UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

IURI GRANVILLE PELISSON MARDEGAN

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2014

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

UMA PLATAFORMA WEB PARA COMPETIÇÃO AUTOMÁTICA DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL EM JOGOS DE TABULEIRO

Orientador: Prof. Dr. Carlos N. Silla Jr.

CORNÉLIO PROCÓPIO
2014

Sumário

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES

BD	Banco de Dados
HTML	HyperText Markup Language (Linguagem de marcação de hipertexto)
I.A.	Inteligência Artificial
WWW	World Wide Web
UTFPR-CP	Universidade Tecnológica Federal do Paraná – Câmpus Cornélio Procópio
CPW	Chess programing wiki (Enciclopédia de programação para Xadrez)

1 INTRODUÇÃO

Com a evolução da sociedade ao longo dos anos surge a necessidade de aprimorar as atividades do dia-a-dia de modo a minimizar custos. Essa necessidade faz com que processos anteriormente realizados de forma manual sejam automatizados. Dessa forma, surge a inteligência computacional que tem com objetivo desenvolver sistemas/máquinas que possam automatizar as atividades desenvolvidas por humanos.

Uma das aplicações de sistemas que utilizam técnicas de inteligência computacional é a capacidade de receber informações, analisar essas informações e emitir uma resposta baseada nos dados informados. Para Kurzweil (1990) a Inteligência Computacional é “a arte de criar maquinas que executam funções que exigem inteligência quando executadas por pessoas”.

A área de pesquisa em Inteligência artificial (I.A.) é muito ampla e aborda diversos tópicos que vão desde a prova de teoremas a sistemas de visão computacional. Uma área de grande interesse dentro da área de I.A. é o seu uso para jogos.

Nesse contexto, plataformas online e desktop em que um jogador humano pode jogar contra a I.A. criada para um jogo específico não são difíceis de serem encontrados hoje em dia. Uma busca no www.google.com por “jogos de tabuleiro online” ou “jogos online” nos retorna diversas opções de jogos com I.A. para jogar contra jogadores humanos.

Contudo, existem poucos sistemas que permitam que diferentes algoritmos de I.A., para jogos de tabuleiros, possam competir uns contra os outros. Um exemplo dos poucos sistemas existentes é o Chess programing wiki (CPW).

O CPW é um repositório que pode ser acessado por qualquer pessoa e é específico para o jogo de xadrez. Para ter acesso é necessário realizar um cadastro no site. No CPW usuário pode ter acesso a códigos em varias linguagens de programação como Java, Delphi, Ruby, C, dentre outras.

Os usuários podem contribuir com diferentes atividades no CPW. Dentre elas estão a busca de falhas, testes de posições e aberturas de jogos. Por exemplo, na abertura de jogos eles recomendam analisar a técnica de “Livro Aberto”, esta técnica

trata de recuperar movimentos de abertura bem sucedidos que foram armazenados anteriormente, para diminuir o tempo gasto durante a analise de posições.

O CPW sugere aos seus usuários plataformas na internet onde se pode comparar o tempo e o desempenho do algoritmo. Neste mesmo ambiente também podemos encontrar técnicas para avaliar as estatísticas do jogo, de modo a indicar se tais técnicas implicam em melhora nos resultados dos algoritmos ou não, com o objetivo de armazenar apenas as melhores jogadas (ARENA, 2014).

Uma limitação do CPW é o fato dele não realizar torneios entre os usuários. Por esse motivo, uma plataforma que permita a organização de torneios entre as I.A. é de grande importância para o ensino de técnicas de I.A. em diversos cursos de graduação.

No contexto da UTFPR-CP, o Prof. Carlos N. Silla Jr. que orienta este trabalho, pede que os alunos da sua disciplina de I.A. implementem a I.A. de um jogo de tabuleiro como sendo um dos trabalhos da disciplina. Um fator motivacional para engajar os alunos na disciplina é além deles criarem a sua própria I.A., eles poderem participar de um campeonato onde a I.A. desenvolvida por cada equipe jogue contra a I.A. desenvolvida pelas demais equipes. Porém como já mencionado, desconhecemos a existência de qualquer trabalho que permita que diferentes técnicas de I.A. possam competir entre si.

Dessa forma o objetivo deste trabalho foi construir uma plataforma web para a competição de técnicas de I.A. em jogos de tabuleiro. Em função do tempo de desenvolvimento deste projeto a plataforma foi desenvolvida de forma genérica utilizando os conceitos de orientação a objetos e foram implementos dois jogos de tabuleiro para testar seu funcionamento e extensibilidade. Os jogos implementados foram o jogo da velha (*tic tac toe*) e o othello (*Reversi*).

O jogo da velha tem uma regra simples onde para ganhar o jogador tem que atingir o objetivo de alinhar três de suas marcas (normalmente “X” e “O”) na horizontal, vertical ou diagonal, em cada jogada pode-se adicionar uma marca em uma casa livre, o tabuleiro consiste em nove casas livres, na Figura - 1 temos um exemplo de jogo da velha em andamento.

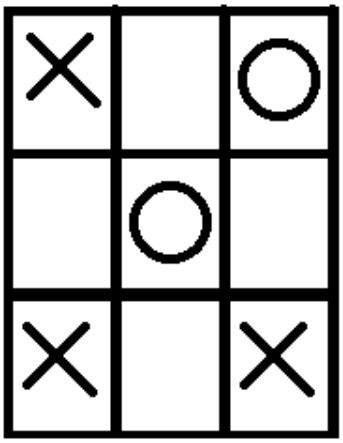


Figura 1 - Tabuleiro Jogo da Velha – figura ilustrando tabuleiro de Jogo da velha.
FONTE: Adaptado de (Velha, 2013).

O Othello inicia com um tabuleiro 8x8 com quatro peças sendo duas brancas e duas pretas, conforme ilustrado na Figura 2, as peças pretas sempre fazem o primeiro movimento e os jogadores se intercalam a cada jogada, o objetivo do jogo é converter as peças adversarias para sua cor, o jogo acaba quando não existir mais casa no tabuleiro e quem possuir mais peças vence, a cada jogada realizada é inserida uma nova peca no tabuleiro, ao inserir esta peça todas as outras que estiverem entre uma peça anteriormente inserida e a nova peça será convertida para a sua cor, seja na horizontal, vertical ou diagonal, a nova peça pode ser inserida em casas ao redor das peças já existentes que permitam criar uma linha diagonal, horizontal ou vertical.

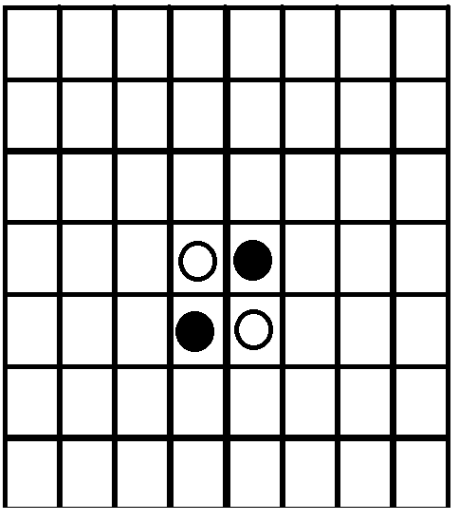


Figura 2 - Tabuleiro Othello – imagem ilustrando estado inicial do tabuleiro de Othello.
FONTE: Adaptado de (Othello, 2013).

É importante ressaltar que, como a plataforma foi desenvolvida segundo os princípios de orientação a objetos, ela permite que novos jogos de tabuleiro possam ser adicionados em trabalhos futuros.

1.1 INTELIGÊNCIA ARTIFICIAL EM JOGOS DE TABULEIRO

A teoria dos jogos visualiza um ambiente de múltiplos agentes onde o impacto de um agente sobre o outro seja significativo, os jogos são uma das primeiras áreas de desenvolvimento de I.A.

Em 1950 quando os computadores se tornaram programáveis foi criado o primeiro jogo de xadrez por Claude Shannon e por Alan Turing, assim abrindo uma nova área na computação que começaria a se desenvolver. Em 1996 foi realizado o primeiro confronto entre um dos melhores jogadores de xadrez do mundo, Garry Kasparov e Deep Blue (SETTI, 2013).

O Deep Blue é um computador de alta performance da IBM programado em C e executado em um sistema operacional AIX, ele é capaz de calcular cerca de 200 bilhões de jogadas em até três minutos, no ano de 1997 o Deep Blue derrotou o então campeão mundial de xadrez, Garry Kasparov (SETTI, 2013).

Outra I.A. para jogos conhecida é o Chinook, desenvolvida pelo canadense Jonathan Schaeffer em 1989 para jogar damas, é capaz de utilizar 39 trilhoes de combinações, em 2007 ela atingiu tal nível que se tornou invencível ou seja se o melhor jogador de damas do mundo realizasse um jogo perfeito contra o Chinook ele iria no máximo empatar(BOLSONI, 2007).

A principal técnica de desenvolvimento de I.A. para jogos de tabuleiro é conhecida como MinMax que busca realizar uma analise em que o software simula o jogo até o final a partir da jogada de seu adversário para assim identificar qual situação ele teria maiores chances de vencer e minimizar as jogadas de seu adversário (RUSSEL, 1995). Aliada ao uso do MinMax outra técnica que pode ser utilizada é o mecanismo de poda Alpha-Beta que tem por objetivo minimizar o número de estados avaliados pelo MinMax. A intuição por trás da poda Alfa-Beta é que se uma jogada é ruim o sistema não precisa estender até o final para ver o quanto ela é ruim, basta analisar a melhor jogada.

2 JUSTIFICATIVA

Atualmente nas aulas de I.A., sem a existência da plataforma desenvolvida neste trabalho possui duas limitações: A primeira limitação é que cada aluno deve desenvolver um projeto completo com implementando as regras do jogo e também a interface gráfica. Ou seja, os alunos perdem tempo desenvolvendo a interface e a regras de um determinado jogo de tabuleiro quando poderiam estar focados no desenvolvimento do módulo de I.A. deste jogo. A segunda limitação é que dependendo do número de alunos que fizerem a disciplina não é viável fazer um campeonato onde a I.A. do sistema desenvolvido por cada aluno jogue contra todas as I.A. desenvolvidas pelos outros alunos. Isso é uma característica importante para a avaliação adequada dos projetos, visto que se forem geradas chaves aleatórias, pode acontecer do segundo melhor trabalho ser desclassificado se, por azar, jogar a primeira partida contra o melhor trabalho desenvolvido.

Para solucionar estes problemas foi criada uma plataforma web para a competição automática de técnicas de I.A. em Jogos de tabuleiro, onde os alunos criam suas I.A. em cima das funcionalidades da plataforma. A plataforma fornece uma interface padrão para os jogos implementados e permite que os alunos enviem seus sistemas para a plataforma de forma que o campeonato seja automatizado.

O Framework implementado tem por objetivo validar as jogadas dos alunos aplicando as regras dos jogos e identificando o vencedor ou uma situação de empate. O Framework também é responsável por identificar se a jogada dos alunos são validas, e caso não sejam o aluno ira perder o confronto automaticamente. Além disso, o framework é responsável por auxiliar os alunos na implementação de suas I.A., visto que ele fornece os estados do jogo mas não implementa os métodos de busca competitiva como o algoritmo de MinMax, nem as heurísticas para avaliação dos estados. A implementação dessas funcionalidades é parte das atividades dos alunos.

Na ocasião da redação deste trabalho, os alunos da disciplina de I.A., do 7º período do curso de Engenharia da Computação da UTFPR-CP utilizaram uma versão preliminar da plataforma.

3 VISÃO DA SOLUÇÃO

Módulo Web: Responsável por gerenciar os usuários e campeonatos, também deve ter informações instrutivas para os usuários de como devem agir para implementar e enviar I.A. desenvolvida por ele. Os usuários cadastrados podem visualizar os campeonatos iniciados por um administrador, e caso desejem eles podem realizar o cadastro para este campeonato. Para que um usuário possa se cadastrar no campeonato é necessário que o mesmo envie o código em Java da sua I.A. O usuário administrador é responsável por encerrar as inscrições. Quando as inscrições são encerradas é gerada a tabela de confrontos, na modalidade todos contra todos. Após os confrontos terem sido gerados, o administrador deve clicar no botão “preparar de confronto” para realizar o confronto entre duas I.A. Quando todos os confrontos forem realizados o campeonato será finalizado e será feita a publicação dos resultados.

Framework: Este módulo é responsável por realizar cada confronto. Um administrador prepara o confronto no modulo web que ira movimentar as classes enviadas para seus respectivos pacotes (player1 ou player2) e atualizar as referencias de tipo de jogo e código do confronto dentro da classe torneio. Atualmente estão implementadas os jogos da velha e Othelo. As informações utilizadas pelo framework nos confrontos são compartilhadas com o módulo web através do BD. Dessa forma, após cada confronto ser realizado, é possível verificar o jogo passo a passa através do módulo web.

3.1 DIAGRAMA DE INTEGRAÇÃO

A figura 3 representa o diagrama de integração da base de dados com o módulo WEB que ira interagir com o framework, atualizando as classes necessárias para que ocorra os confrontos.

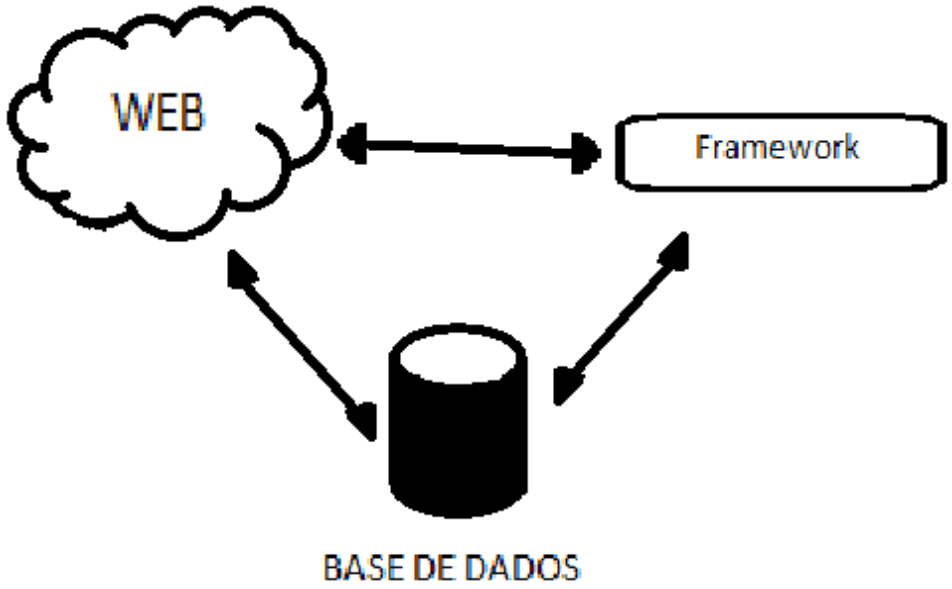


Figura 3 - Diagrama de integração.

4 ESCOPO DA SOLUÇÃO

O sistema foi modelado e implantado de modo a realizar consultas e gerar relatórios sobre os dados gerados durante os confrontos, qualquer usuário cadastrado no sistema tem acesso a um arquivo Interface java para desenvolver sua I.A. e participar do Campeonato.

4.1 OBJETIVO GERAL

O objetivo geral deste projeto foi desenvolver uma plataforma web para a competição automática (Campeonatos) de sistemas de I.A. para jogos de tabuleiro.

4.2 OBJETIVOS ESPECÍFICOS

- Desenvolver um módulo de gerenciamento de usuários.
- Desenvolver um módulo de gerenciamento dos campeonatos. Este módulo permite campeonatos todos contra todos.
- Desenvolver de uma plataforma para que os usuários se cadastrem no campeonato, fazendo o envio do jogador
- Os usuários terem acesso ao resultado dos seus confrontos.
- Implementar todas as funcionalidades do jogo da velha na plataforma.
- Implementar todas as funcionalidades do jogo othello na plataforma.
- Implementar dois tipos de players na plataforma. Um player “dummy” que faz jogadas aleatórias.
- Desenvolver de um breve tutorial sobre como usar os recursos da plataforma para que novos jogos possam ser implementados por outros desenvolvedores.

5 FUNCIONALIDADES, DIAGRAMAS E REQUISITOS

- **Essencial**: Foi desenvolvida de forma completa, inicial e prioritária;
- **Importante:** Foi desenvolvida de forma completa, porém da-se prioridade anteriormente às funcionalidades de precedência Essencial.

Tabela 1 - Descrição de Funcionalidades

Id	Módulo	Nome	Prioridade
F1	WEB	Cadastro de usuários	Essencial
F2	WEB	Liberação de cadastros	Importante
F3	WEB	Gerencia de campeonato	Essencial
F4	WEB	Usuário em campeonato	Essencial
F5	WEB, Framework	Confronto de usuários	Essencial
F6	Framework	Analise de resultados	Essencial

Cadastro de usuários: permite aos usuários solicitarem a inscrição pela web para poder participar da comunidade ativa que participa de campeonatos de I.A.

- **Liberação de cadastros por usuário administrador:** permite ao administrador aprovar ou reprovar as solicitações de cadastro
- **Gerencia de campeonato:** permite ao administrador iniciar e finalizar campeonatos.
- **Usuário em campeonato:** permite ao usuário com cadastro aprovado, se inscrever em campeonatos abertos com o envio de um arquivo com sua I.A..
- **Confronto de usuários:** permite ao administrador realizar os confrontos através da ação “preparar”, após realizar tal ação ele deve compilar e executar o framework que estará preparado para o confronto.
- **Analise de resultados:** permite ao usuário analisar os resultados de cada confronto, campeonato ou usuário.

5.1 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso de alto nível é apresentado na Figura 4.

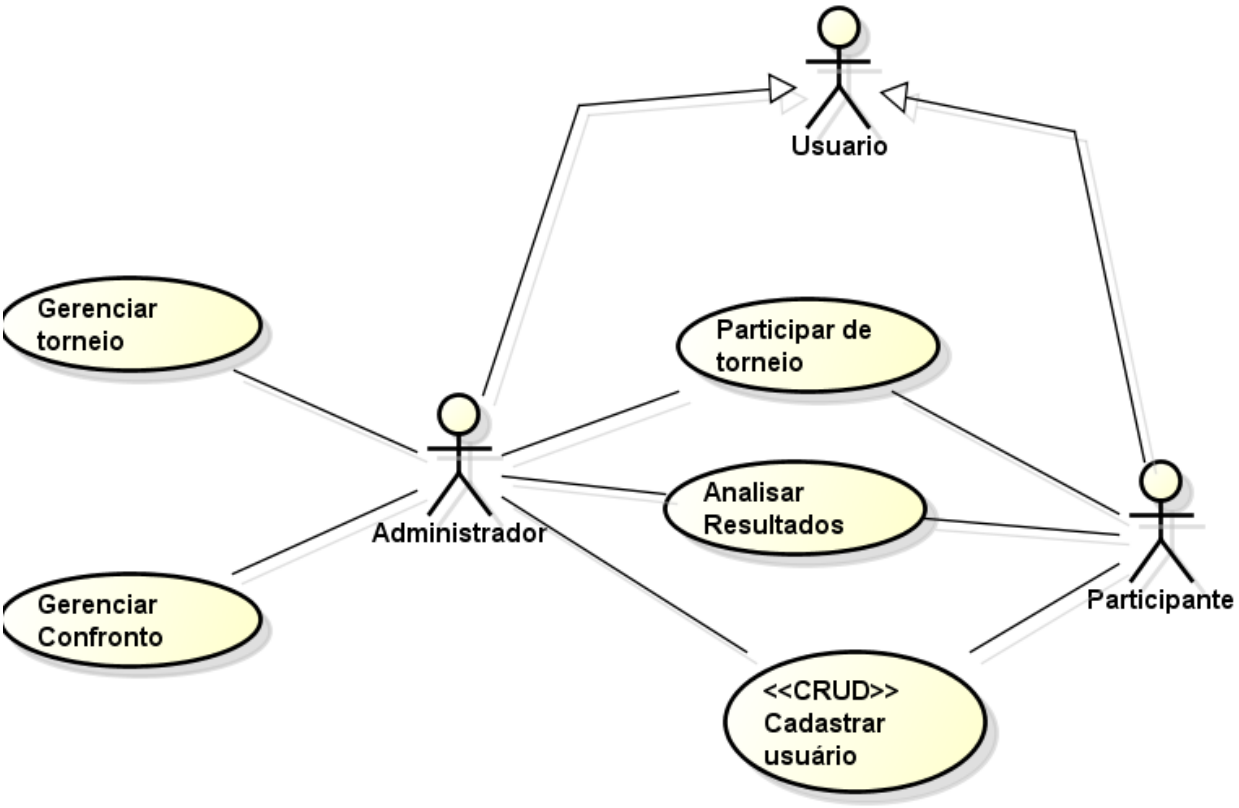


Figura 4 – Diagrama de caso de uso

5.2 REQUISITOS

Requisitos funcionais são apresentados na Tabela 2.

Tabela 2 - Requisitos funcionais

Id	Módulo	Nome	Requisitos não funcionais
R1	WEB	Permitir login dos usuários.	RN1
R2	WEB	Permitir cadastro e edição de usuários.	RN2, RN3, RN4, RN5
R3	WEB	Permitir cadastro de campeonatos.	RN5, RN6, RN7, RN8
R4	WEB	Permitir cadastro de módulo de jogo.	RN5
R5	WEB	Permitir inscrição em campeonato.	RN5, RN11
R6	WEB	Permitir gerar confrontos.	RN9
R7	WEB	Permitir visualização de replay.	RN10
R8	FRAMEWORK	Realizar persistência de jogada.	RN12, RN13, RN14, RN15
R9	FRAMEWORK	Realizar jogo sem persistência. (Para possibilitar testes no desenvolvimento)	RN16
R10	FRAMEWORK	Classe principal de utilizar métodos apenas das classes abstratas em questão de Jogo e Jogador.	

A tabela 3 apresenta os requisitos não funcionais.

Tabela 3 - Requisitos não funcionais.

Id	Módulo	Nome
RN1	WEB	Permitir login apenas de usuários cadastrados.
RN2	WEB	Permitir cadastro de usuário apenas por administradores.
RN3	WEB	Permitir que usuário não administrador possa editar seus dados.
RN4	WEB	Permitir que campo de usuário denominado administrador seja visualizado apenas por usuários administrador.
RN5	WEB	Permitir persistência de dados apenas se todos os campos obrigatórios estiverem preenchidos.
RN6	WEB	Permitir cadastro de campeonatos apenas aos administradores.
RN7	WEB	Permitir visualização de lista de usuários.
RN8	WEB	Permitir visualização de lista de confrontos.
RN9	WEB	Permitir gerar confrontos apenas se haver usuários cadastrados no campeonato
RN10	WEB	Permitir visualização de replay apenas em confrontos com movimentos já cadastrados.
RN11	WEB	Permitir inscrição em campeonatos com situação aberta.
RN12	FRAMEWORK	Realizar persistência apenas de id, de usuário cadastrado.
RN13	FRAMEWORK	Realizar persistência de jogadas validas.
RN14	FRAMEWORK	Realizar persistência apenas de Jogos cadastrados no modulo WEB
RN15	FRAMEWORK	Realizar persistência apenas de confrontos devidamente cadastrados no modulo WEB.
RN16	FRAMEWORK	Realizar confronto sem persistência quando usuário informar booleano persistência igual false.

6 LIMITES E RESTRIÇÕES

6.1 LIMITES DA SOLUÇÃO

A aplicação não realiza o confronto diretamente, este confronto vai depender de uma ação humana que ira inserir as classes I.A. enviadas no momento do cadastro dos usuários no campeonato, esta ação é realizado pelo botão “preparar”, em seguida compilar este código para gerar os resultados, na solução atual esta ação é de responsabilidade de um administrador.

6.2 RESTRIÇÕES DA SOLUÇÃO

- A linguagem de programação será JAVA sendo assim é necessário que o ambiente possua o JAVA;
- Para a utilização do sistema é recomendado uma velocidade de conexão de no mínimo 1.0 *mbps.*
- Recomenda – se que a resolução de tela seja de 1366x768 pixels ou superior pois as telas do módulo web foram criadas para esta resolução.
- Possuir instalado um navegador compatível com XHTML 1.0 *Transitional*, como *Firefox, Internet Explorer,* entre outros.

7 DESCRIÇÃO DOS USUÁRIOS

A tabela – 4 apresenta a descrição dos usuários do sistema.

Tabela 4 - Descrição dos Usuários

ID	NOME	DECRIÇÃO
U1	Administrador	Responsável por liberar o cadastro de novos usuários, gerenciar os campeonatos e realizar os confrontos no framework.
U2	Participante	Realiza o download dos arquivos para desenvolver sua I.A., analisa resultados por confronto, usuário ou campeonato, cadastra-se em campeonatos com situação de inscrição com o upload do seu arquivo de I.A.

8 DESENVOLVIMENTO E ARQUITETURA DO SISTEMA

8.1 ARQUITETURA

Ambos os módulos são dispostos em uma arquitetura MVC com separação por pacotes o diagrama pode ser visualizado na figura 5 que ilustra o diagrama de pacotes conforme separação das classes:

- Pacote MODEL: consistido por classe com a funcionalidade de representar as tabelas da BD.
- Pacote DAO: apresentara classe responsáveis por fazer a persistência na BD e as consultas necessárias partindo do principio que cada classe da model tem uma DAO equivalente.
- Pacote MANAGER: contem as classes que aplicam as regras de negocio de cada uma das telas, sendo assim existi um manager para cada interface no modulo WEB, este pacote esta presente no modulo web apenas, devido ao framework não necessitar de telas de apresentação.

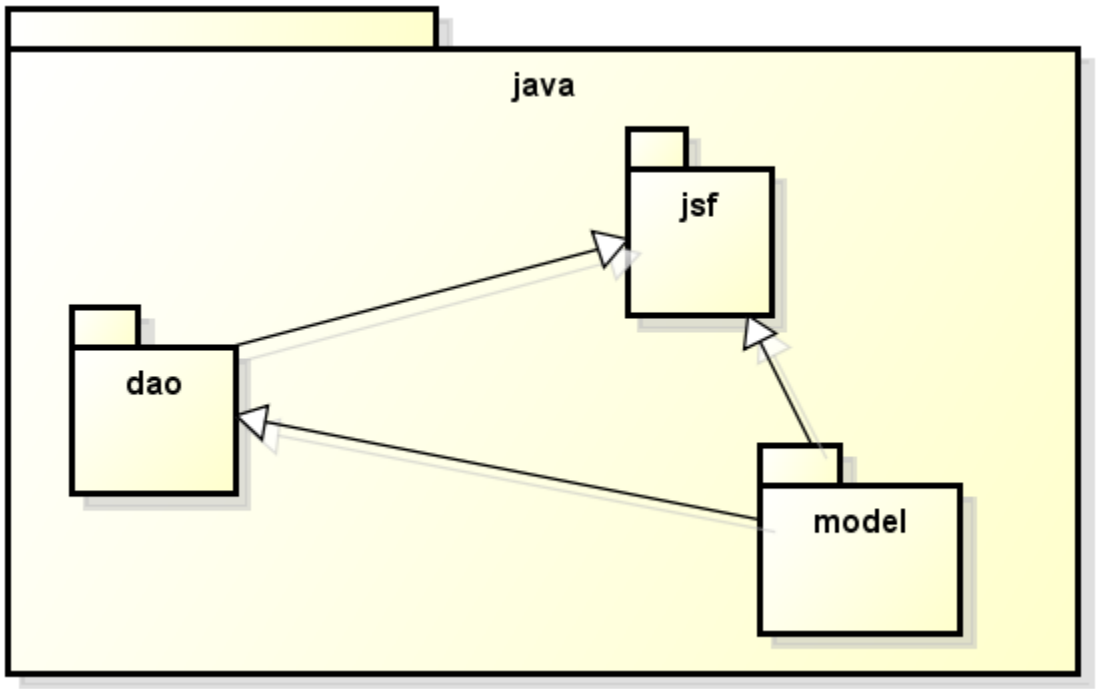


Figura 5 – Diagrama de pacote web

8.2 METODOLOGIA

A metodologia adotada para projeto foi o Cascata, devido a ter os requisitos bem definidos e sabendo que não existira alteração durante o desenvolvimento, é possível finalizar por completo cada etapa antes de iniciar a próxima.

O modelo cascata foi proposto por Royce em 1970 e durante aproximadamente 10 anos foi o único modelo com aceitação geral, este modelo é um dos mais importantes e serve de padrão para diversos outros modelos, grande parte do sucesso do modelo cascata esta no fato de ele ser voltado para a documentação (LEITE, 2007).

9 TÉCNOLOGIAS UTILIZADAS

Postgres / PgAdmin III

Postgres, ferramenta em código aberto para armazenamento e gerencia de dados (POSTGRES, 2013), juntamente com o PgAdmin III, recurso de desenvolvimento projetado para atender as necessidades do usuário e ajudar na gerencia do postgres (PGADMIN, 2013). Ambas as ferramentas foram utilizadas para gerenciar e armazenar os dados do projeto web e do framework.

NetBeans

IDE NetBeans é um ambiente de desenvolvimento de código aberto que ajuda os desenvolvedores a criar projetos web, desktop e aplicações moveis nas linguagens PHP, Java(SE,EE,FX ou ME), HTML5, Groovy, Java Card (NETBEANS, 2013).

Esta ferramenta foi utilizada para auxiliar o desenvolvimento e gerencia do código e também ira auxiliar na etapa de testes.

Java

Java é uma linguagem de programação Orientada a Objetos desenvolvida em 1995, na empresa Sun Microsystems (JAVA, 2013).

Esta tecnologia foi utilizada no desenvolvimento do código.

Java Server Faces

JSF é uma framework com características MVC para WEB focado em interfaces baseadas em eventos. Por seguir o padrão MVC, uma de suas melhores vantagens é a clara separação entre a visualização e regras de negócio(JSF, 2013).

Foi utilizada para desenvolver a interface das paginas web que serão acessadas pelo usuário.

Apache TomCat

O software Tomcat, desenvolvido pela Fundação Apache, permite a execução de aplicações para web (APACHE TOMCAT, 2013).

10 CRONOGRAMA INICIAL

Esta seção é responsável por apresentar na Tabela – 5 uma representação do cronograma inicial.

Tabela 5 - Cronograma inicial

Marcos do Projeto	Artefatos	Data Prévia de Início	Data Prévia de Término
Preparar Ambiente	Caracterização das necessidades para o desenvolvimento.	01/10/2013	10/10/2013
Estudar o Processo	Plano de iterações.	11/10/2013	20/10/2013
Estudar Tecnologias	-	21/10/2013	30/10/2013

- **Preparar ambiente**: etapa que foi instalada todas as ferramentas para o desenvolvimento do projeto.
- **Estudar processo**: etapa que foi realizada a analise das fases e processos adotados para o desenvolvimento do projeto.
- **Estudar tecnologia**: estudo das tecnologias que foram adotadas durante a implementação.

11 CRONOGRAMA OFICIAL

O cronograma oficial e as iterações descritas na seção 8 são apresentados na Tabela 6.

Tabela 6 - Cronograma oficial

2013/2014										
Atividades/ Meses	**Set**	**Out**	**Nov**	**Dez**	**Jan**	**Fev**	**Mar**	**Abr**	**Mai**	**Jun**
Estudo do Processo e Tecnologia e construção do framework										
Construção do módulo web.										
Análise dos Resultados e Redação do Trabalho de Diplomação.										
Preparação da Defesa										
Reunião com Orientador										

12 MODO DE USO

12.1 MÓDULO WEB

Usuário

Conforme ilustrado na figura 6 os usuários devem ser cadastrados por um usuário administrador, este administrador por sua vez também pode cadastrar novos administradores.

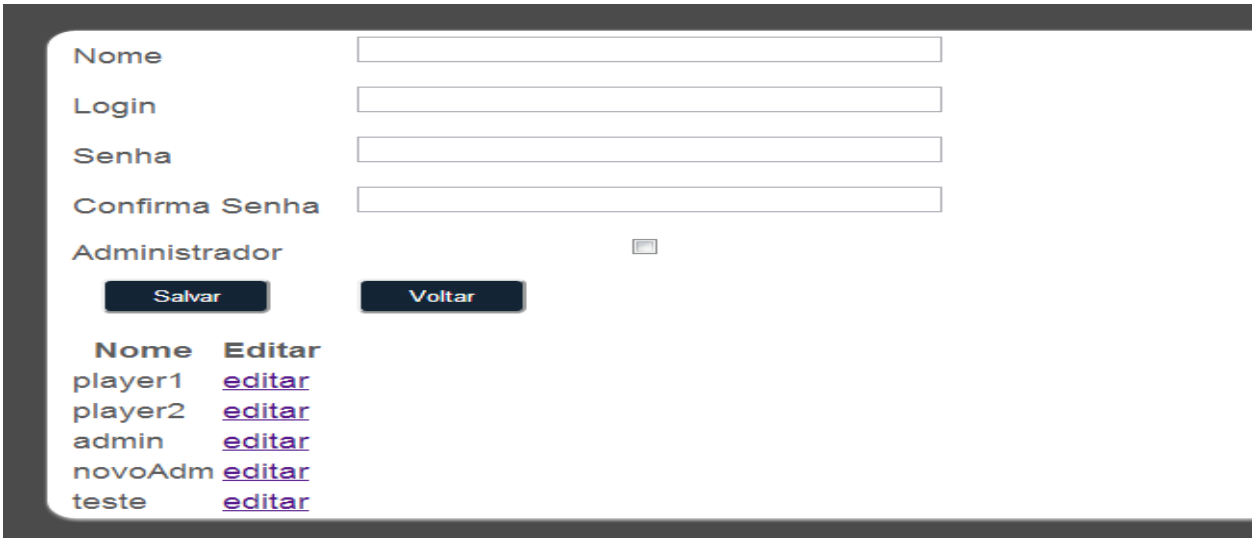


Figura 6 – Tela de cadastro de usuários

Jogo

O cadastro de jogo representa quais modalidades de jogos estão implementadas no framework, o cadastro destes poderá ser efetuado apenas por administradores conforme a figura 7, o intuito deste cadastro é para que no momento de abrir um campeonato seja selecionado qual a modalidade de jogo, para que os alunos que efetuarem o cadastro saibam qual tipo de I.A. devem desenvolver.

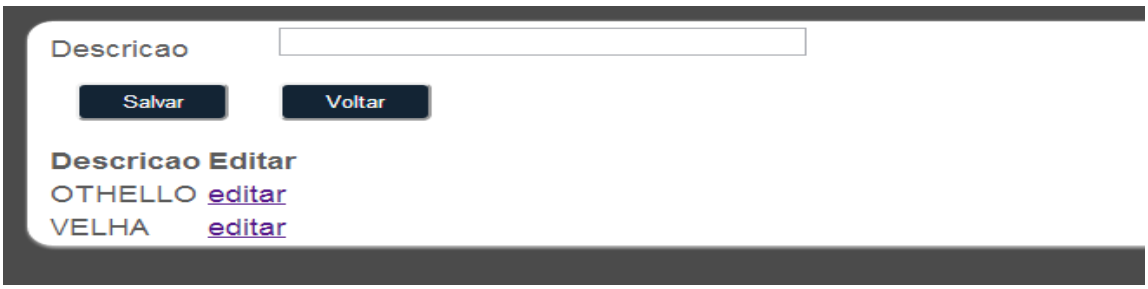


Figura 7 – Tela Jogo

Campeonato

A figura 8 demonstra como o usuário administrador poderá cadastrar campeonatos, para isto ele deve escolher qual o tipo de jogo deste campeonato.

Figura 8 – Tela Campeonato 1

O mesmo poderá também fechar o cadastro de usuários no campeonato e gerar o confronto entre os usuários que já estiverem cadastrados, a tela responsável por estas ações pode ser vista na figura 9 tais dados são exibidos apenas em edições de campeonatos, os usuários cadastrados e os confrontos gerados estão realçados.

Figura 9 – Tela campeonato 2

Os usuários normais terão acesso apenas a lista de usuários cadastrados neste campeonato e a lista de confrontos gerados esta visão esta apresentada na figura 10, com a lista de confrontos os usuários poderão visualizar o replay dos jogos conforme figura 11.

Figura 10 – Tela campeonato 3

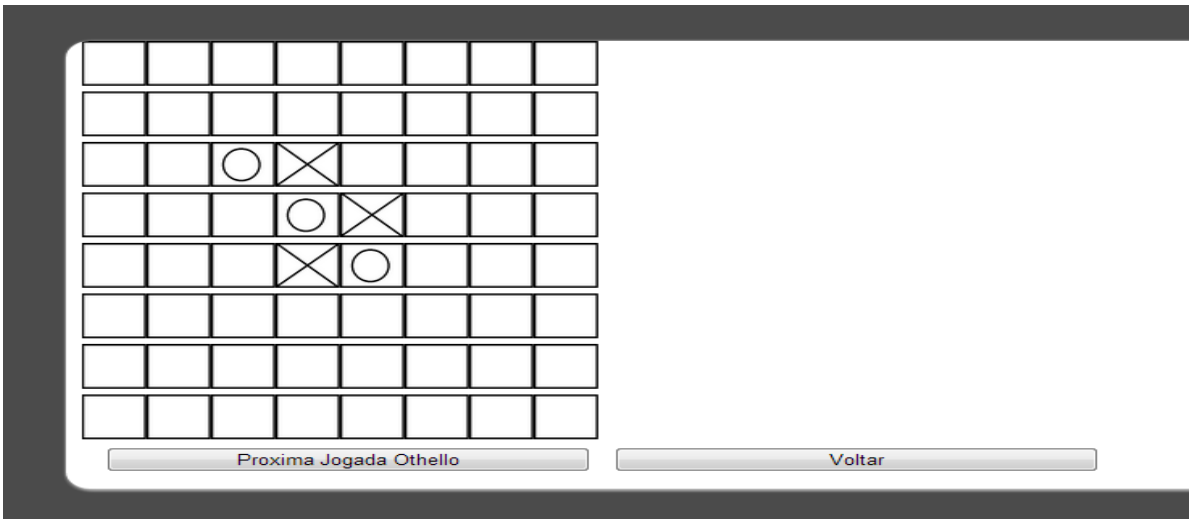


Figura 11 – Tela replay

Na figura – 12 está apresentado a tela com o resultado do campeonato realizado entre os jogadores dos alunos e dois players que realizam jogadas randômicas.

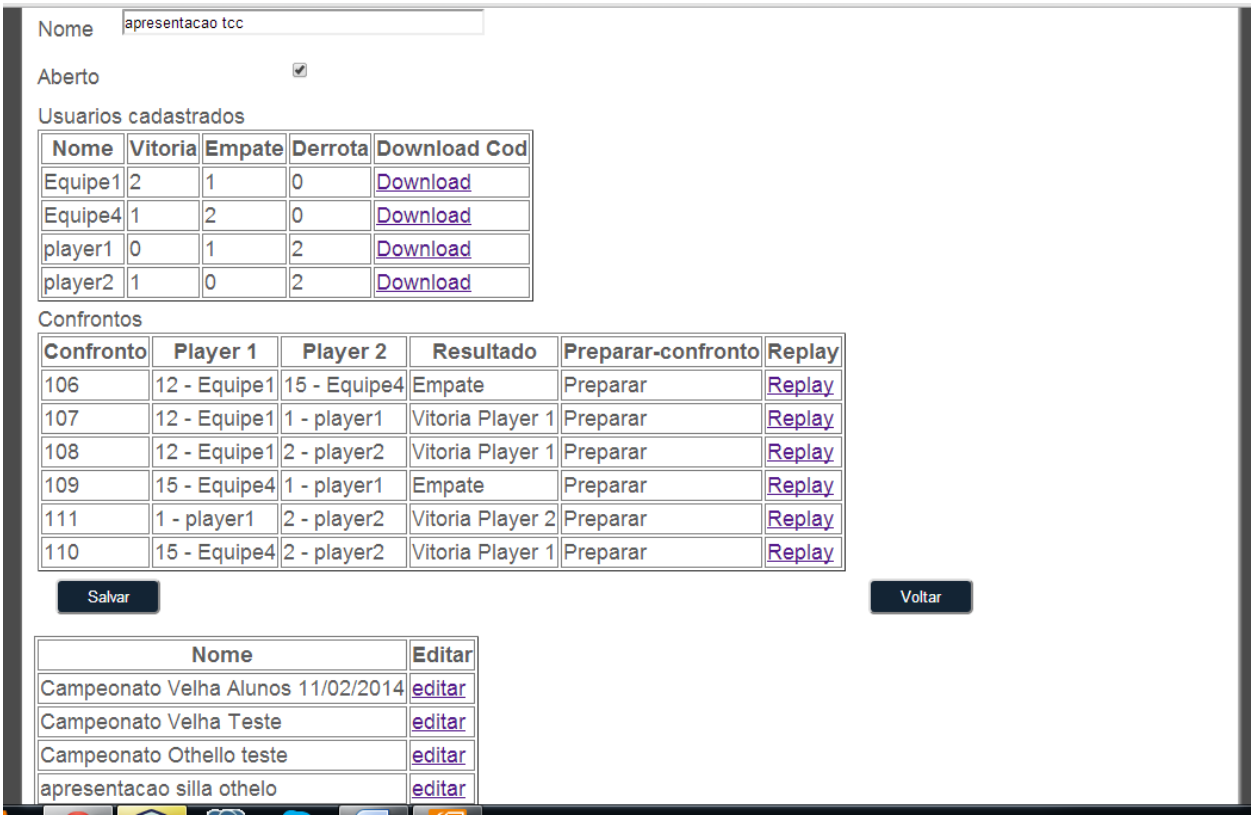


Figura 12 - Tela resultado de campeonato de Alunos

Inscrição em campeonato

O usuário logado seleciona o campeonato que deseja participar e informa o código fonte do seu jogador na área de texto informada, este modo de informar o código da I.A. foi optado pois futuramente pode ser feita uma validação do código java na hora do cadastro e com isso será economizado tempo na hora do torneio pois jogadores com erro de sintaxe não serão cadastrados, uma limitação é o uso de uma única classe, porem como a linguagem Java nos possibilita o uso de classes internas isto não limitara o uso da orientação a objetos, sendo assim esta forma de envio apenas ira garantir que não existirão jogadores com códigos invalido, deste modo ira agilizar o fluxo na hora de efetuar os confrontos, esta tela esta ilustrada na figura 13.

Figura 13 – Tela de inscrição.

12.2 FRAMEWORK

Usuário

Estará disponível para download no site uma copia do código fonte para que os interessados possam criar seus players, incluso neste código fonte temos um player para o jogo da Velha e outro para o Othello, estes players podem ser usados para teste visto que eles buscam apenas o próximo movimento, a seleção destas I.A. previamente desenvolvidas será realizada como se fosse um outro jogador, porém a persistência devera estar desativada pois elas não possuem um id de usuário válido.

Criando um player

Os usuários podem criar um Player a partir da classe AbstractPlayer apresentada na figura 14 e disponível no código fonte, esta classe deve ser estendida pois seus métodos são chamado na classe principal, sendo assim o usuário Administrador que tiver acesso ao servidor irá apenas substituir estas classes no framework e compilar para que o confronto seja realizado.

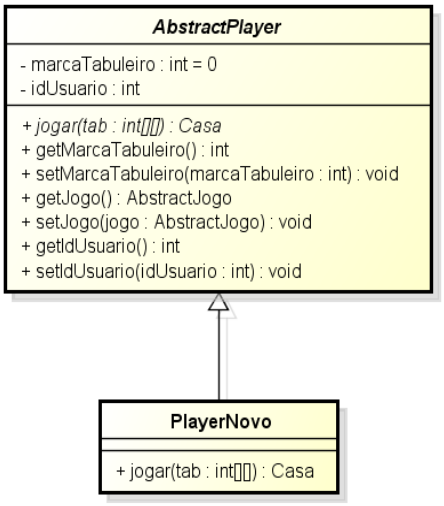


Figura 14 – Novo Player

Criando um jogo

Os interessados em criar um novo validador de regras devem estender a classe AbstractJogo, e implementar sua regra de negócio conforme os métodos que são chamados na classe principal, esta ação esta ilustrada na figura 15.

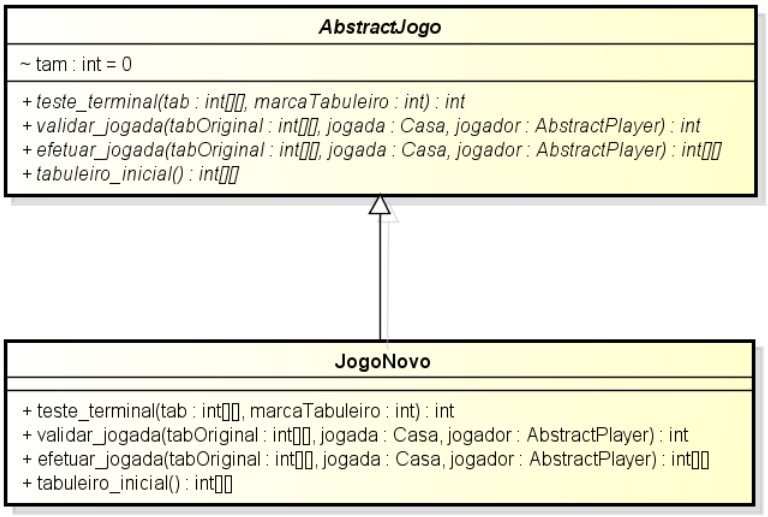


Figura 15 – Novo Jogo

Efetuando um confronto

O responsável por efetuar os confrontos deve substituir os players no projeto, informar o tipo de jogo e os ids dos usuários do Player1 e Player2, em seguida ele ira informar true para o parâmetro de persistência e executar o modulo, após isso o resultado do jogo já poderá ser visto no replay do confronto no módulo web.

O trecho de código que devera ser alterado na classe Jogar é apresentado na Figura 16.

```
16      * @author luri
17      */
18     public class Jogar {
19
20         static int X = 1, O = -1, validar, CONFRONTO = 1;
21         static Boolean persistencia = false;
22         private static Connection conexao = null;
23
33     //         alterna entre os jogos
34             if (true) {
35                 jogo = new JogoVelha();
36                 player = new PlayerVelha();
37                 player.setMarcaTabuleiro(X);
38                 player.setJogo(jogo);
39                 player.setIdUsuario(1);
40
41                 player2 = new PlayerVelha();
42                 player2.setMarcaTabuleiro(O);
43                 player2.setJogo(jogo);
44                 player2.setIdUsuario(2);
               } else {
                 jogo = new JogoOthello();
                 player = new PlayerOthello();
48               player.setMarcaTabuleiro(X);
49               player.setJogo(jogo);
50
                 player2 = new PlayerOthello();
52               player2.setMarcaTabuleiro(O);
53               player2.setJogo(jogo);
54           }
```

Figura 16 – Trecho de código a ser alterado a cada confronto

13 DIAGRAMAS

13.1 DIAGRAMA DE ENTIDADE RELACIONAMENTO.

O diagrama de entidade relacionamento, apresentado na Figura 17, é compartilhado pelos dois módulos da plataforma.

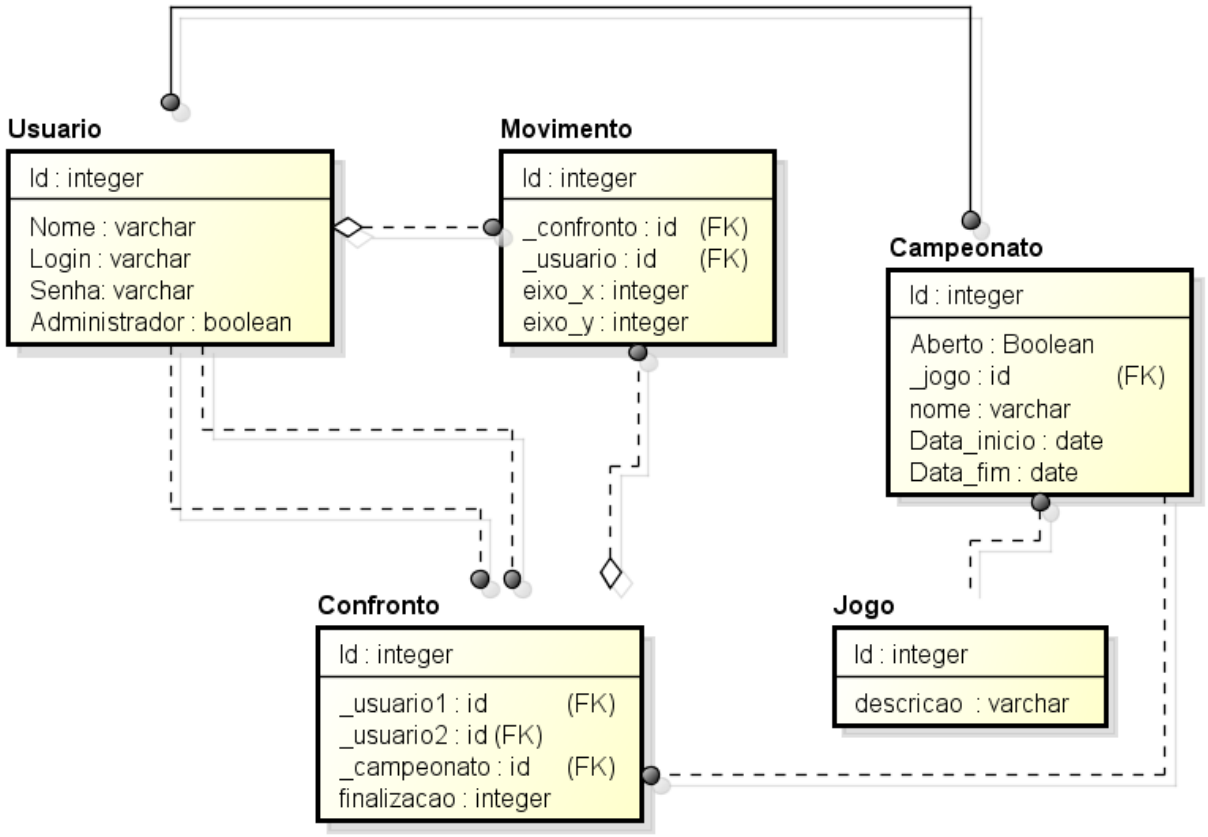


Figura 17 – Diagrama entidade Relacionamento

13.2 MÓDULO WEB

No diagrama de classe apresentado na figura 18 e 19 estão visíveis as classes modelo desenvolvidas no para o modulo web.

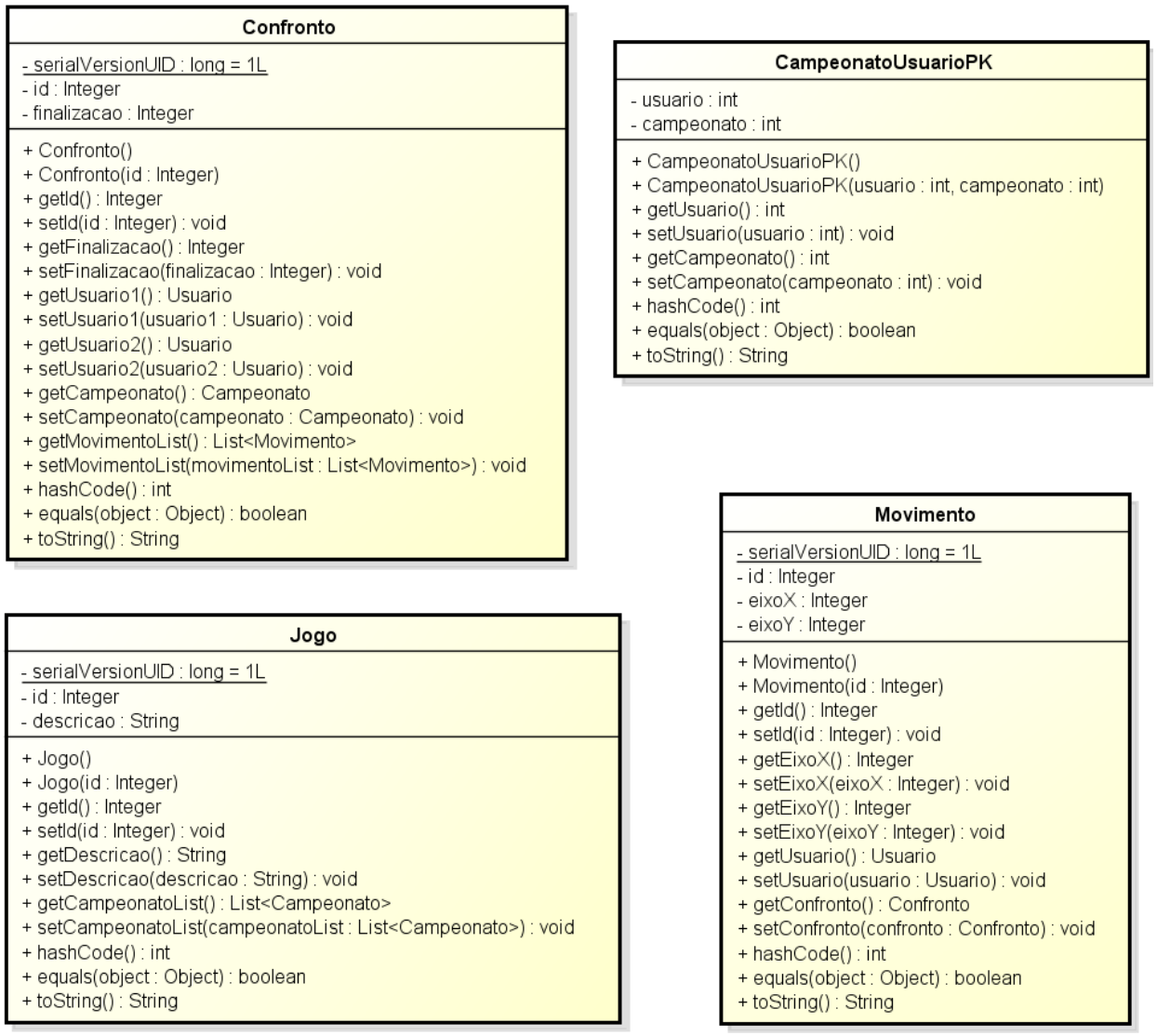
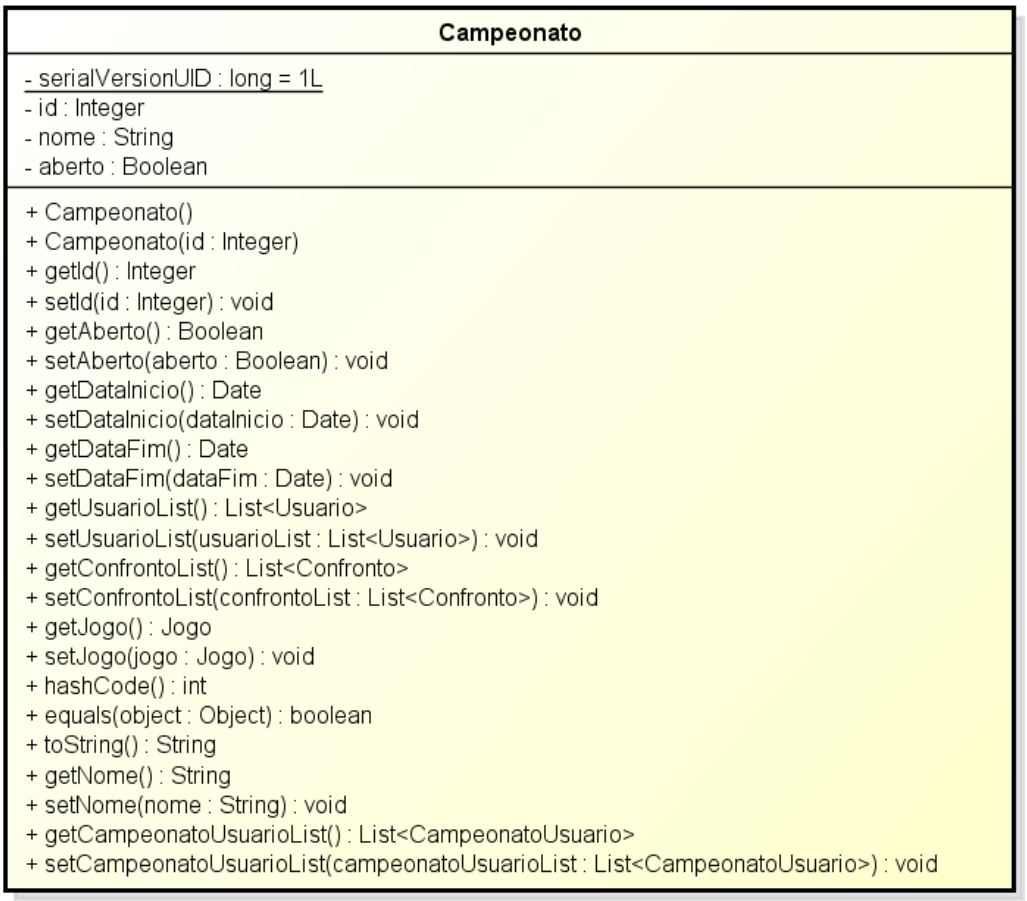


Figura 18 - classes model Web 1



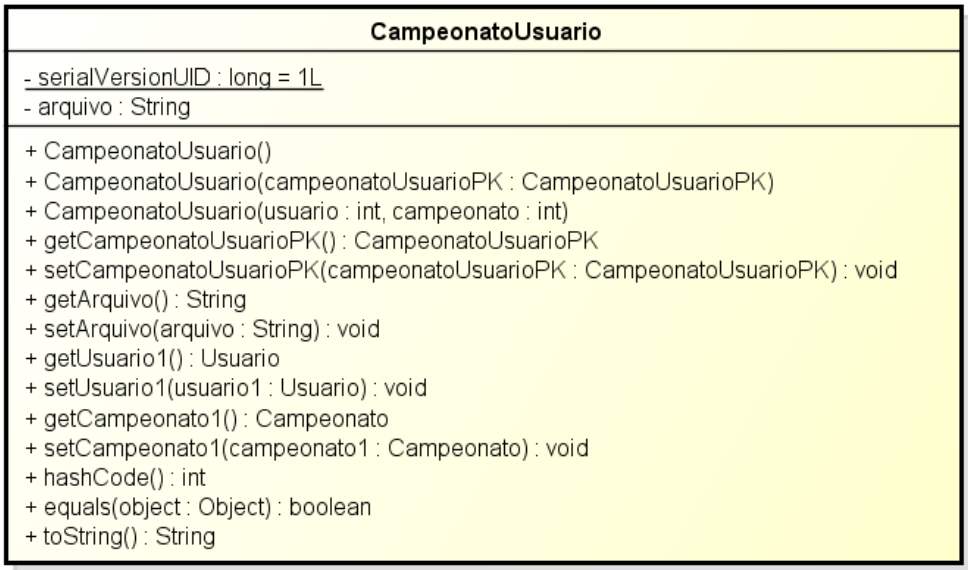


Figura 19 - classes model Web 2

13.3 FRAMEWORK

O diagrama de classe ilustrado na figura 20 e figura 21 apresenta a estrutura atual das classes do framework e o diagrama de classes apresentado na figura 22 ilustra onde devem acontecer as alterações para que seja adicionado novos players e jogos a este modulo.

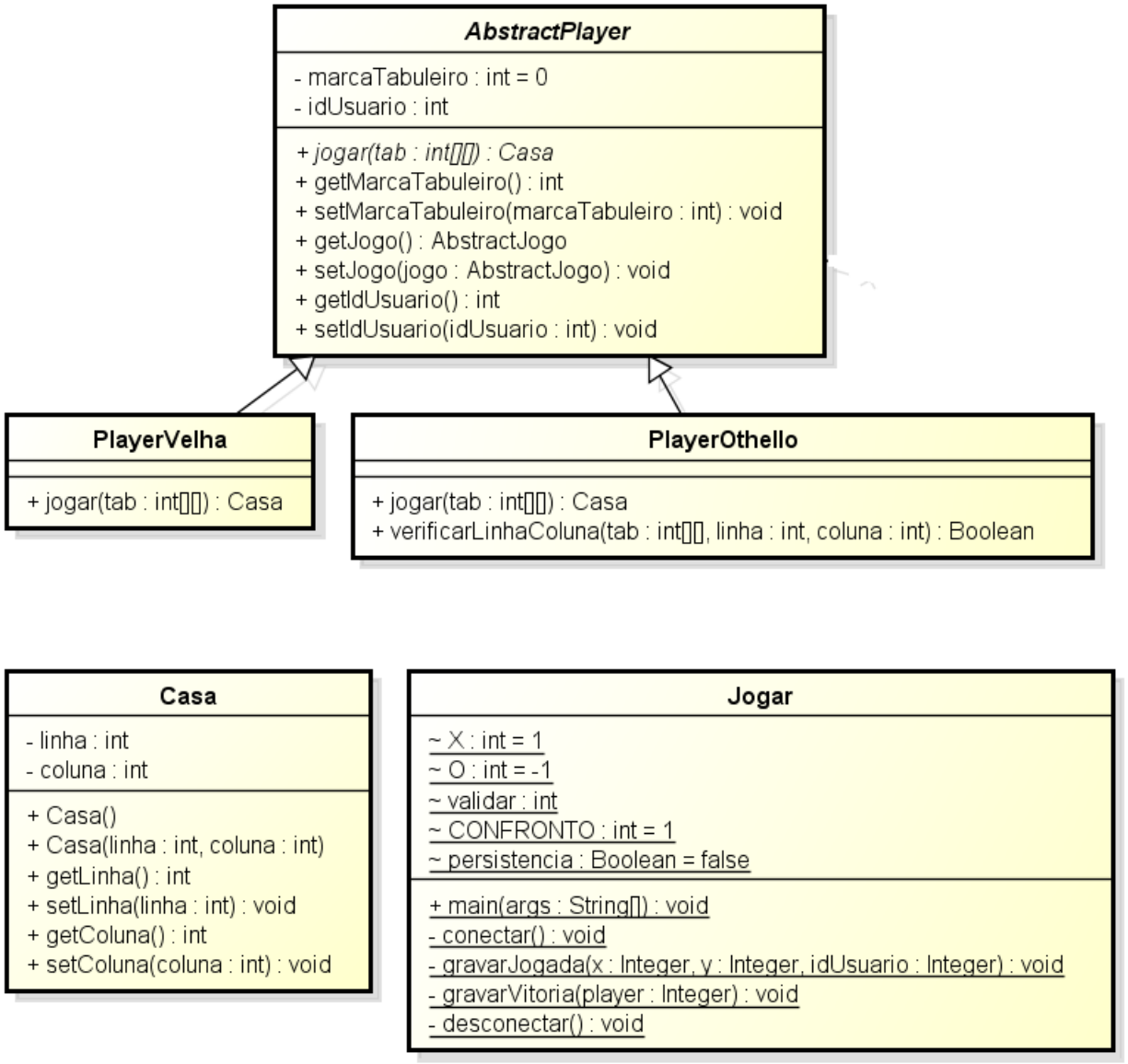


Figura 20 – Diagrama de classe framework parte 1

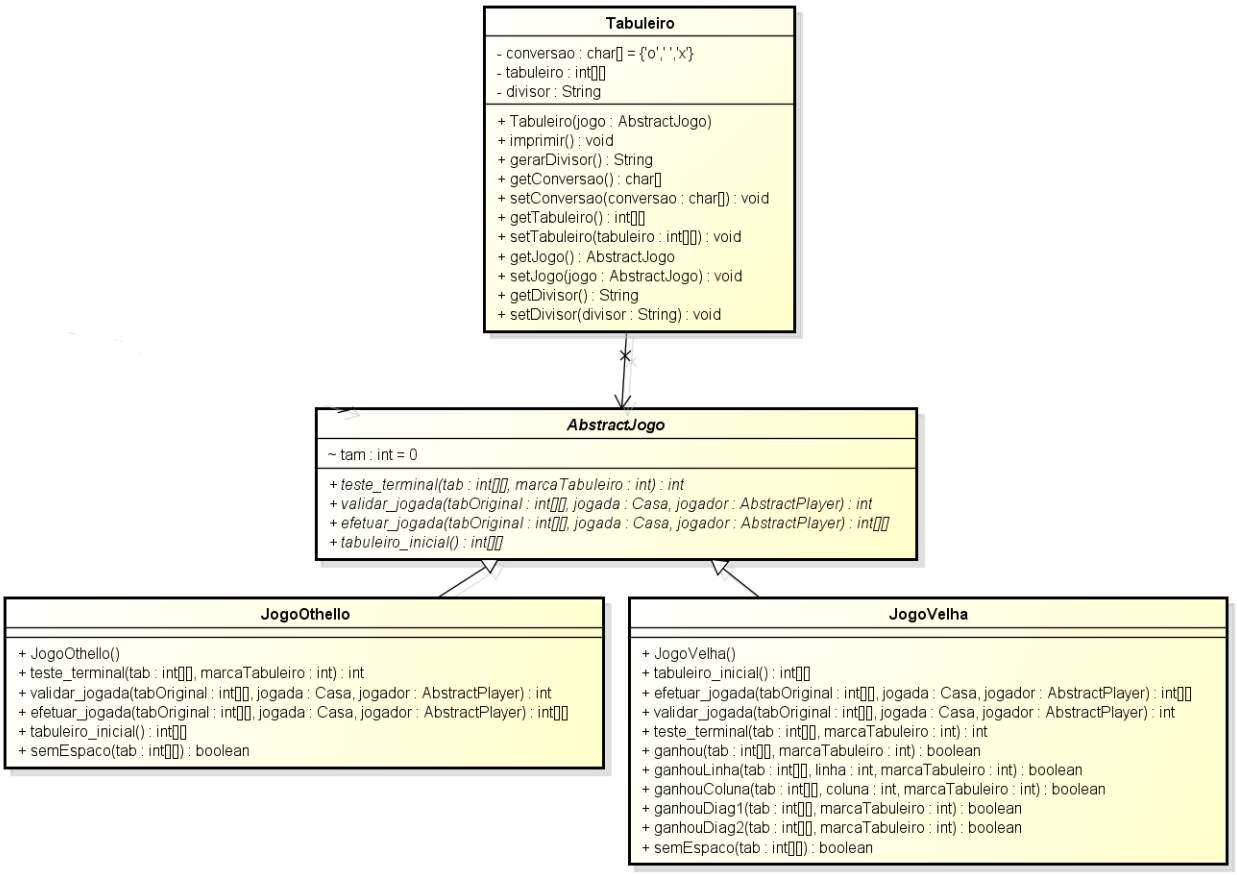


Figura 21 – Diagrama de classe framework parte 2

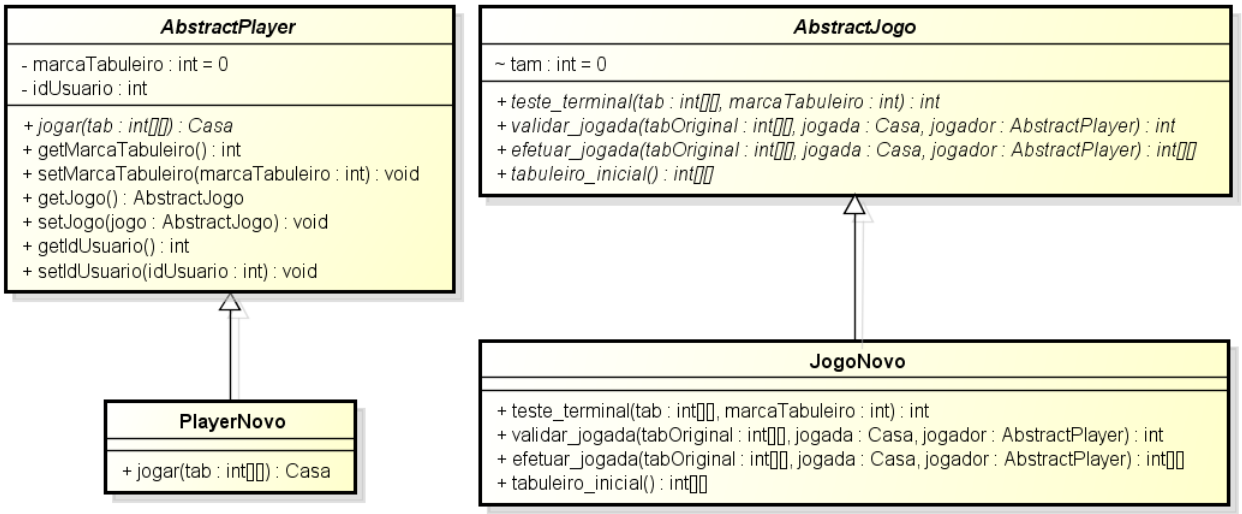


Figura 22 – Trecho de código a ser inserido para novo player e novo jogo

14 EXEMPLO DE EXTENSÃO DA PLATAFORMA PARA UM NOVO JOGO.

Nesta seção apresentamos como a plataforma pode ser extendida para a criação de novos jogos. Para efeitos de ilustração, vamos considerar a implementação do jogo de damas.

A figura 23 apresenta a criação de um novo modulo de jogo chamado JogoDama, para o funcionamento é necessário implementar os métodos abstratos teste_terminal(), validar_jogada(), efetuar_jogada() e tabuleiro_inicial().

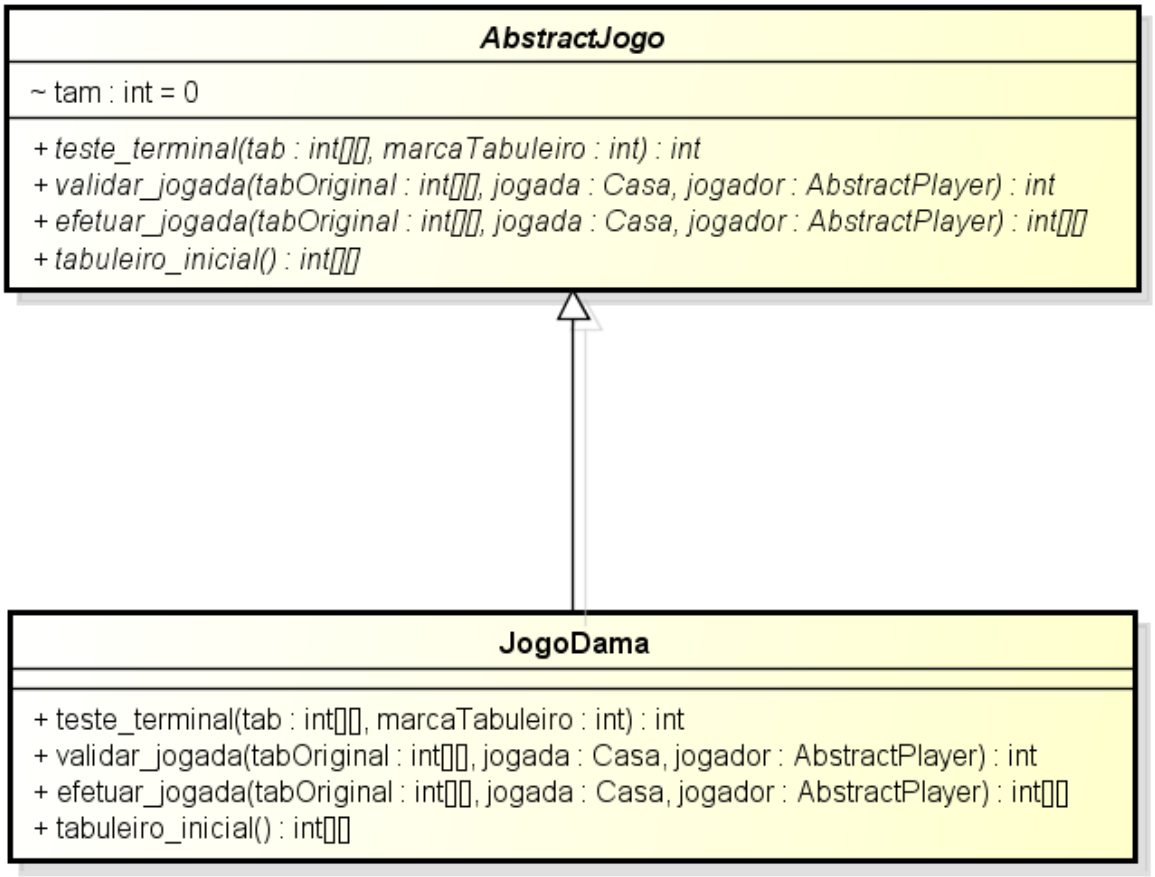


Figura 23 – Diagrama de classe para criar jogo Dama

O método teste_terminal() é responsável por verificar se o jogo teve um vencedor ou empate.

O método validar_jogada() tem como objetivo receber qual casa o jogador deseja jogar, e responder se esta jogada é possível ou não.

O método efetuar_jogada() é responsável por realizar a jogada no tabuleiro a partir da casa que o jogador solicitou, neste método sera realizada as demais alterações do tabuleiro quando necessário, ex: um peça “comida” nas damas ou uma casa convertida no othello.

Por fim o método tabuleiro_inicial() é responsável por colocar as peças no estado inicial do tabuleiro quando necessário, no caso das damas ele so precisa limpar as casas.

15 CONCLUSÃO

Neste trabalho foi desenvolvida uma plataforma web com o proposito de auxiliar os alunos nas aulas de I.A., facilitando o desenvolvimento dos trabalhos que envolvam as técnicas de busca competitiva. A plataforma permite que os alunos possam desenvolver apenas o código da I.A. de um determinado jogo sem se preocupar com os detalhes de interface e de regras de validação, visto que as regras já estão implementadas conforme estrutura apresentada na figura-21.

A plataforma desenvolvida automatiza os campeonatos e confrontos propostos aos alunos que utilizam o espirito competitivo em favor do aprendizado e assim aumentando o interesse dos mesmos em se aprofundar no conteúdo no busca de se sobressaírem no projeto.

Futuramente o sistema pode ser utilizado fora do campus Cornélio Procópio, aumentando o numero de usuários e com isso a complexidade dos códigos desenvolvidos, pois quanto mais jogadores, mais interessante será se sobressair.

Na versão atual da plataforma foram implementados dois jogos: o jogo da Velha e o othello. Porém com base na arquitetura desenvolvida é possível desenvolver diversos outros jogos de tabuleiro. É importante ressaltar que a plataforma foi projetada e desenvolvida de modo a tentar minimizar o impacto da adição de um novo jogo de tabuleiro.

Na ocasião da redação deste trabalho, os alunos da disciplina de I.A., do 7º período do curso de Engenharia da Computação da UTFPR-CP, utilizaram a plataforma para fazer um dos projetos da disciplina.

16 REFERÊNCIAS

APACHE TOMCAT. **Apache Tomcat - Welcome. Apache Tomcat.**
Disponível: em: <http://tomcat.apache.org/index.html>. Acesso em 30 de Julho de 2013

BOLSONI, 2007. **Canadense cria jogo de damas invencível**.
Disponível em: <http://onoticias.blogspot.com.br/2007/07/canadense-cria-jogo-de-damas-invencvel.html>. Acesso em 30 de Julho de 2013.

JAVA. **O que é a tecnologia Java e por que é necessária?.**
Disponível em: <http://www.java.com/pt_BR/download/faq/whatis_java.xml>. Acesso em 30 de Julho de 2013

JSF. **Apresentação.**
Disponível em: < http://www.trainingtecnologia.com.br/java/java-server-faces-jsf >. Acesso em 30 de Julho de 2013

KURZWEIL, Ray. The Age of Spiritual Machines. Massachusetts: The MIT Press, 1990.

RUSSEL, Stuart e NORVIG, Peter. Inteligência Artificial. Editora: Prentice Hall, São PauloSP, 1995.

LEITE, 2007. **O Modelo Cascata.**
Disponivel em: <http://engenhariadesoftware.blogspot.com.br/2007/03/o-modelo-cascata.html>. Acesso em 03 de Agosto de 2013.

NETBEANS. **NetBeans IDE 7.3.1 Release Information**.
Disponível em: <https://netbeans.org/community/releases/73/>. Acesso em 30 de Julho de 2013

PGADMIN. **Introduction**.
Disponível em: <http://www.pgadmin.org/>. Acesso em 30 de Julho de 2013

POSTGRESQL. **Sobre o PostgreSQL**.
Disponível em: <http://www.postgresql.org.br/sobre>. Acesso em 30 de Julho de 2013

SETTI, 2013. **A primavera da inteligência artificial.**
Disponível em: <http://veja.abril.com.br/blog/ricardo-setti/tema-livre/a-primavera-da-inteligencia-artificial/>. Acesso em 30 de Julho de 2013

OTHELLO, 2013. **Jogos t45.**
Disponível em: <http://jogos.t45ol.com/jogo/843/othello.html>. Acesso em 30 de Julho de 2013

VELHA, 2013. **Jogos da velha.**
Disponível em: <http://www.jogosdesustos.com/2012/07/jogo-da-velha-assustador.html>. Acesso em 30 de Julho de 2013

ARENA, 2014. **Engine Testing**
Disponível em: <https://chessprogramming.wikispaces.com/Engine+Testing>. Acesso em 21 de Fevereiro de 2014