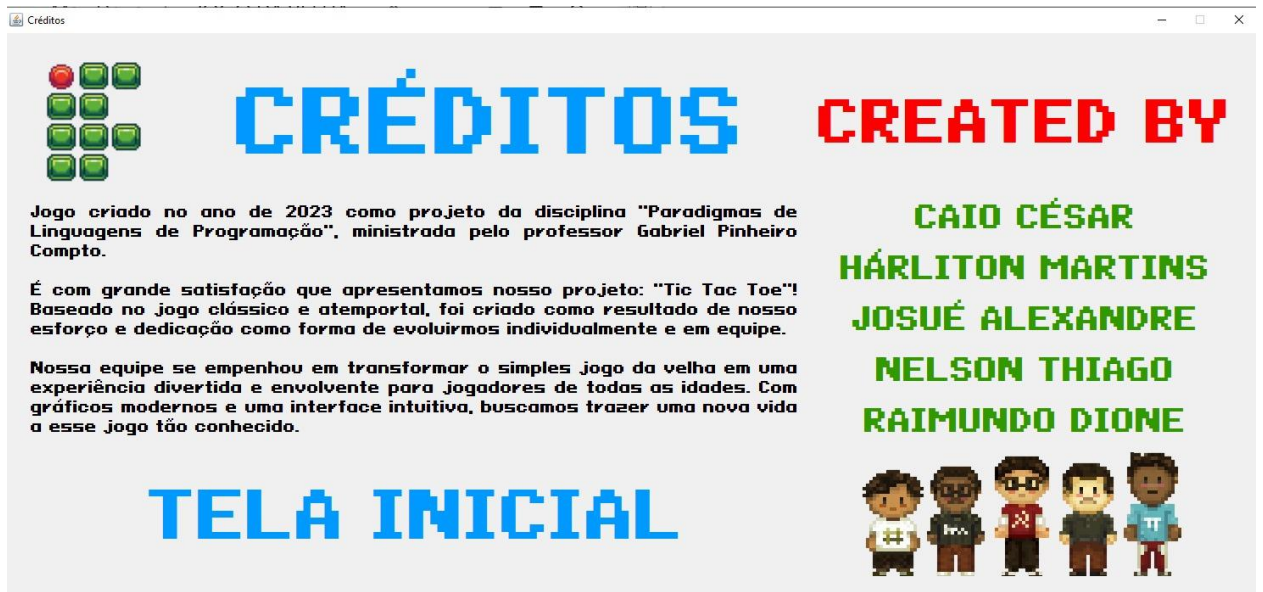


Figura 06 – Interface da tela de créditos do jogo

## 8. REFERÊNCIAS

QUEIROZ, Danilo Nascimento; OLIVEIRA, Dayvid Victor Rodrigues; ALMEIDA, Henrique Alexandre M. S.; SALES, Tiago Bezerra do Amaral. Especificação dos Requisitos do Software BrightChef. Recife, 2009. Disponível em: <[https://drive.google.com/drive/u/0/folders/1TcEhLzud-8f\\_JAAq6HenAHr0gWHL6wKT](https://drive.google.com/drive/u/0/folders/1TcEhLzud-8f_JAAq6HenAHr0gWHL6wKT)>.

Acesso em: 20 jun. 2023.

Sommerville, I. (2019). *Engenharia de software. 10ª edição*. Pearson Universidades, ISBN: 978-8543024974.

Pressman, R.S.; Maxim, B.R. (2021). *Engenharia de software. 9ª edição*. AMGH. ISBN: 978-6558040101.