Adailson Almeida1; André Vinicius Vieira Mildemberg1; Luiz Carlos Rosa Junior²

*1 Graduando do curso de Análise e desenvolvimento de sistemas do Centro Universitário Unifacear;*

*2 Docente do curso de Análise e desenvolvimento de sistemas do Centro Universitário Unifacear*

**RESUMO**

*Realizamos o desenvolvimento de um jogo de damas em linguagem Java, para realizar o planejamento do desenvolvimento do sistema utilizamos diagramas UML, como o diagrama de casos de uso e diagrama de classes.*

*O sistema consiste em um jogo de damas com as regras brasileiras do jogo, como por exemplo: A partida é realizada entre dois jogadores, em tabuleiro 8X8 (64 casas) alternadamente brancas e pretas, com 12 peças brancas e 12 peças pretas.*

*Para gerar uma maior competitividade nos jogadores o sistema conterá um ranqueamento entre todos os jogadores com base em pontos que eles ganharão nas vitorias de cada jogo.*

*Palavras chave: damas, jogador, Java, diagramas.*

***ABSTRACT***

*We developed the game of checkers in Java language, to carry out the planning of the development of the system uses UML diagrams, such as the use-case diagram and class diagram.*

*The system consists of a game of checkers with the classic rules of the game, such as: The game is played between two players, on an 8X8 (64) board alternately white and black, with 12 white pieces and 12 black pieces.*

*To generate a greater characteristic in the players the system will contain a ranking among all the players based on points that they will win in the victories of each game.*

*Keywords: checkers, player, Java, diagrams.*

1. **INTRODUÇÃO**

Para este projeto nós desenvolvemos um jogo de damas em linguagem Java com base nos princípios de orientação a objetos (POO). Para nos auxiliar no desenvolvimento nos utilizamos a IDE (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado) Eclipse.

Além disso, no processo de desenvolvimento nós utilizamos diagramas baseados na UML para definirmos de forma mais concreta o que o sistema deverá ou não conter, para desta forma sermos mais produtivos no desenvolvimento. Os diagramas utilizados foram o diagrama de classes e o diagrama de casos de uso.

1. **DESENVOLVIMENTO**

2.1 REGRAS

De acordo com o site “BrainKing” existem diversas versões variantes no jogo de damas entre elas as damas internacionais, Canadianas, alquerques, checas e brasileiras, que inevitavelmente é a versão mais popular no país. O nosso jogo será baseado na versão brasileira. As regras do jogo de damas brasileiro são:

* Tabuleiro 8X8
* Partida realizada entre dois jogadores
* 24 ao todo
* 12 pretas e 12 brancas
* Peças brancas começam
* A peça só anda para frente
* Anda uma casa por vez
* Quando atinge a última linha a peça vira dama
* A dama anda para frente e para trás sem limites de casas
* A dama não “passar por cima” de nenhuma peça aliada
* Se houver a possibilidade a captura é obrigatória
* A peça só pode ser capturada se houver um espaço livre na mesma diagonal que a peça que irá capturar está
* As peças normais podem realizar a captura indo para trás
* Se no mesmo lance se apresentar mais de um modo de capturar, é obrigatório executar o lance que capture o maior número de peças (Lei da Maioria).
* Após 20 lances sucessivos, sem captura ou deslocamento de pedra, a partida é declarada empatada.

2.2 FERRAMENTAS UTILIZADAS NO DESENVOLVIMENTO

WIP - COMING SOON

2.3 DIAGRAMA DE CASOS DE USO

Um diagrama de casos de uso é um diagrama dinâmico ou de comportamento na UML (Linguagem de Modelagem Unificada). O cientista de computação JACOBSON descreve casos de uso da seguinte forma, podemos dizer que um caso de uso é um documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo.

Com base nesta descrição desenvolvemos o diagrama de casos de uso do projeto tentando incluir todas os casos de uso do projeto.

A seguir na figura 1, o Diagrama de casos de uso do projeto:

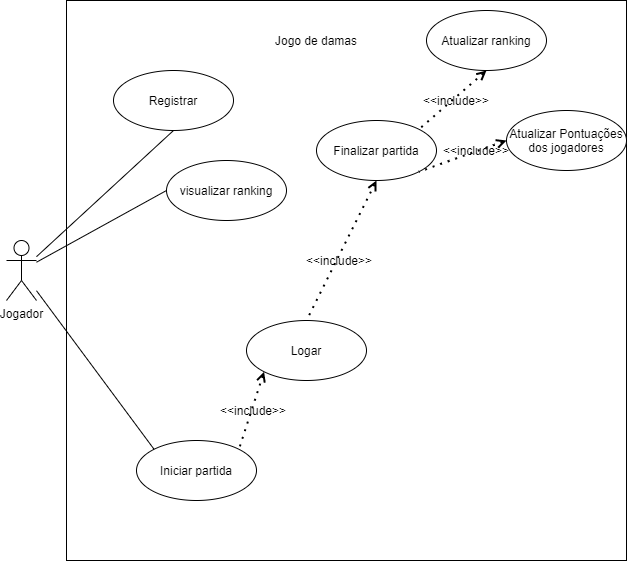


FIGURA 1: Diagrama de casos de uso do projeto.

FONTE: Os autores (2020)

2.4 Diagrama de classes

Segundo Donald Bell no livro Fundamentos básicos de UML: O propósito do diagrama de classes é mostrar os tipos que estão sendo modelados no sistema e a relações entre eles. Nessa perspectiva observa-se que é fulcral a elaboração de um diagrama de classes para um sistema afim de documentar corretamente o funcionamento do sistema e facilitar futuras manutenções no código.

Para a melhor visualização do diagrama de classes neste artigo nos dividimos ele em quatro partes que estão representadas nas figuras 2, 3, 4, 5. Após isso dos itens 2.4.1 até o 2.4.X temos uma breve descrição das principais classes do sistema.

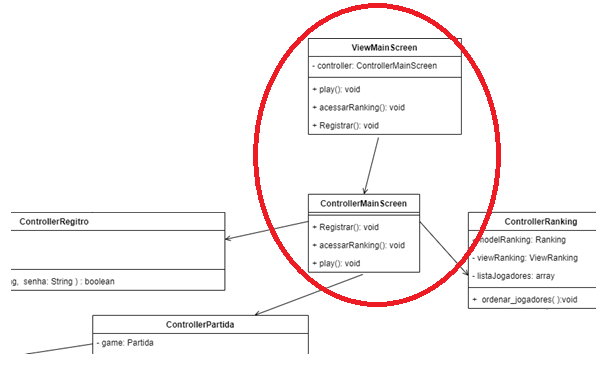


FIGURA 2: Parte central do diagrama.

FONTE: Os autores (2020)

Na figura 2 podemos destacar as classes view e controller da tela principal, a partir dela podemos dividir o sistema em três ramificações que são as classes de registro, de ranking e a parte principal do sistema que são as classes da partida.

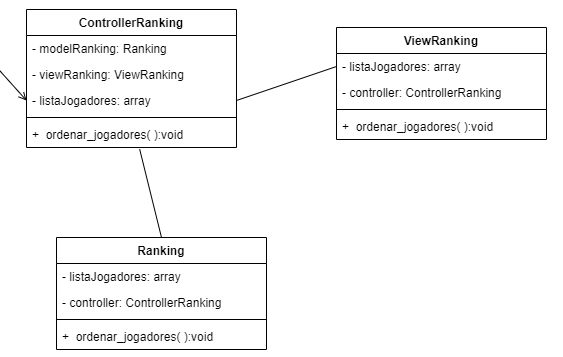


FIGURA 3: Classes de Ranking.

FONTE: Os autores (2020)

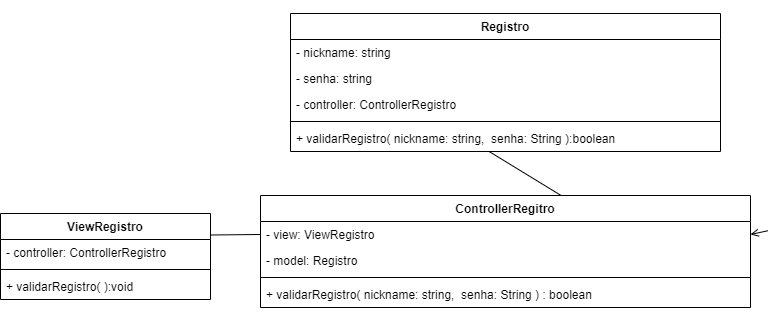


FIGURA 4: Classes de registro.

FONTE: Os autores (2020)

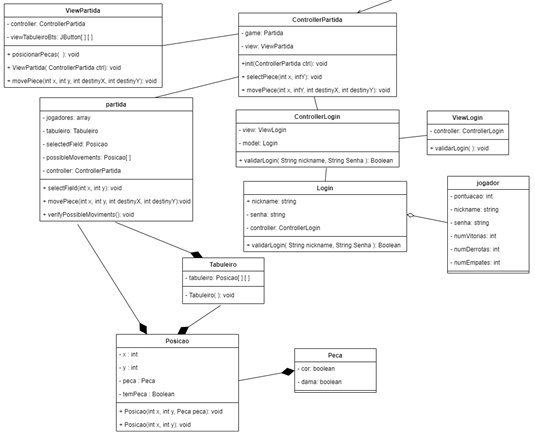


FIGURA 5: Classes da partida.

FONTE: Os autores (2020)

2.4.1 Classes ViewMainscreen e ControllerMainScreen

Como nós desenvolvemos nosso sistema baseado nos padrões de projeto MVC, para funcionar como núcleo do sistema temos as classes View e controller da tela principal, representadas na figura 2, elas permitem ao usuário acessar as outras ramificações do sistema.

2.4.2 Classes de Ranking

Nessa Ramificação do sistema estão as classes model, view e controller do ranking, que estão representadas na figura 3, essas classes serão responsáveis por extrair da base de dados do sistema as informações sobre as pontuações de cada jogador e exibir ao usuário ordenando do jogador com maior pontuação para o com menor.

2.4.3 Classes de registro

As classes de registro assim como as classes de ranking estão separadas nas classes model, view e controller. Essas classes são responsáveis por validar e cadastrar os dados do jogador na base de dados, para ele poder realizar login posteriormente, para se registrar o jogador precisa informar um nickname e sua senha.

2.4.3 Classes da partida

Por essa ser a ramificação mais complexa e ao mesmo tempo a mais importante do sistema dividimos a explicação dela em alguns sub tópicos.

2.4.3.1 Classe ControllerPartida

Essa classe é instanciada no momento que o usuário entra na opção jogar, então ela instancia as classes de login, para os dois jogadores realizarem login no sistema, então só após os jogadores logarem no sistema ela instancia as classes model e view da partida passando como parâmetro para classe model dois objetos Jogador, que são retornados pela classe de login, contendo as informações sobre os jogadores.

2.4.3.2 Classes de login

As classes de login também dividas no padrão MVC recebem as informações de nickname e senha do usuário e fazem a validação se o a senha corresponde a esse login e se estão devidamente cadastrados no sistema, após realizada a validação a classe model login instancia um objeto jogador com as informações do jogador que realizou login e retorna para a classe Controller partida para atualizar as estatísticas do jogador após o fim da partida.

2.4.3.3 Classe Partida

A classe model da partida contém todas as regras de negocio do jogo, é ela que realiza a validação dos movimentos da partida, organiza a vez de cada jogador jogar, determina o fim da partida, atualiza as estatísticas de cada jogador e salva na base de dados.

2.4.3.4 Classe Tabuleiro, Posicao e Peca

A classe tabuleiro é instanciada pela classe Partida, ela contem uma matriz de objetos da classe Posicao. Essa classe é utilizada para realizar as validações da classe Partida.

A classe Posicao possui os atributos de coordenadas X e Y, um atributo peca que é um objeto da classe Peca e um atributo booleano temPeca para verificar se nessa posição tem uma peça.

A classe Peca tem dois atributos o primeiro contendo um valor de 0 ou 1 para representar a cor dessa peça e um atributo booleano para verificar se a peça é uma dama.

1. **CONCLUSÃO**

**4. REFERÊNCIAS**

### BELL, DONALD. Fundamentos básicos de UML: O diagrama de classes. 19, dezembro de 2016. IBM Corporation.

### 

### JACOBSON, IVAN: [Object Oriented Software Engineering: A Use Case Driven Approach](https://www.amazon.com.br/Object-Oriented-Software-Engineering-Approach/dp/0201544350/ref=sr_1_3?qid=1585572037&refinements=p_27%3AIvar+Jacobson&s=books&sr=1-3). 1 ed.Editora: Addison Wesley. 1, julho de 1992.

### 

### Regras jogo de damas. 11, maio de 208. Disponível em: https://brainking.com/pt/GameRules?tp=120 >. Acesso em: 11, de setembro de 2020.

### 