

**UNIVERSIDADE POSITIVO  
NÚCLEO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
ENGENHARIA DA COMPUTAÇÃO**

## **TABULEIRO DE XADREZ INTERATIVO**

**Cristiano Moreira da Costa**

**Monografia apresentada à disciplina Trabalho de Conclusão do Curso como requisito parcial à conclusão do Curso de Engenharia da Computação, orientada pelo Professor Alessandro Zimmer.**

**UP/NCET**

**Curitiba**

**2008**

# TERMO DE APROVAÇÃO

Cristiano Moreira da Costa

Tabuleiro de Xadrez Interativo

Monografia aprovada como requisito parcial à conclusão do curso de Engenharia da Computação da Universidade Positivo, pela seguinte banca examinadora:

Professor Alessandro Zimmer (Orientador)

Professor Edson Pedro Ferlin

Professora Maristela Regina Weinfurter

Curitiba, 08 de dezembro de 2008.

## AGRADECIMENTOS

Este projeto é o resultado de muito trabalho, esforço e dedicação ao longo deste ano e que não poderia ter ocorrido sem a compreensão, apoio, amizade ou ensinamentos das pessoas que cito aqui e que contribuem cada um com um importante aspecto em minha vida.

Agradeço a melhor mãe e ao melhor pai do mundo, Chirlei e Ernani da Costa, cuja presença foi, é e será sempre fundamental em minha vida e sem os quais nada disso seria realidade hoje. Muito obrigado não só pelos maiores presentes do mundo que são a vida, educação, valores e princípios como também pelo apoio e direcionamento.

Meus avós e padrinhos Tereza e Osny Moreira (*in memoriam*), sempre presentes em minha vida e que me são tão queridos como meus pais.

Sandra Aiolfi, minha namorada, por sempre ter acreditado em mim, por ser o meu porto seguro e completar o meu ser.

Gislaine Costa, minha querida irmã que mesmo sendo mais nova sempre trouxe palavras sábias de apoio quando eu menos esperava.

Aos amigos pela compreensão da minha ausência este ano.

Aos professores do curso tanto pelos conhecimentos curriculares repassados quanto pelas lições do que representa ser um engenheiro.

## RESUMO

Este trabalho apresenta a proposta de desenvolvimento de um tabuleiro de xadrez interativo construído com o objetivo de aprofundar o conhecimento nos conceitos computacionais Interface Humano Computador (IHC), Sistemas Distribuídos e *Rich Internet Applications* (RIA) fazendo uso de uma película sensível ao toque e sua respectiva placa controladora, uma interface gráfica construída em Java FX representando o tabuleiro e a validação das jogadas implementada com Java RMI. Toda a interação dos jogadores com o tabuleiro se dá via a película sensível ao toque que envia sinais elétricos para sua placa controladora, responsável por traduzir esses sinais para coordenadas cartesianas correspondentes ao local tocado. Essas coordenadas são repassadas ao tabuleiro que determina qual a peça escolhida para ser movimentada e para onde ela será movida. Após a captação dessas informações é acionado o módulo de validação da jogada que funciona de forma distribuída e retornará se a jogada é válida ou não.

**Palavras chave: Xadrez, Interface Humano Computador, Sistemas Distribuídos, *Rich Internet Application*, Película Sensível ao Toque.**

# INTERACTIVE CHESSBOARD

## ABSTRACT

This work presents a proposal about developing an interactive chessboard built with the objective to acquire more knowledge in computer concepts like Human Interface Device (HID), Distributed Systems and *Rich Internet Applications* (RIA) using a touch screen and its respective controller board, one graphical interface built on Java FX representing the board and the validation of the moves using Java RMI. The players will interact with the board using the touch screen, and it will send electrical signals to the controller board, responsible by translate these signals to the cartesian coordinates of the touch. These coordinates are passed to the board and will show what piece will be moved and to where it will be moved. After that the distributed module will receive these informations and will return if the move is valid or not.

**Key words: Chess, Human Interface Device, Distributed Systems, Rich Internet Application, Touch Screen.**

## SUMÁRIO

|   |           |
|---|-----------|
| <u>CAPÍTULO 1 - INTRODUÇÃO .....</u>                      | <u>11</u> |
| <u>CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA .....</u>           | <u>13</u> |
| 2.1 – Xadrez.....   | 13        |
| 2.2 – Interfaces <i>TouchScreens</i> .....                | 16        |
| <u>CAPÍTULO 3 – ESPECIFICAÇÃO DO PROJETO .....</u>        | <u>18</u> |
| 3.1 – Software .....                                      | 18        |
| 3.1.1 – Diagrama de Classe .....                          | 20        |
| 3.1.2 – Diagrama de Caso de Uso .....                     | 20        |
| 3.1.3 – Diagrama de Seqüência.....                        | 21        |
| 3.2 - Hardware.....                                       | 21        |
| 3.3 – Viabilidade Técnico Econômica .....                 | 23        |
| 3.4 – Cronograma do Projeto .....                         | 24        |
| <u>CAPÍTULO 4 – DESENVOLVIMENTO E IMPLEMENTAÇÃO .....</u> | <u>25</u> |
| 4.1 – Interface <i>Touchscreen</i> .....                  | 25        |
| 4.2 – Software .....                                      | 28        |
| 4.2.1 – Representação Gráfica .....                       | 28        |
| 4.2.2 – Servidor.....                                     | 30        |
| <u>CAPÍTULO 5 – VALIDAÇÃO E RESULTADOS .....</u>          | <u>32</u> |
| <u>CAPÍTULO 6 - CONCLUSÃO .....</u>                       | <u>34</u> |
| <u>CAPÍTULO 7 - REFERÊNCIAS .....</u>                     | <u>36</u> |
| <u>ANEXO A – ARTIGO TÉCNICO.....</u>                      | <u>38</u> |
| <u>ANEXO B – MANUAL .....</u>                             | <u>42</u> |

## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 2.1 – Posição inicial das peças de xadrez no tabuleiro.....      | 14 |
| Figura 2.2 – Movimentação do Bispo .....                                | 14 |
| Figura 2.3 – Movimentação da Torre.....                                 | 14 |
| Figura 2.4 – Movimentação da Dama.....                                  | 15 |
| Figura 2.5 – Movimentação do Cavalo.....                                | 15 |
| Figura 2.6 – Movimentação do Peão.....                                  | 15 |
| Figura 2.7 – Movimentação do Rei.....                                   | 16 |
| Figura 3.1 – Diagrama de Blocos do Projeto.....                         | 18 |
| Figura 3.2 – Fluxograma básico de funcionamento do software .....       | 19 |
| Figura 3.3 – Diagrama de Bloco do Hardware .....                        | 22 |
| Figura 3.4 – Cronograma do Projeto e Gráfico de Gantt.....              | 24 |
| Figura 4.1 – Configuração dos <i>Jumpers</i> da Placa Controladora..... | 26 |
| Figura 4.2 – Conexão do cabo USB e cabo da película .....               | 27 |
| Figura 4.3 – Início do processo de calibração para um ponto .....       | 27 |
| Figura 4.4 – Fim do processo de calibração para um ponto .....          | 27 |
| Figura 4.5 – Programa principal da interface gráfica .....              | 29 |
| Figura 4.6 – Interface gráfica apresentando um jogo em andamento.....   | 29 |
| Figura 4.7 – Classe de conexão com o servidor RMI.....                  | 30 |
| Figura 4.8 – Comunicação com o servidor RMI.....                        | 30 |
| Figura 4.9 – Classe Interface.....                                      | 31 |
| Figura 5.1 – Protótipo do tabuleiro .....                               | 33 |
| Figura 5.2 – Protótipo do Tabuleiro Anexado ao Servidor.....            | 33 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 2.1 – Peças do Xadrez .....                             | 13 |
| Tabela 3.1 – Tabela de Custos .....                            | 23 |
| Tabela 4.1 – Formato de Comunicação da Placa Controladora..... | 26 |



## LISTA DE SIGLAS

**NCET** – Núcleo de Ciências Exatas e Tecnológicas

**UP** – Universidade Positivo

**JDK** – *Java Development Kit*

**JVM** – *Java Virtual Machine*

**JRE** – *Java Runtime Environment*

**RC** – *Release Candidate*

**RIA** – *Rich Internet Application*

**RMI** – *Remote Method Instance*

## LISTA DE SÍMBOLOS

$\Omega$  - ohm

## CAPÍTULO 1 - INTRODUÇÃO

A computação tornou-se uma ferramenta imprescindível no cotidiano da humanidade, agilizando processos manuais e permitindo a criação de novas soluções para problemas específicos. O papel do engenheiro da computação é criar e desenvolver ferramentas no intuito de disporem de tecnologia de ponta adequada para o trabalho a que se prestam e para que fiquem cada vez mais naturais em seu uso.

A interatividade entre o homem e o computador tem evoluído desde os primórdios da computação. Antes eram cabos conectores para entrada de dados e fitas de papel perfurado como saída de dados, logo depois os monitores e teclados com essas mesmas funções respectivamente, e posteriormente o mouse.

Uma das últimas evoluções nessa interação homem-máquina são as interfaces sensíveis ao toque que se popularizam em diversos produtos e setores, como os setores comerciais e bancários que já dispõem de máquinas de atendimento ao cliente na forma de quiosques eletrônicos e caixas eletrônicos respectivamente. Tais interfaces permitem uma interatividade muito mais natural e intuitiva em seu uso.

Sua aplicação em aparelhos comerciais é vista em equipamentos de posicionamento global (GPS), tocadores de música (MP3 *Players* como o *iPod Touch* da *Apple*), consoles de videogame (*Nintendo DS*), computadores pessoais portáteis (*Palmtops*) e inclusive novos computadores como o *Microsoft Surface*.

Com o atual estágio de utilização da Internet e a globalização cada vez mais presente no dia-a-dia de organizações empresariais e na vida das pessoas, surgiu a necessidade de mobilidade e disponibilidade dos sistemas de computação atual. Para atender a isso recorre-se aos Sistemas Distribuídos, que também nos fornecem outras vantagens como escalabilidade e portabilidade.

Já o mercado de jogos eletrônicos tem apresentado um crescimento vigoroso na última década, de acordo com SANTEE (2005) e PERUCIA (2005), e se tornou um setor que demanda mão de obra extremamente especializada e crítica, pois seu desenvolvimento não é trivial se comparado com o desenvolvimento de soluções empresariais, por exemplo, além de ser um mercado de alta visibilidade.

Este projeto, pelos motivos acima explicados, pretende integrar tais tecnologias em um tabuleiro de xadrez interativo, como forma de demonstração e aprimoramento dos conhecimentos adquiridos ao longo do curso, e também demonstrar a aptidão em criar soluções eletrônicas envolvendo tecnologias de ponta.

## CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA

### 2.1 – Xadrez

Conforme o regulamento atual da Federação Internacional de Xadrez (FIDE 2008), o xadrez é um jogo para duas pessoas praticado sobre um tabuleiro de sessenta e quatro espaços em um arranjo matricial de oito linhas por oito colunas. Ele contém trinta e duas peças, sendo dezesseis para cada jogador, divididas entre seis tipos diferentes. O objetivo do jogo é capturar a peça adversária mais importante, o rei, pelo cheque mate. Outra característica importante ressaltada em LARAMÉE (2000) é que o xadrez é considerado o jogo de informação perfeita, pois durante toda a partida ambos os jogadores tem conhecimento da situação da partida.













| Peça   | Representação Gráfica  | Quantidade por Jogador |
|--------|--|------------------------|
| Rei    |  ou      | 1                      |
| Dama   |  ou  | 1                      |
| Torre  |  ou  | 2                      |
| Bispo  |  ou  | 2                      |
| Cavalo |  ou  | 2                      |
| Peão   |  ou  | 8                      |

Tabela 2.1 – Peças do Xadrez

A distribuição inicial das peças no tabuleiro é realizada tal e qual demonstrado na figura 2.1.



Figura 2.1 – Posição inicial das peças de xadrez no tabuleiro

Cada peça possui uma movimentação única, que basicamente se dá na seguinte forma:

- O Bispo movimenta-se nas diagonais quantas posições forem convenientes, conforme a figura 2.2.

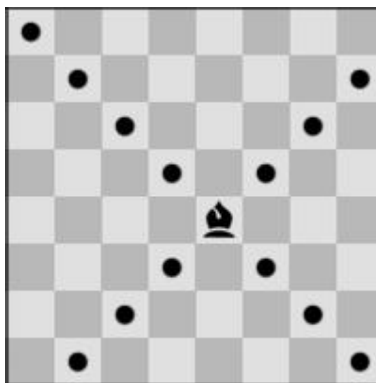


Figura 2.2 – Movimentação do Bispo

- A Torre movimenta-se na horizontal ou na vertical quantas posições forem convenientes, conforme a figura 2.3.

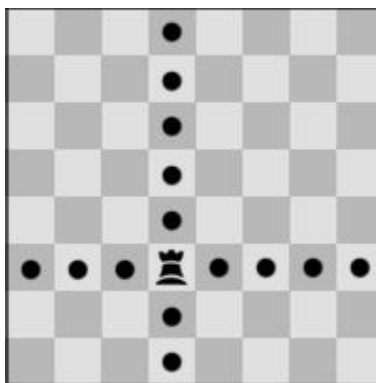


Figura 2.3 – Movimentação da Torre.

- A Dama movimenta-se quantas posições forem convenientes tanto nas diagonais quanto na vertical ou horizontal, como demonstrado na figura 2.4.

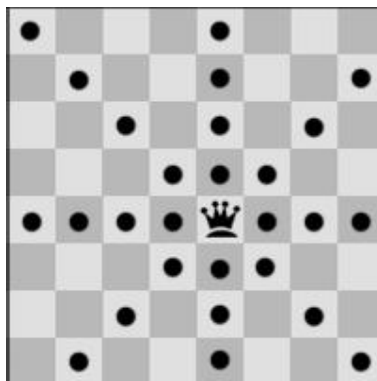


Figura 2.4 – Movimentação da Dama.

- O Cavalo movimenta-se em “L”, ou seja, três casas em um sentido e uma casa a noventa graus desse sentido, conforme a figura 2.5.

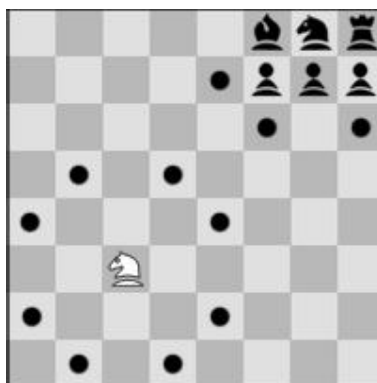


Figura 2.5 – Movimentação do Cavalo.

- O Peão movimenta-se na vertical e captura peças adversárias na diagonal, conforme a figura 2.6.

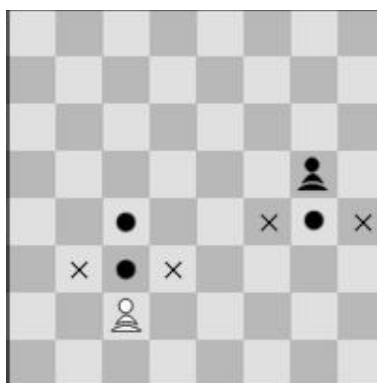


Figura 2.6 – Movimentação do Peão.

- O Rei pode se movimentar para qualquer posição adjacente aquela em que ele se encontra, como visto na figura 2.7.

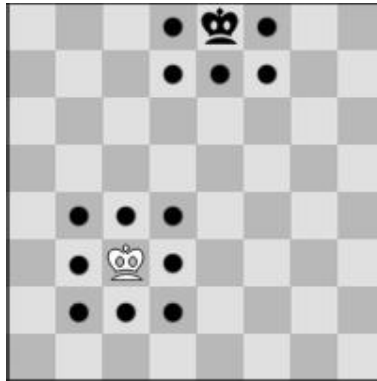


Figura 2.7 – Movimentação do Rei.

## 2.2 – Interfaces *TouchScreens*

Interfaces *TouchScreens* são responsáveis por detectar os toques em uma determinada fonte de imagens, e dessa forma podem perfeitamente substituir ou complementar outras formas de interação homem-máquina, como teclado ou mouse por exemplo. Sua utilização hoje em dia é amplamente aceita como visto em FILHO (2008), haja visto o número de telefones celulares, videogames, *palmtops* e quiosques de atendimento ao público que fazem uso dessa interface. As principais tecnologias de interfaces *touchscreen* são:

- Resistiva: São formadas por duas camadas resistivas que quando se unem no ponto do toque comportam-se como dois divisores de tensão, sendo assim determinado qual foi o ponto do toque. São encontradas películas de quatro, cinco ou oito fios, sendo que quanto maior esse número, maior será a precisão na determinação do ponto de toque. Películas com essa tecnologia são mais baratas mas oferecem menos transparência do que as outras e podem ser danificadas por agentes de forças externos, mas não por poeira ou água e por isso são muito utilizadas atualmente.
- Onda Acústica de Superfície (*Surface Acoustic Wave – SAW*): Utiliza ondas ultrasônicas que passam pela película. Ao ser tocada, parte dessas ondas são absorvidas e é por essa dessa alteração que o toque é detectado. Películas dessa tecnologia podem ser danificadas por elementos externos ou mesmo ter sua eficiência reduzida.
- Capacitiva: As camadas da película são revestidas com material que pode gerar um potencial capacitivo. Pela aproximação de algo que também possua potencial capacitivo,



como o dedo humano, o potencial da película é alterado e analisado para determinar o local do toque. Podem ser acionados tanto pelo toque de um simples dedo como de uma caneta desenvolvida para isso, sendo que não são afetados por elementos externos e possuem boa transparência.

- Infravermelho: Podem trabalhar de duas maneiras, sendo uma delas por temperatura mas que necessita de um contato aquecido e tem uma velocidade de varredura mais lenta se comparado com as outras tecnologias; a outra maneira utiliza-se de uma matriz de sensores infravermelhos que determinam a posição baseando-se na verificação de quais sensores foram interrompidos. São as superfícies mais duráveis e utilizadas em muitas aplicações militares.

## CAPÍTULO 3 – ESPECIFICAÇÃO DO PROJETO

Este trabalho tem como objetivo principal a criação de um tabuleiro de xadrez interativo conectado a um microcomputador para ser jogado por duas pessoas. O diagrama abaixo demonstra os módulos principais do projeto.

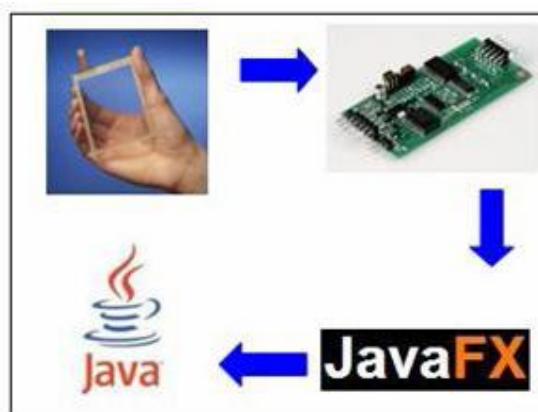


Figura 3.1 – Diagrama de Blocos do Projeto

O requerimento de *hardware* mínimo para execução do projeto é um processador de 1 GHz ou superior, 512 MB de memória RAM e 200 MB de espaço livre no disco rígido.

### 3.1 – Software

Todo o processamento do jogo será feito via camada de *software*, considerando o armazenamento das posições das peças no tabuleiro, a verificação de jogada válida ou inválida e o controle do turno do jogador.

Não fará parte do projeto o controle do tempo, a sugestão de jogadas a partir da peça escolhida, o desenvolvimento de técnicas de inteligência artificial para que o jogador possa jogar contra o computador ou até mesmo um computador possa jogar contra outro computador em virtude da complexidade que isso acarretaria ao projeto..

O fluxograma básico do jogo é demonstrado na figura 3.2:

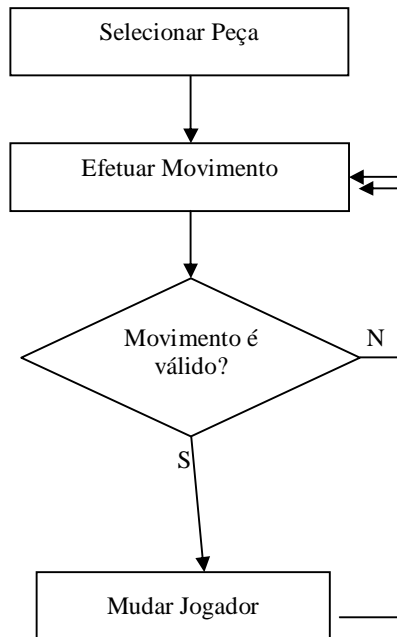
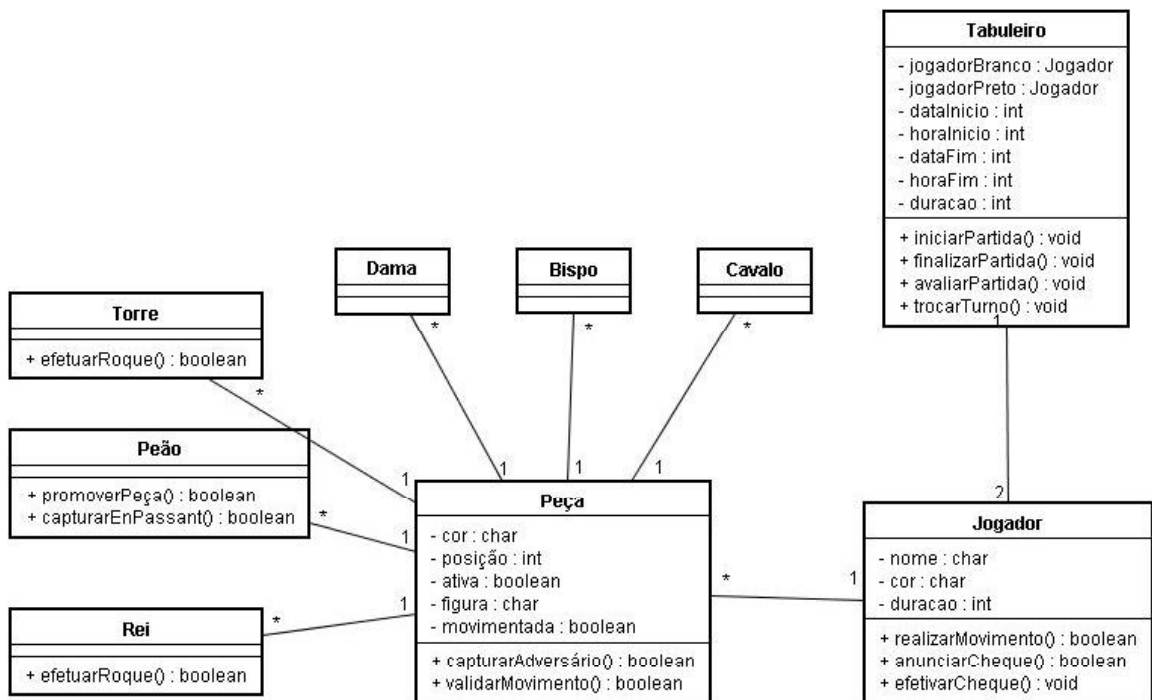


Figura 3.2 – Fluxograma básico de funcionamento do software

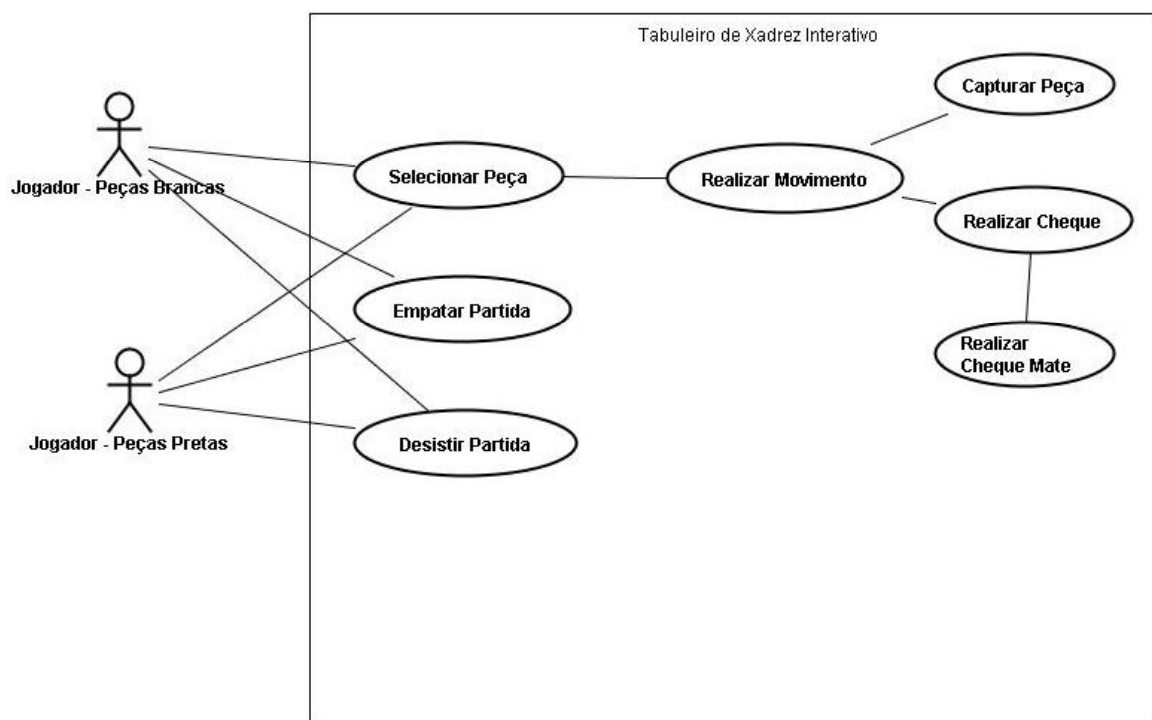
A plataforma de desenvolvimento planejada consiste em sistema operacional *Microsoft Windows*, linguagem java, e IDE *Netbeans*. O ambiente de execução inicial será no mesmo sistema operacional da plataforma de desenvolvimento por definição e para limitar o escopo do projeto, mas por ser feito em java esse ambiente de execução poderia ser em *linux*, por exemplo, bastando para isso que seja efetuado um estudo a respeito de compatibilidade das versões de *software* posteriormente definidos.

Em função da escolha da tecnologia java que é orientada a objetos por definição, é necessário realizar uma análise orientada a objetos cujo resultado pode ser visto nos diagramas de classe, de casos de uso e de seqüência abaixo, de acordo com a metodologia vista em SOMMERVILLE (2007).

### 3.1.1 – Diagrama de Classe

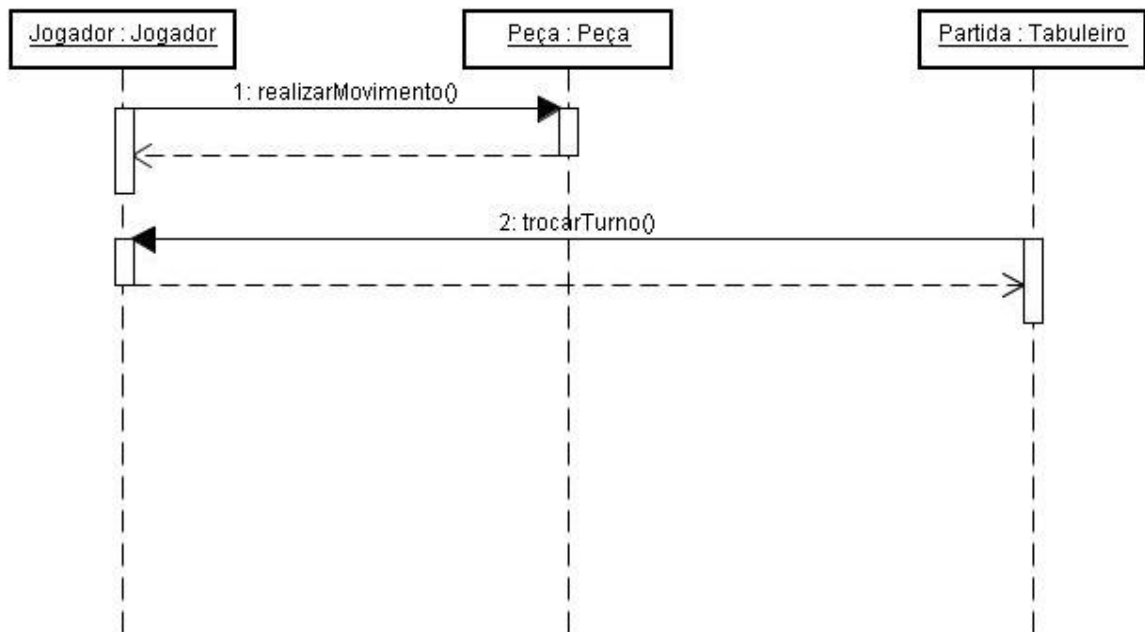


### 3.1.2 – Diagrama de Caso de Uso

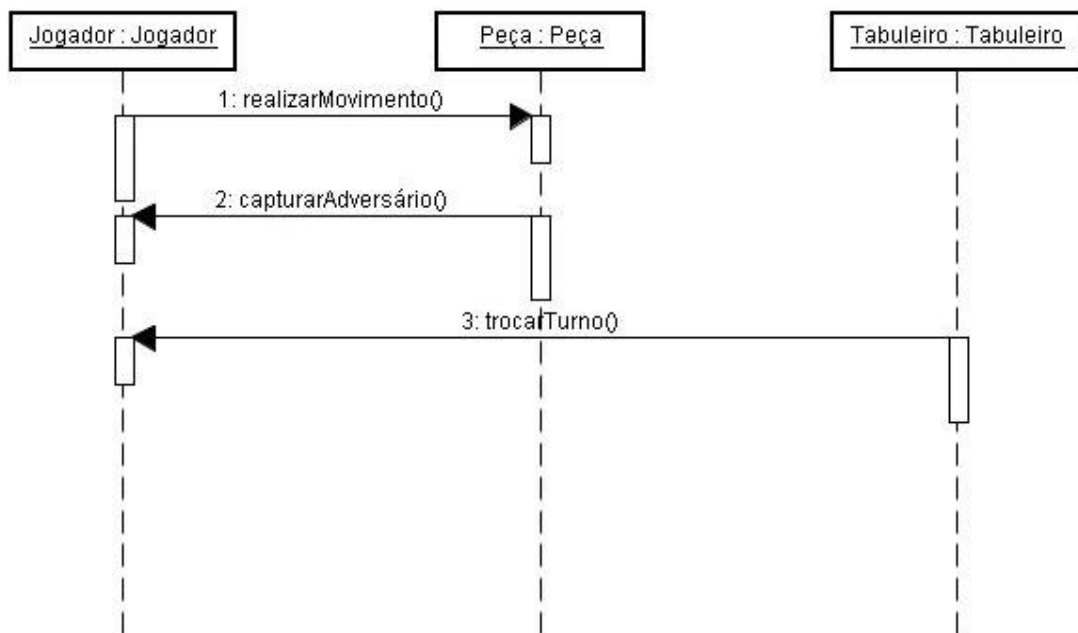


### 3.1.3 – Diagrama de Seqüência

#### 3.1.3.1 – Movimento Normal



#### 3.1.3.2 – Movimento com Captura



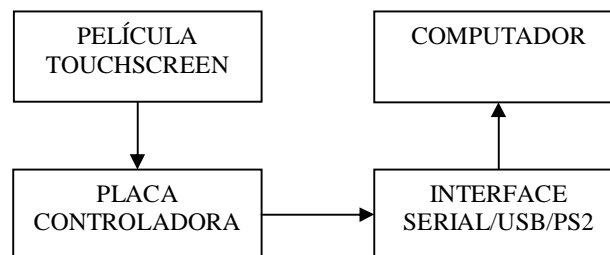
### 3.2 - Hardware

A camada de *hardware* apresenta toda a interatividade necessária para utilização do jogo pelos usuários, e ela se divide entre interface *touchscreen* e placa controladora.

A carga da interface *touchscreen* fica a visualização do jogo e a entrada de dados do projeto. Por visualização entende-se que os jogadores terão o tabuleiro com o posicionamento das peças de acordo com o andamento do jogo e outras informações que poderão ser pertinentes como de quem é o turno, o tempo gasto por jogador no tempo total do jogo e últimas jogadas, e a entrada de dados corresponde unicamente a qual movimento de peça o jogador pretende realizar..

A placa controladora esta conectada a interface *touchscreen* e ao microcomputador. Sua utilização se faz necessária para poder interpretar a informação capturada pela interface que deve ser entregue para o microcomputador interpretar qual é a jogada selecionada pelo usuário. Seu funcionamento, demonstrado na figura 3.3 é feito da seguinte forma:

- Receber os impulsos elétricos da interface touchscreen;
- Processar e converter em coordenadas de eixo cartesiano (X,Y);
- Enviar ao microcomputador pela interface escolhida.



**Figura 3.3 – Diagrama de Bloco do Hardware**

### 3.3 – Viabilidade Técnico Econômica

Os custos inicialmente previstos para este projeto se encontram na tabela 3.1.

|    | A           | B  | C          | D            | E            | F             |
|----|-------------|--|------------|--------------|--------------|---------------|
| 1  | Categoria   | Recurso  | Quantidade | Unid. Medida | Custo        | Custo Total   |
| 2  | Equipamento | Notebook   | 1          | Quantidade   | R\$ 1.000,00 | R\$ 1.000,00  |
| 3  | Equipamento | Desktop  | 1          | Quantidade   | R\$ 800,00   | R\$ 800,00    |
| 4  | Instalações | Instalações  | 9          | Mês          | R\$ 220,00   | R\$ 1.980,00  |
| 5  | Insumos     | Ferramentas  | 1          | Quantidade   | R\$ 300,00   | R\$ 300,00    |
| 6  | Insumos     | Fotocópias   | 1000       | Quantidade   | R\$ 0,10     | R\$ 100,00    |
| 7  | Insumos     | Encadernações                                      | 5          | Quantidade   | R\$ 10,00    | R\$ 50,00     |
| 8  | Insumos     | Livros específicos                                 | 5          | Quantidade   | R\$ 60,00    | R\$ 300,00    |
| 9  | Insumos     | Display LCD Touchscreen                            | 1          | Quantidade   | R\$ 200,00   | R\$ 200,00    |
| 10 | Insumos     | Kit Touchscreen                                    | 1          | Quantidade   | R\$ 50,00    | R\$ 50,00     |
| 11 | Insumos     | Cabo USB   | 1          | Quantidade   | R\$ 20,00    | R\$ 20,00     |
| 12 | Insumos     | Placa Controladora                                 | 1          | Quantidade   | R\$ 100,00   | R\$ 100,00    |
| 13 | Insumos     | Monitor 15"  | 1          | Quantidade   | R\$ 500,00   | R\$ 500,00    |
| 14 | Insumos     | Frete Fedex  | 1          | Quantidade   | R\$ 600,00   | R\$ 600,00    |
| 15 | Insumos     | Imposto Receita Federal                            | 1          | Quantidade   | R\$ 1.100,00 | R\$ 1.100,00  |
| 16 | Insumos     | Multa Receita Federal                              | 1          | Quantidade   | R\$ 500,00   | R\$ 500,00    |
| 17 | Insumos     | Frete UPS  | 1          | Quantidade   | R\$ 200,00   | R\$ 200,00    |
| 18 | Insumos     | Imposto Receita Federal                            | 1          | Quantidade   | R\$ 350,00   | R\$ 350,00    |
| 19 | Mão de Obra | Coordenador  | 180        | Hora         | R\$ 4,00     | R\$ 720,00    |
| 20 | Mão de Obra | Orientador   | 180        | Hora         | R\$ 16,00    | R\$ 2.880,00  |
| 21 | Mão de Obra | Engenheiro   | 540        | Hora         | R\$ 35,00    | R\$ 18.900,00 |
| 22 | Software    | Pacote Microsoft (Windows, Office, Project, Visio) | 2          | Quantidade   | R\$ 600,00   | R\$ 1.200,00  |
| 23 | Software    | Netbeans   | 1          | Quantidade   | R\$ -        | R\$ -         |
| 24 | Software    | Java Virtual Machine                               | 1          | Quantidade   | R\$ -        | R\$ -         |
| 25 |             |  |            |              |              | R\$ 31.850,00 |

**Tabela 3.1 – Tabela de Custos**

### 3.4 – Cronograma do Projeto

O cronograma previsto de estudo, projeto, implementação, validação e implantação do projeto pode ser observado na figura 3.4.

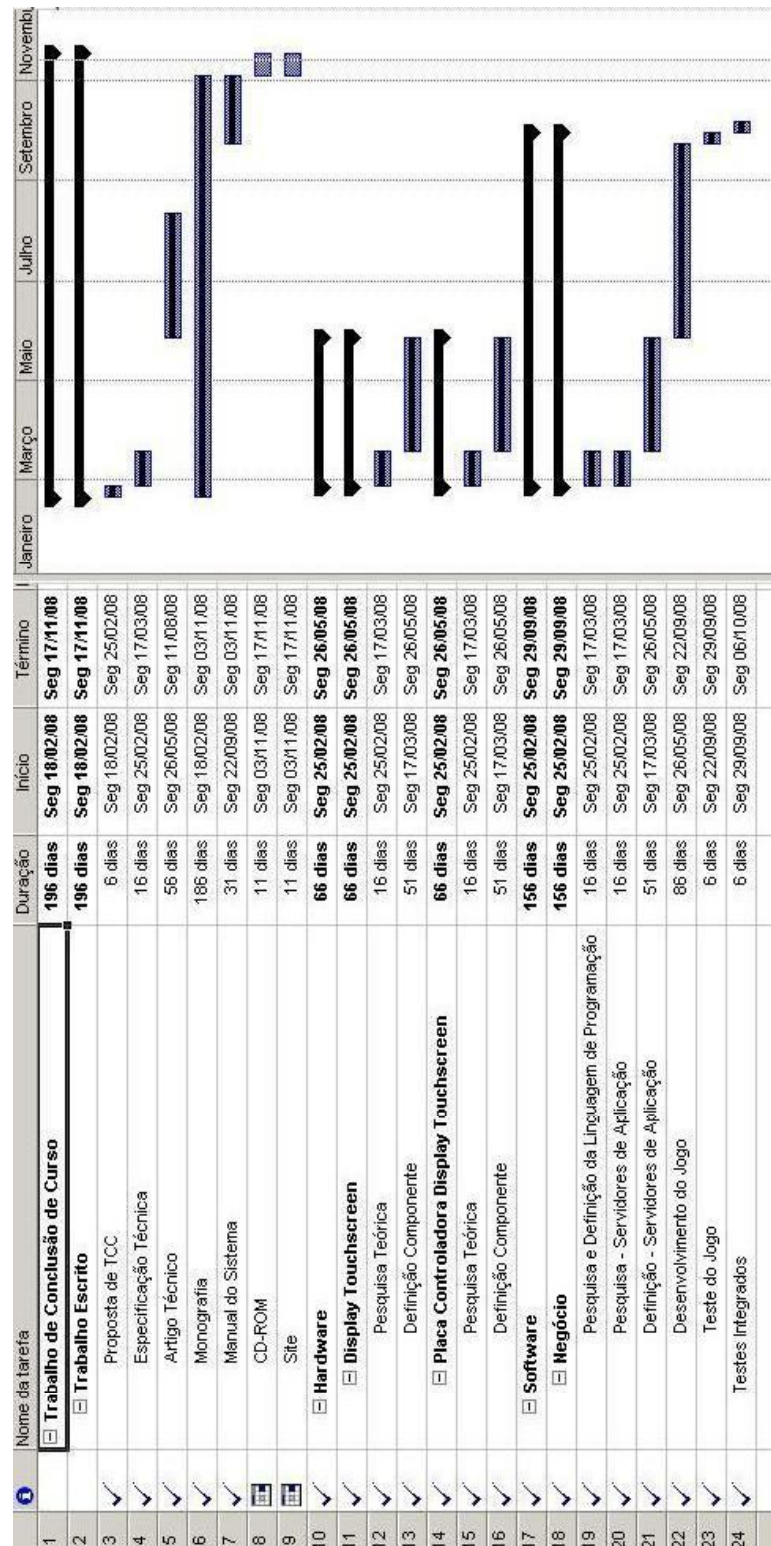


Figura 3.4 – Cronograma do Projeto e Gráfico de Gantt



## CAPÍTULO 4 – DESENVOLVIMENTO E IMPLEMENTAÇÃO

Conforme o capítulo 3, esse projeto foi dividido em duas partes, uma de *hardware* e *software*, detalhadas a seguir.

### 4.1 – Interface *Touchscreen*

Os critérios utilizados para seleção da interface *touchscreen* foram baseados em aplicação, durabilidade, suporte, documentação e custo.

O ambiente previsto para a utilização do protótipo implementado foi o de mostras e/ou feiras científicas, que não apresentam condições extremas de utilização tais como ambientes úmidos, de temperaturas extremas ou insalubres como fábricas ou ambientes externos. Já as condições de uso demandam uma interface com boa durabilidade, pois a proposta de utilização do jogo é justamente ser acionado pela interface *touchscreen*.

O suporte e a documentação representaram fatores fortes na escolha do fabricante, visto que seria necessário interagir com o equipamento de forma a compreender totalmente o seu funcionamento. E já que esse trabalho é de cunho acadêmico, o custo foi outra variável muito importante no processo de decisão do componente.

Pelos motivos acima, optou-se por um *touchscreen* de matriz resistiva, cinco fios, da empresa Bergquist. Sua configuração para uso foi feita conforme a indicação do próprio fabricante, e optou-se pelo uso da placa controladora em configuração USB em função do equipamento utilizado para desenvolvimento e testes não apresentar portas serial ou PS/2.

As especificações elétricas da placa controladora seguem abaixo:

Interface *Touchscreen* – Resistivo, 5 fios

Resolução – 10 bits (1096 x 1096), 1.048.578 pontos

Velocidade de Transmissão – De 40 a 100 pontos por segundo

Freqüência de Clock – 6 MHz

Temperatura de Operação – 0°C até 70°C

Temperatura de Armazenamento – - 40°C até 85°C

Proteção contra estática – 24 kV

Alimentação – 5 VDC  $\pm$ 10%

| Protocolo USB            |       |       |       |       |       |       |       |       |
|--------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Byte                     | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1                        | 0     | 0     | 0     | 0     | 0     | 0     | 0     | P     |
| 2                        | X7    | X6    | X5    | X4    | X3    | X2    | X1    | X0    |
| 3                        | 0     | 0     | 0     | 0     | X11   | X10   | X9    | X8    |
| 4                        | Y7    | Y6    | Y5    | Y4    | Y3    | Y2    | Y1    | Y0    |
| 5                        | 0     | 0     | 0     | 0     | Y11   | Y10   | Y9    | Y8    |
| Protocolos RS-232 e PS/2 |       |       |       |       |       |       |       |       |
| Byte                     | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1                        | 1     | P     | X11   | X10   | X9    | Y11   | Y10   | Y9    |
| 2                        | 0     | X8    | X7    | X6    | X5    | X4    | X3    | X2    |
| 3                        | 0     | Y8    | Y7    | Y6    | Y5    | Y4    | Y3    | Y2    |
| 4                        | 0     | 0     | 0     | 0     | X1    | X1    | Y1    | Y0    |

Tabela 4.1 – Formato de Comunicação da Placa Controladora

A opção escolhida foi pela comunicação USB, sendo para isso necessário configurar os *jumpers* da placa controladora antes de efetuar a conexão entre a película, a placa e o computador. No modo USB, os *jumpers* J1 e J2 correspondem a alimentação da placa e devem estar abertos pois a alimentação é fornecida pela porta USB. Já os *jumpers* J4 e J5 devem estar conectados entre si e o J6 deve estar fechado, ficando os demais abertos. Essa ação configura para a comunicação de dados via porta USB. A figura 4.1 demonstra a configuração.

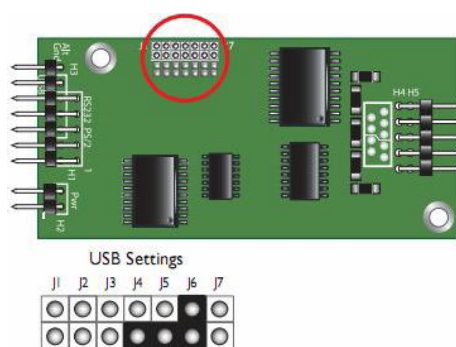
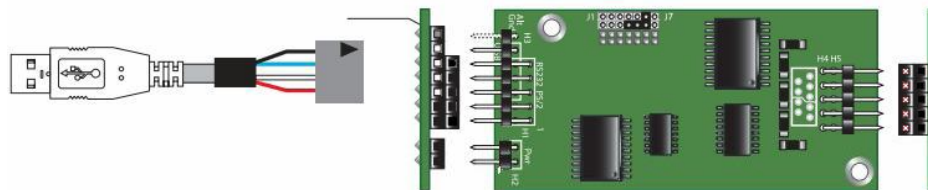


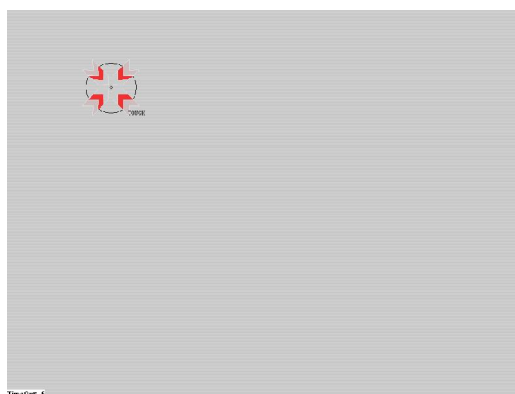
Figura 4.1 – Configuração dos *Jumpers* da Placa Controladora

Em seguida é realizada a conexão do cabo USB e do cabo da película, conforme demonstrado na figura 4.2.

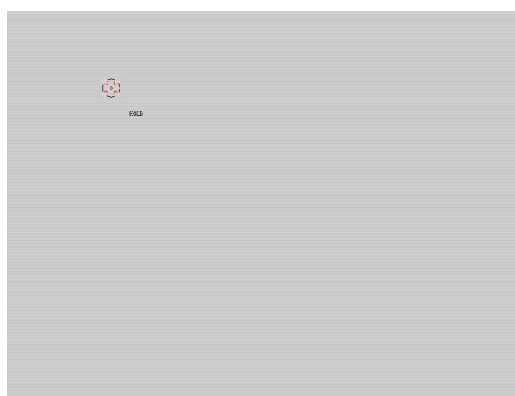


**Figura 4.2 –Conexão do cabo USB e cabo da película**

Após o equipamento estar conectado e seu *driver* instalado, é necessário efetuar a rotina de calibração, demonstrada nas figuras 4.3 ao iniciar a calibração de um ponto e 4.4 ao finalizar a calibração deste mesmo ponto .



**Figura 4.3 – Início do processo de calibração para um ponto**



**Figura 4.4 – Fim do processo de calibração para um ponto**

## 4.2 – Software

O bloco de software foi dividido em duas partes, sendo a primeira responsável pela representação gráfica do jogo e a segunda pela validação de jogadas que é efetuada de forma distribuída. Cabe salientar que justamente em função dessa distribuição, ambas as partes necessitam agregar componentes adicionais para efetuar a conexão entre eles e que é explicado de forma mais detalhada a seguir.

### 4.2.1 – Representação Gráfica

A idéia que motivou o desenvolvimento do projeto foi a implementação de um jogo de xadrez que fosse jogado entre duas pessoas mas a partir de dispositivos diferentes, por exemplo, um dos jogadores poderia jogar a partir de um microcomputador contra outro jogador operando de um celular; ou um jogador operando a partir de um celular contra outro que estivesse utilizando um *palmtop*. Em função do prazo do projeto o escopo foi delimitado para uma interface em um microcomputador.

Contudo, pelo projeto ser idealizado utilizando a linguagem java, cuja principal vantagem da tecnologia é a reutilização de código independente do dispositivo, foi buscado uma solução que permitisse que o mesmo programa utilizado no microcomputador pudesse ser utilizado em um celular ou um *palmtop* por exemplo, foram analisadas as opções que a tecnologia java fornecia.

A tecnologia Java ME, própria para dispositivos portáteis, não apresentava aspectos positivos por não implementar tecnologias como AWT e SWING de forma integral, o que prejudica a compatibilidade entre diferentes dispositivos. Já o Java FX, tecnologia java para aplicações RIA (*Rich Internet Application*), apesar de ser uma tecnologia nova e em versão preview, prevê que não haverá divergências em suas ferramentas para interfaces gráficas para os diversos dispositivos que acessem a aplicação. Somando a isso o fato de ser uma tecnologia inovadora e apoiada por nomes sólidos na indústria da computação como Sun Microsystems e IBM, Java FX foi a escolhida para ser utilizada na construção da interface gráfica. Na figura 4.5 é apresentado o programa principal responsável por construir a interface gráfica, que é visualizada na figura 4.6.

```

1  /*
2  * Main.fx
3  *
4  * Created on 11/09/2008, 20:25:58
5  */
6
7  package client;
8
9  /**
10 * @author Cristiano
11 */
12
13 import javafx.application.Frame;
14
15 var myGame : ChessBoard = ChessBoard();
16
17 Frame {
18     title: "TCC"
19     width: 570
20     height: 590
21     x: 0
22     y: 0
23     closeAction: function() { java.lang.System.exit( 0 ); }
24     visible: true
25
26     stage: myGame.viewBoard();
27 }

```

**Figura 4.5 – Programa principal da interface gráfica**



**Figura 4.6 – Interface gráfica apresentando um jogo em andamento**

Além de construir uma interface que respondesse aos comandos do usuário era necessário efetuar a comunicação com o servidor, e para isso foi necessário construir uma classe de conexão e efetuar sua chamada dentro da interface. Tal classe é mostrada na figura 4.7, onde se deve atentar para a linha 15 que é responsável por efetuar a conexão com o módulo de validação da jogada.

```

1 package client;
2
3 /**
4  *
5  * @author Cristiano
6  */
7 import java.rmi.*;
8 import server.ServidorMenu;
9
10 public class ConexaoRMI {
11
12     public static ServidorMenu getConnection()
13         throws Exception
14     {
15         return (ServidorMenu) Naming.lookup("//localhost:1099/Servidor");
16     }
17
18 }

```

**Figura 4.7 – Classe de conexão com o servidor RMI**

A figura 4.8 demonstra na interface gráfica, após a conexão feita com o servidor na linha 164, como as informações são repassadas para o módulo servidor na linha 165 e o teste de validade da jogada na linha 166.

```

164 var remoteServer : ServidorMenu = ConexaoRMI.getConnection();
165 var validMove : Boolean = remoteServer.movePiece(pieceFrom.nome,pieceTo.nome);
166 if(validMove==true){

```

**Figura 4.8 – Comunicação com o servidor RMI**

#### 4.2.2 – Servidor

Para a validação das jogadas de forma distribuída foi criada uma aplicação servidor que além de armazenar todo o andamento atual do jogo também efetua a validação das jogadas e troca de turnos. Para esse caso foram estudadas as diferentes formas de distribuição de *software* que a tecnologia Java permite.

*Enterprise Java Beans* (EJB) e Java EE foram considerados a princípio e demonstraram ser ótimas soluções, mas foram descartados em função da complexidade e tempo necessários para aprendizado, além de requererem servidores de aplicação para poderem ser executados. Já o RMI demonstrou ser uma boa alternativa em função de sua complexidade menor, necessitar apenas de serviços de rede para ser executado e por já ser ensinado em sala de aula em matérias do curso.

Para lidar com Java RMI é necessário construir uma classe servidora que irá conter toda a regra de negócio necessária e uma classe interface que será responsável por

disponibilizar os métodos necessários do servidor no serviço RMI. A figura 4.9 demonstra a classe interface que é utilizada tanto pelo servidor quanto pela interface gráfica.

```
1 package server;
2
3 import java.rmi.*;
4
5 /**
6  *
7  * @author Cristiano
8  */
9 public interface ChessBoard extends Remote {
10     public void showBoard() throws RemoteException;
11
12     public void movePiece(String from, String to) throws RemoteException;
13 }
```

**Figura 4.9 – Classe Interface**

## CAPÍTULO 5 – VALIDAÇÃO E RESULTADOS

A primeira etapa de validação consistiu na utilização do kit *touchscreen*. Após configuração, revisão, e instalação do kit, seu funcionamento foi testado a partir do software *Microsoft Paint*, aplicativo de desenho que acompanha toda e qualquer instalação dos sistemas operacionais *Microsoft Windows*. Outra validação foi a utilização do kit *touchscreen* em substituição ao mouse na operação do computador. Ambas as etapas foram concluídas com utilização satisfatória. Para ambos os testes, cinco voluntários foram requisitados para efetuá-los.

Durante os testes iniciais foi observado que cada usuário apresentou uma diferente utilização do sistema em função do grau de visão, da pressão, área de contato e tempo de contato do toque na película sensível ao toque. Isso determinou que o sistema deve ser calibrado para cada usuário de forma a apresentar um funcionamento satisfatório e que o tabuleiro deve ter uma medida mínima para atender aos mais diferentes usuários. Para determinar um tamanho satisfatório do tabuleiro foram criados três modelos, sendo o primeiro com quadrados de 50 X 50 pixels, o segundo com quadrados de 75 X 75 pixels e um terceiro com quadrados de 100 X 100 pixels, sendo que a opção intermediária (75 X 75 pixels) foi a que demonstrou melhores resultados para todos os voluntários.

Por fim, a validação mais complexa ficou na conexão entre a interface gráfica e o aplicativo servidor, em função da utilização de uma versão do JDK ainda em versão RC pela utilização do Java FX e também por mudanças na operação de uma distribuição RMI entre essa versão e as versões anteriores. Resolvido isso, o sistema passou a operar perfeitamente.

A figura 5.1 mostra o protótipo montado composto pela película sensível ao toque, a placa controladora e um monitor LCD colocado na horizontal para se valer de maior semelhança com um tabuleiro de xadrez. A figura 5.2 demonstra o protótipo acoplado ao computador utilizado no desenvolvimento do *software* com o servidor iniciado e pronto para receber as requisições da interface.





**Figura 5.1 – Protótipo do tabuleiro**



**Figura 5.2 – Protótipo do Tabuleiro Anexado ao Servidor**

## CAPÍTULO 6 - CONCLUSÃO

A utilização de uma interface *touchscreen* torna o uso de um computador mais natural do que com a utilização de um *mouse*, porém necessita de certas adaptações como tamanho e diferente arranjo dos componentes gráficos de um sistema.

Java FX se mostrou eficiente e com recursos suficientes para construir uma aplicação RIA, restando agora aguardar pela liberação da versão Java FX *Mobile* para constatar se a mesma interface gráfica utilizada em um microcomputador poderá ser utilizada em outro dispositivo sem maiores problemas.

Java RMI demonstrou ser uma tecnologia simples de lidar para construção dos servidores mas confiável, e serviu a contento o propósito de deixar a validação da jogada para ser executada de forma distribuída, cabendo apenas se certificar de que todos os parâmetros necessários para execução do servidor estejam sendo informados.

Outro aspecto positivo foi a demonstração de que a união dessas três tecnologias funciona a contento e sem embaraços.

Como sugestão de trabalhos futuros visando a evolução do projeto, é possível considerar a implementação de um módulo responsável por determinar jogadas e dessa forma uma pessoa poder ter o computador como adversário; a melhoria da interface considerando uma diferenciação da peça selecionada para as outras, o acréscimo de relógios, um para cada jogador, para controlar o tempo do jogo e/ou a indicação de possíveis jogadas a partir da seleção de qual peça será movimentada; aguardar a liberação do Java FX *Mobile* para utilizar a interface em outro dispositivo e realizar um jogo entre duas pessoas em dispositivos diferentes; e considerar o desenvolvimento desse jogo de forma embarcada.

Partindo para outros projetos que possam ser criados utilizando os conceitos aqui estudados, pode-se considerar que o desenvolvimento orientado a objetos fez com que fosse criada uma infra-estrutura que pode ser evoluída para outros jogos de tabuleiro, além de ser possível portar o próprio jogo de xadrez ou outro que possa ser criado para dispositivos portáteis.

Já na parte do kit *touchscreen*, fica a sugestão de criar outros aplicativos que possam ser operados a partir dele ou mesmo a adaptação de um sistema operacional para ser utilizado em sua maior parte pelo *touchscreen*, substituindo assim o *mouse*. Uma boa inspiração para isso seria o *Microsoft Surface*, equipamento utilizado como se fosse uma mesa normal.

## CAPÍTULO 7 - REFERÊNCIAS

The Bergquist Company. **Touch Screen**.

<<http://www.bergquistcompany.com/objects%2FTouch%20Screen%20Website%2FTouchscreenhome.cfm>> - Acessado em março de 2008.

SOMMERVILLE, Ian. **Engenharia de Software**. 8<sup>a</sup> ed. São Paulo: Pearson Addison-Wesley, 2007.

Deitel, H. M.; Deitel, P. J. **Java: Como Programar**. 3<sup>a</sup> ed. Porto Alegre: Bookman, 2001.

Filho, Hildeberto L. **Interface Touchscreen**. Monografia. Curitiba: Universidade Positivo. 2007.

Elotouch, Touch-Screen e Controladora. **Descrição do funcionamento e descrição tecnologia Touch-Screen**. <<http://www.elotouch.com.br>>. Acesso em março de 2008.

Sun Microsystems. **Enterprise Java Beans Technology, Specification 3.0**.

<<http://java.sun.com/products/ejb/>>. Acesso em março de 2008

Sun Microsystems. **Java Platform, Enterprise Edition 5 (Java EE 5)**.

<<http://java.sun.com/javaee/technologies/javaee5.jsp>>. Acesso em março de 2008

Sun Microsystems. **Java FX**. < <http://java.sun.com/javafx/>>. Acesso em março de 2008

Sun Microsystems. **Java RMI**.

<<http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp>>. Acesso em março de 2008

François Dominic Lamarée. **Chess Programming**.

< <http://www.gamedev.net/reference/articles/article1014.asp>>. Acesso em fevereiro de 2008

Anselmo, Fernando. **Tudo sobre a JSP com o NetBeans em Aplicações Distribuídas**.

Florianópolis: Visual Books, 2005.

Santee, André. **Programação de Jogos com C++ e DirectX**. São Paulo: Novatec, 2005.

Perucia, Alexandre S.; Berthêm, Antônio C.; Bertschinger, Guilherme L.; Menezes, Roberto R. C.. **Desenvolvimento de Jogos Eletrônicos: Teoria e Prática**. São Paulo: Novatec, 2005.