

**Universidade de São Paulo**  
**Escola de Engenharia de São Carlos**  
**Departamento de Engenharia Elétrica e**  
**Computação**

**DESENVOLVIMENTO DE UMA**  
**PLATAFORMA PARA UM JOGO DE**  
**XADREZ INTELIGENTE CONTROLADO**  
**POR RECONHECIMENTO DE VOZ**

**Aluno:**

Pablo Henrique Alves de Oliveira (nºUSP:6920435)

**Orientador:**

Evandro L. L. Rodrigues

São Carlos

Novembro de 2014



**Pablo Henrique Alves de Oliveira**

**DESENVOLVIMENTO DE UMA  
PLATAFORMA PARA UM JOGO DE  
XADREZ INTELIGENTE CONTROLADO  
POR RECONHECIMENTO DE VOZ**

Trabalho de Conclusão de Curso  
Apresentado à Escola de Engenharia de São Carlos  
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em Eletrônica  
ORIENTADOR: Evandro Luis Linhari Rodrigues

São Carlos  
Novembro de 2014

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

048d Oliveira, Pablo Henrique Alves de  
Desenvolvimento de uma plataforma para um jogo de xadrez inteligente controlado por reconhecimento de voz / Pablo Henrique Alves de Oliveira; orientador Evandro Luis Linhari Rodrigues. São Carlos, 2014.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2014.

1. Reconhecimento de Voz. 2. Webspeech API. 3. Inteligência Artificial. 4. Xadrez. 5. JavaScript. 6. HTML. I. Título.

# FOLHA DE APROVAÇÃO

Nome: Pablo Henrique Alves de Oliveira

Título: "Desenvolvimento de uma plataforma para um jogo de xadrez controlado por reconhecimento de voz"

Trabalho de Conclusão de Curso defendido e aprovado  
em 25/11/2014,

com NOTA 9,5 (NOVE, CINCO), pela Comissão Julgadora:

*Prof. Associado Evandro Luís Linhari Rodrigues - (Orientador - SEL/EESC/USP)*

*Prof. Associado Ivan Nunes da Silva - (SEL/EESC/USP)*

*Prof. Dr. Dennis Brandão - (SEL/EESC/USP)*

Coordenador da CoC-Engenharia Elétrica - EESC/USP:  
Prof. Associado Homero Schiabel

## **Dedicatória**

Dedico este trabalho acima de tudo aos meus pais, Patrícia e Vassil, e aos meus irmãos, Lucas e Gabriel, que sempre me apoiaram e me ajudaram nos momentos difíceis, e sem os quais eu nunca teria conseguido chegar aonde eu cheguei.



## **Agradecimentos**

Agradeço primeiramente aos meus pais, que sempre confiaram em mim e fizeram de tudo para me proporcionar a oportunidade de estudar na melhor universidade do País. Seus conselhos, sua paciência e seu amor foram imprescindíveis para que eu conseguisse ir atrás de atingir meus sonhos.

Agradeço também aos meus irmãos e familiares por terem sempre me apoiado, mesmo à distância; e aos meus amigos, que fizeram com que meus anos de faculdade fossem os melhores anos de minha vida.

Por fim, agradeço ao professor Evandro, que sempre me incentivou crescer e a buscar experiências que me fizessem evoluir não só como aluno, mas também como pessoa.





## Resumo

O objetivo deste projeto é implementar uma plataforma virtual para a realização de jogos de xadrez contra o computador com os movimentos do usuário controlados por reconhecimento de voz e os do computador determinados por inteligência artificial. Esta plataforma será programada em HTML, CSS e JavaScript, e terá como principal função a interpretação como movimentos de xadrez dos comandos de voz recebidos via Webspeech API, uma interface web que realiza o reconhecimento da fala e a interpretação das palavras ditas pelo usuário. Uma vez interpretado o lance, o computador deve realizá-lo e em seguida determinar qual o melhor movimento em resposta.

**Palavras-chave:** Reconhecimento de voz, xadrez, HTML, JavaScript, Webspeech API, inteligência artificial



## **Abstract**

The goal of the project is to implement a virtual user vs. computer chess platform, in which the user moves will be controlled by a voice recognition system and the computer moves will be determined by artificial intelligence. This platform will be programmed in HTML, CSS and JavaScript, and will have as main role the interpretation as chess movements of the voice commands received via Webspeech API, a web interface that does the voice recognition and interprets the words said by the user. Once the move is interpreted, the computer must perform it and then determine the best move to use as a response.

**Keywords:** Voice recognition, chess, HTML, JavaScript, PHP, Webspeech API, artificial intelligence



## Lista de figuras

Figura 1 – Configuração inicial do Jogo de Xadrez .....	20
Figura 2 – Movimento do Rei, retirado do site <a href="http://baquara.com/xadrez/movimentos.htm">http://baquara.com/xadrez/movimentos.htm</a> .....	22
Figura 3 – Movimento da Dama, retirado do site <a href="http://baquara.com/xadrez/movimentos.htm">http://baquara.com/xadrez/movimentos.htm</a> .....	23
Figura 4 – Movimento da Torre, retirado do site <a href="http://baquara.com/xadrez/movimentos.htm">http://baquara.com/xadrez/movimentos.htm</a> .....	23
Figura 5 – Movimento do Bispo, retirado do site <a href="http://baquara.com/xadrez/movimentos.htm">http://baquara.com/xadrez/movimentos.htm</a> .....	24
Figura 6 – Movimento do Cavalo, retirado do site <a href="http://baquara.com/xadrez/movimentos.htm">http://baquara.com/xadrez/movimentos.htm</a> .....	24
Figura 7 – Movimento do Peão, retirado do site <a href="http://baquara.com/xadrez/movimentos.htm">http://baquara.com/xadrez/movimentos.htm</a> .....	25
Figura 8 – Página de Demonstração da Webspeech API.....	30
Figura 9 – Diferentes formas de se definir IA (Russell e Norvig, 1995, p.2).....	31
Figura 10 – Configuração da Árvore de Movimentos Possíveis.....	33
Figura 11 – Algoritmo Minimax.....	34
Figura 12 – Algoritmo Alfa-Beta .....	36
Figura 13 – Nível Zero do DFD do Software <i>Xadrez por Reconhecimento de Voz</i> ....	38
Figura 14 – Imagens utilizadas como peças, retirado do site <a href="http://jpe.gs/wiki/Image:Chess_set_symbols_igor_k_01.svg">http://jpe.gs/wiki/Image:Chess_set_symbols_igor_k_01.svg</a> .....	40
Figura 15 – Página web com a configuração inicial do tabuleiro de xadrez .....	40
Figura 16 – Nível 1 do DFD do Software <i>Xadrez por Reconhecimento de Voz</i> .....	41
Figura 17 – Nível 2 do DFD do Software <i>Xadrez por Reconhecimento de Voz</i> (Checar Movimento).....	42
Figura 18 – Processo de Reconhecimento de Voz: início .....	45
Figura 19 – Processo de Reconhecimento de Voz: captura da fala.....	45
Figura 20 – Processo de Reconhecimento de Voz: Realização do movimento.....	46
Figura 21 – Pontuação das peças de Xadrez.....	48
Figura 22 – Tela inicial da página <i>Xadrez por Reconhecimento de Voz</i> .....	51
Figura 23 – Tela de jogo da página <i>Xadrez por Reconhecimento de Voz</i> .....	51

## **Lista de abreviaturas e siglas**

*API*            *Application Programming Interface*

*HTML*          *HyperText Markup Language*

*CSS*            *Cascading Style Sheets*

*RV*             *Reconhecimento de Voz*

*IA*             *Inteligência Artificial*

*DFD*           *Diagrama de Fluxo de Dados*

1.	Introdução.....	17
2.	Objetivos e Estrutura da Monografia.....	19
3.	Conceitos Teóricos .....	20
3.1.	O Jogo de Xadrez.....	20
3.1.1.	Termos enxadrísticos.....	21
3.1.2.	As peças .....	21
3.1.3.	Movimentos especiais.....	25
3.1.4.	Dinâmica do jogo .....	26
3.2.	HTML e CSS .....	26
3.3.	JavaScript .....	26
3.3.1.	Variáveis .....	27
3.3.2.	Operadores.....	28
3.3.3.	Estruturas de Controle.....	28
3.3.4.	Funções .....	29
3.3.5.	Orientação a Objeto .....	29
3.4.	Reconhecimento de fala.....	29
3.4.1.	Webspeech API .....	30
3.5.	Inteligência Artificial.....	31
3.5.1.	Inteligência Artificial e o jogo de Xadrez .....	31
4.	Implementação e Desenvolvimento do Projeto .....	38
4.1.	Desenvolvimento do jogo de xadrez em HTML, CSS e JavaScript .....	39
4.1.1.	Inicialização do tabuleiro e posicionamento das peças .....	39
4.1.2.	Implementação das regras e dinâmica do jogo do Xadrez.....	41
4.2.	Implementação Reconhecimento de Voz.....	43
4.2.1.	Webspeech API .....	43
4.2.2.	Hospedagem da Página Web no Dropbox .....	46
4.2.3.	Speech Synthesis API .....	47
4.3.	Implementação da Inteligência Artificial.....	47
4.3.1.	Avaliação de Posições.....	47
4.3.2.	Algoritmo Negamax.....	48
4.3.3.	Algoritmo Alfa-Beta .....	50
5.	Resultados e Análise.....	51
6.	Considerações Finais .....	55



6.1. Trabalhos Futuros .....	56
7. Referências .....	57
APÊNDICE A – index.html .....	17
APÊNDICE B – estilo.css .....	61
APÊNDICE C – reconhecerVoz.js .....	62
APÊNDICE D – iniciarTabuleiro.js .....	63
APÊNDICE E – movimentarPeca.js .....	0
APÊNDICE F – aplicarIA.js .....	0

## 1. Introdução

Tecnologias de reconhecimento de fala, ou reconhecimento de voz, permitem que computadores recebam informações a partir de microfones e interpretem a fala humana para a realização de determinadas ações a partir de comandos de voz. Estes sistemas podem ser treinados para reconhecer um padrão específico de fala, e podem ser projetados para reconhecer fala contínua ou para exigir que o usuário fale pausadamente.

Esses conceitos têm sido constantemente estudados, porém ainda oferecem algumas dificuldades aos seus desenvolvedores. O principal problema está na diferença das características da fala de diferentes interlocutores, o que pode fazer com que um mesmo sistema consiga interpretar o comando dado por uma pessoa, mas não consiga interpretar o mesmo comando dado por outra pessoa. Outro problema está nas diferentes entonações utilizadas durante a fala, que na linguagem escrita são traduzidas para acentuações e pontuações. Para o ouvido humano, essas diferenças são facilmente percebidas, mas já se tornam mais difíceis de serem ensinadas a uma máquina. Um terceiro problema está na ambiguidade de frases e palavras, que podem dificultar o trabalho da máquina de interpretação do sentido do que foi dito. O desafio se encontra não apenas no recebimento e reconhecimento das palavras, mas também na interpretação de sentenças e na criação de lógicas que permitem a utilização desta informação em alguma aplicação.

Apesar das dificuldades, hoje em dia é cada vez mais comum encontrar aplicações para esta tecnologia. Um dos exemplos mais conhecido é a *Siri* nos aparelhos da Apple, que realiza buscas no Google e no celular a partir de comandos de voz. Outra aplicação que vem sendo bastante usada é utilização desta tecnologia no setor automobilístico, permitindo que usuários realizem certas ações enquanto dirigem sem a necessidade de tirar as mãos do volante. Este projeto visa mostrar uma aplicação simples desta tecnologia na indústria de jogos permitindo que movimentos de xadrez sejam realizados sem a necessidade de um tabuleiro físico.

Outro conceito bastante ligado à indústria de jogos é o de inteligência artificial, que tem como objetivo ensinar máquinas a tomar decisões e realizar ações por conta própria. Em jogos, personagens que não são controlados por seres humanos

são constantes, e possuem na maioria das vezes um papel fundamental na dinâmica e na determinação da dificuldade destes.

A escolha do jogo de xadrez foi feita por este possuir uma estrutura e uma dinâmica muito bem definidas, com uma notação simples para seus movimentos, havendo a possibilidade de deficientes visuais jogarem partidas contra computadores ou on-line caso conheçam as regras da notação e tenham a capacidade de memorizar o tabuleiro, ou então pessoas com tetraplegia, uma vez que os jogos poderiam ser jogados apenas pela fala.

O software que realizará de fato o jogo de xadrez será programado em HTML, uma linguagem usada para a produção de textos para web, em conjunto com CSS, linguagem de formatação de textos escritos em HTML, JavaScript, utilizado na implementação da ferramenta de reconhecimento de voz e para a lógica de funcionamento do programa. Já para a captura dos comandos de voz será utilizada uma interface web chamada Webspeech API, que deverá sofrer algumas modificações para se adequar a este projeto. Será implementada também uma IA que ficará responsável pela análise dos movimentos do usuário e a determinação de um movimento em resposta, propiciando certa dinâmica no jogo. O projeto será então hospedado pelo site dropbox.com, que aparece como uma alternativa simples para hospedagem de páginas estáticas. Todas essas plataformas utilizadas foram escolhidas por apresentarem uma ampla documentação e facilidade na busca de informações.

## 2. Objetivos e Estrutura da Monografia

Como objetivos deste projeto pode-se salientar:

- i. Implementação de uma plataforma que permita a realização de Jogos de xadrez
- ii. Aplicação de conceitos e ferramentas de reconhecimento de voz
- iii. Implementação de uma inteligência artificial para fazer o papel de adversário do usuário

A monografia divide-se em 5 partes:

- Parte 1 – Introdução do tema tratado e definição dos objetivos;
- Parte 2 – Descrição de conceitos técnicos;

Parte 3 – Implementação e desenvolvimento:

- Parte 4 – Discussão e análise dos resultados;
- Parte 5 – Considerações finais.

### 3. Conceitos Teóricos

#### 3.1. O Jogo de Xadrez

O xadrez é um jogo de tabuleiro antigo, originado a partir de outro jogo, chamado Chaturanga. A partir do século XV o xadrez começou a tomar uma forma parecida com a atual, sendo nesse período criadas as principais regras vistas hoje em dia.

Um kit utilizado para uma partida de xadrez é constituído basicamente por um tabuleiro quadriculado 8x8, 16 peças brancas e 16 peças pretas. As 16 peças, de ambas as cores, são compostas por 8 peões, 2 torres, 2 cavalos, 2 bispos, 1 rei e 1 dama, sendo inicialmente dispostas como na figura 1.

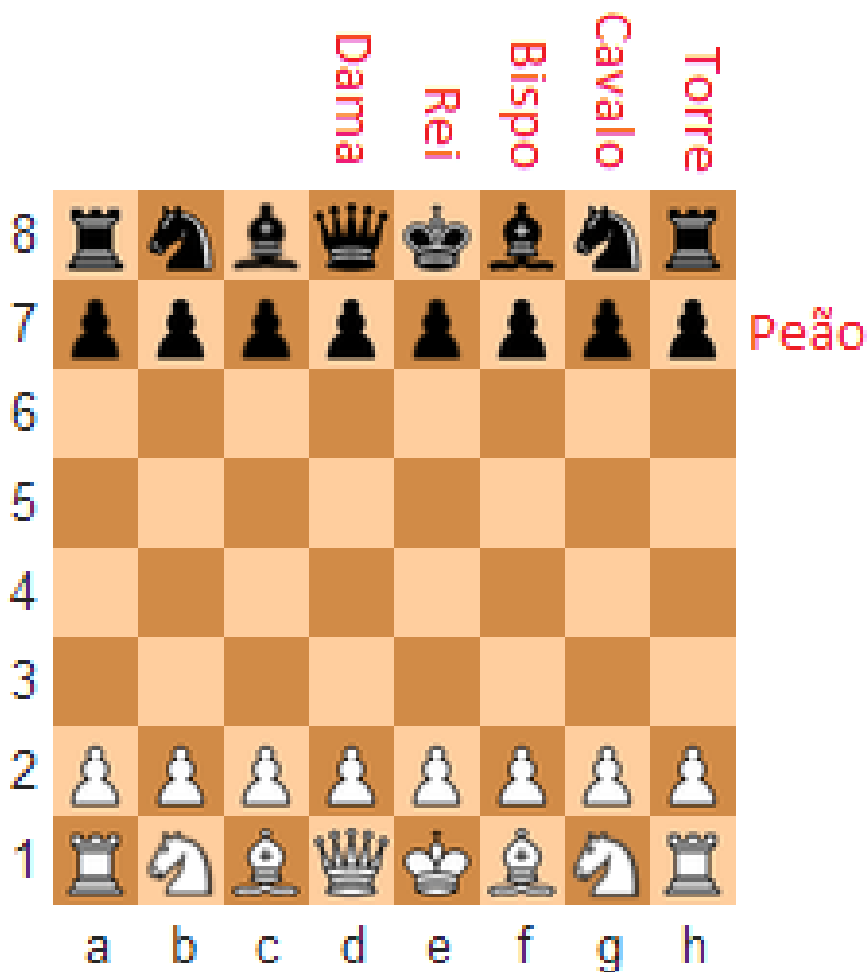


Figura 1 – Configuração inicial do Jogo de Xadrez

### 3.1.1. Termos enxadrísticos

Antes de se tentar entender a dinâmica do jogo de xadrez é necessário ter conhecimento de alguns termos:

- Casa do tabuleiro: cada quadrado do tabuleiro corresponde a uma casa. As letras de “a” a “h” e os números de “1” a “8” vistos na figura 1 mapeiam o tabuleiro em linhas e colunas, permitindo uma fácil especificação de jogadas realizadas.
- Movimento: o movimento de uma peça ocorre quando a movemos de uma casa para outra em que não haja outra peça da mesma cor. Peças diferentes possuem regras diferentes de movimentação, as quais serão explicadas mais à frente.
- Captura: a captura de uma peça ocorre quando ao fim de um movimento de uma determinada peça A passa a ocupar o lugar de uma peça B do adversário. Diz-se então que a peça A capturou a peça B, ou que a peça B foi capturada pela peça A. Na maioria dos casos o movimento de captura é o mesmo movimento normal da peça, havendo exceção apenas no caso do peão.
- Ataque: diz-se que uma peça A está atacando uma determinada casa do tabuleiro quando ela pode se movimentar para esta casa realizando alguma captura.
- Defesa: a defesa de uma peça ocorre quando o jogador realiza algum movimento que interrompa algum ataque àquela peça.
- Xeque: um xeque acontece quando o rei sofre um ataque.
- Xeque-Mate: ocorre quando um xeque não é possível de ser defendido, levando então à derrota por parte do jogador com o rei atacado.

### 3.1.2. As peças

Como dito anteriormente, cada peça possui um movimento diferente, e para cada uma atribui-se também um valor simbólico. Apesar de ser bastante utilizado para se avaliar jogadas e posições, esse valor não influencia no resultado do jogo.

- Rei: é a peça mais importante do jogo. Caso um dos jogadores tenha seu rei atacado por alguma peça adversária, este jogador é obrigado a protegê-lo, podendo fazer isso de 3 formas: movendo o rei para uma casa não atacada, colocando uma peça entre o rei e a peça atacante, ou capturando a peça atacante. Movimenta-se no máximo uma casa para qualquer direção, contanto que a sua casa de destino não esteja sendo atacada por outra peça, como é o caso das casas g1 e g2 na figura 2. O rei não recebe pontuação, uma vez que sua captura, ou um xeque-mate, leva à derrota do jogador.

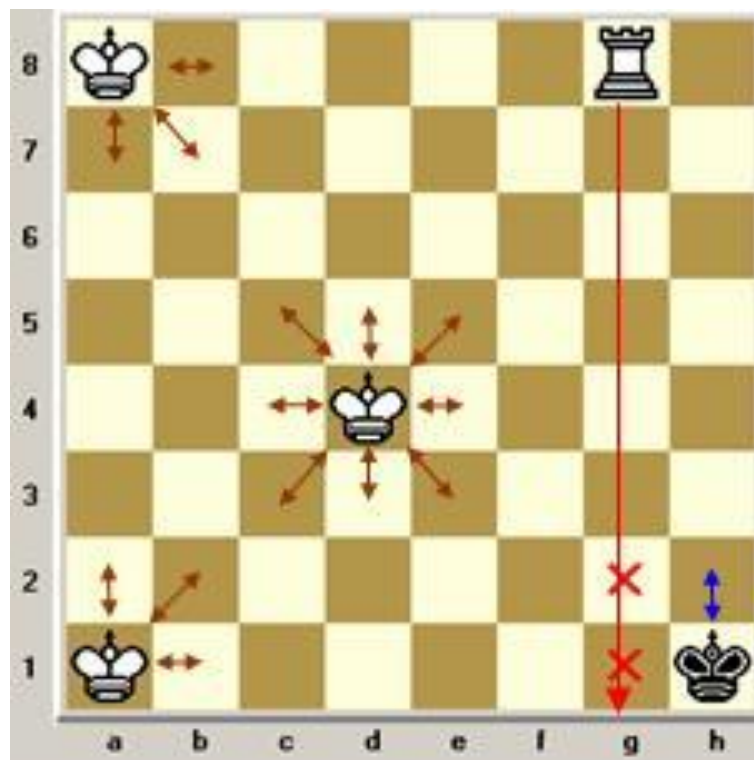


Figura 2 – Movimento do Rei, retirado do site <http://baquara.com/xadrez/movimentos.htm>

- Dama: também conhecida como rainha, é a peça mais poderosa do jogo, podendo se movimentar para qualquer direção quantas casas desejar, contanto que não haja obstrução no caminho por parte de outras peças. Vale 9 pontos.

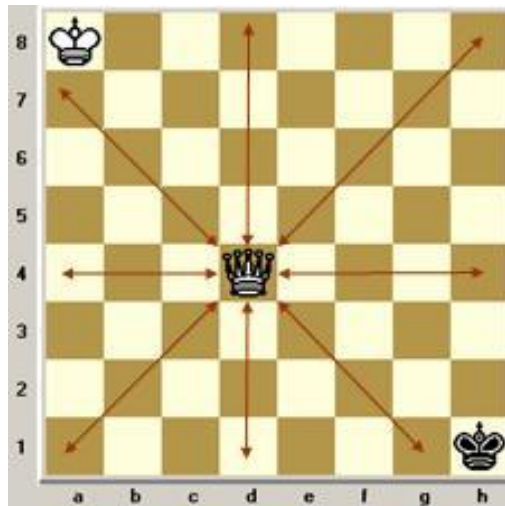


Figura 3 – Movimento da Dama, retirado do site <http://baquara.com/xadrez/movimentos.htm>

- Torre: movimenta-se na vertical ou na horizontal quantas casas desejar, contanto que não haja obstrução no caminho por parte de outras peças. Vale 5 pontos.

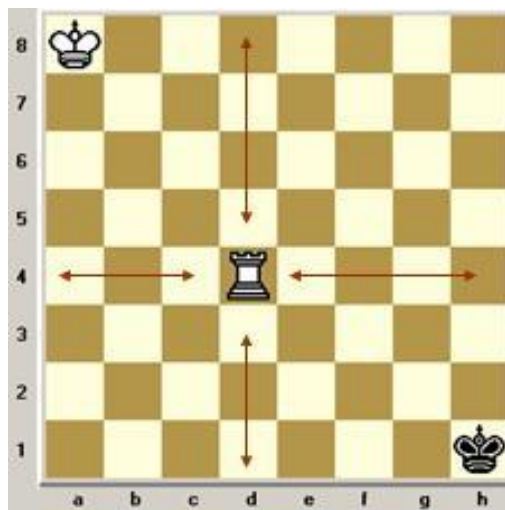


Figura 4 – Movimento da Torre, retirado do site <http://baquara.com/xadrez/movimentos.htm>

- Bispo: movimenta-se na diagonal quantas casas desejar, contanto que não haja obstrução no caminho por parte de outras peças. Cada jogador possui um bispo nas casas claras e outro na escuras. Vale 3 pontos.



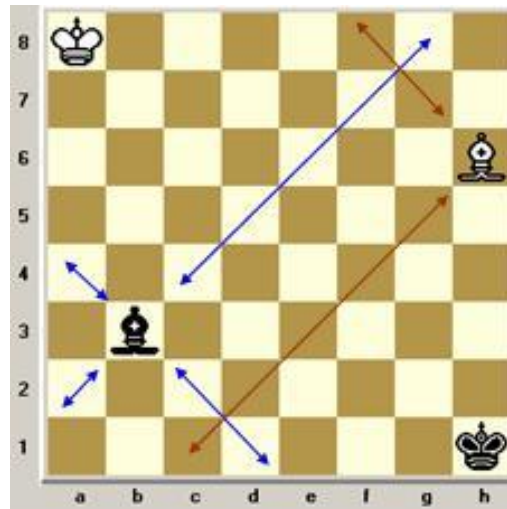


Figura 5 – Movimento do Bispo, retirado do site <http://baquara.com/xadrez/movimentos.htm>

- Cavalo: movimenta-se em forma de um L, sendo 1 casa na horizontal ou vertical e depois 2 casas na vertical ou horizontal, respectivamente. O cavalo é a única peça que pode saltar sobre as outras para realizar seu movimento. Vale 3 pontos.

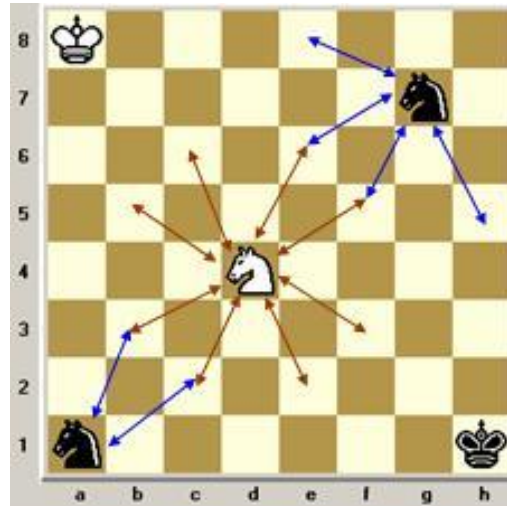


Figura 6 – Movimento do Cavalo, retirado do site <http://baquara.com/xadrez/movimentos.htm>

- Peão: é a peça mais fraca do jogo, podendo se movimentar apenas uma casa e direção ao lado adversário, porém com umas peculiaridades. Caso seja seu primeiro movimento, o jogador pode movê-lo duas casas, em vez de apenas uma. Diferente das outras peças, seu movimento de captura e seu movimento real são diferentes, permitindo que capture peças apenas nas diagonais logo à sua frente. Vale 1 ponto.

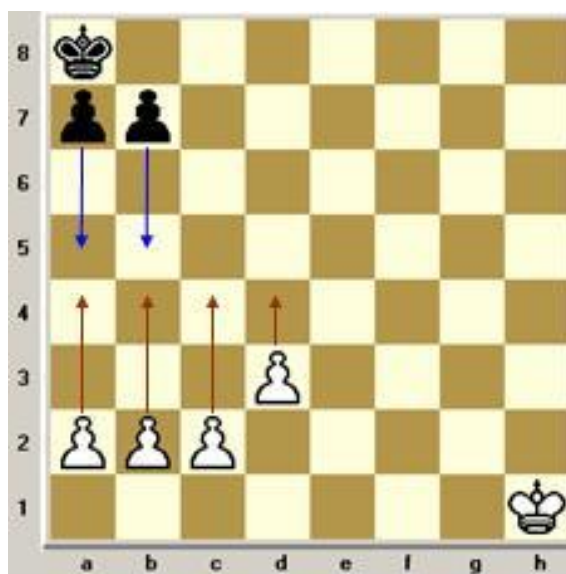


Figura 7 – Movimento do Peão, retirado do site <http://baquara.com/xadrez/movimentos.htm>

### 3.1.3. Movimentos especiais

Além dos movimentos citados, existem outros três movimentos especiais:

- **Roque:** é o único movimento do jogo em que duas peças são movidas simultaneamente. O rei se move 2 casas em direção a uma das torres, e em seguida esta torre salta sobre o rei passando a ocupar a casa a seu lado. Para que o roque aconteça algumas regras devem ser seguidas: nenhuma das duas peças pode ter se movimentado anteriormente; o rei não pode estar em xeque, terminar o movimento em xeque ou passar por uma casa atacada por uma peça adversária; não pode haver peças entre o rei e a torre. O roque possui o objetivo de proteção do rei.
- **Promoção:** acontece quando um peão atinge a última linha do lado adversário. Neste caso, o jogador pode escolher transformar o peão em qualquer outra peça do jogo, com exceção do rei, adquirindo suas características de movimento.
- **En Passant:** é uma captura especial de peões realizada apenas por peões brancos na quinta linha do tabuleiro ou peões pretos na quarta linha do tabuleiro. Trata-se de um movimento bastante raro.

### 3.1.4. Dinâmica do jogo

A dinâmica do jogo é bastante simples: dois jogadores, um com as peças brancas e outro com as pretas, realizam seus movimentos alternadamente (com o primeiro movimento sendo das peças brancas) com o objetivo de se dar um Xeque-Mate no rei adversário.

## 3.2. HTML e CSS

Criada pelo físico britânico Tim Berners-Lee, a linguagem HTML é uma linguagem de marcação, e é utilizado para a criação de páginas web. Já a linguagem CSS é uma linguagem de folha de estilo, utilizada para se determinar o formato que os documentos escritos em linguagens de marcação, como HTML e XML, serão apresentados. O uso destas duas linguagens proporciona uma separação entre o conteúdo do texto e sua formatação. Ambas as linguagens podem ser escritas em qualquer editor de texto, como um bloco de notas, havendo a necessidade de serem salvas como .html ou .css para poderem ser então interpretadas pelos navegadores.

## 3.3. JavaScript

A linguagem de programação JavaScript foi desenvolvida nos anos 90 pelo programador Brendan Elch, da empresa Netscape Communications. Possui uma sintaxe bastante parecida com as linguagens de programação C e Java, porém, apesar desta semelhança tanto em sua sintaxe quanto no nome, não é relacionada de forma alguma com elas, e nem foi originada da linguagem Java. É hoje uma das principais linguagens para programação web, principalmente em aplicações *client-side*, ou seja, sem a necessidade de troca de informações com o servidor.

O sufixo “-Script” indica que se trata de uma linguagem de script, isto é, pode ser executada dentro de outros programas com o intuito de estender a funcionalidade destes. Com a linguagem JavaScript é possível controlar, criar, alterar e excluir elementos em páginas HTML.

A principal vantagem de uma linguagem de script com relação a uma linguagem compilada está no processo de escrita do código e em sua execução. Na

linguagem compilada deve-se escrever o código-fonte, que será traduzido para linguagem de máquina pelo compilador, gerando um código executável, que por sua vez será executado no computador. Esse código executável, por ser escrito em linguagem de máquina, é praticamente impossível de ser alterado. Isso faz com que sempre que se quiser realizar alterações no programa deve-se primeiro alterar o código-fonte e passar pelo processo de compilação novamente.

Uma linguagem de Script, por outro lado, facilita a realização de tais alterações, uma vez que um interpretador previamente instalado realiza o trabalho de leitura e execução do código-fonte a partir de seu próprio texto, ou seja, caso seja necessário fazer alguma alteração, basta alterar o código-fonte e executar o programa, sem necessidade de um procedimento de compilação, gerando-se certa agilidade.

Para que esse processo funcione é necessário que o interpretador da linguagem em questão esteja devidamente instalado na máquina e que o usuário possua algum ambiente para se escrever o código-fonte. No caso da linguagem JavaScript, praticamente todos os computadores pessoais no mundo possuem pelo menos um interpretador desta linguagem, o que faz com que programas escritos em JavaScript sejam de fácil acesso a qualquer pessoa. Códigos em JavaScript podem ser escritos em qualquer bloco de notas, também presentes na grande maioria dos computadores pessoais, devendo ser salvos no formato “.js”.

Assim como grande parte das linguagens de programação, a linguagem JavaScript possui variáveis de diferentes tipos, operadores, estruturas de controle e funções; além disso, possibilita a criação e utilização de objetos.

### **3.3.1. Variáveis**

A linguagem JavaScript é uma linguagem de “tipagem dinâmica”, isto é, as variáveis podem ou não ser criadas com um tipo específico, podendo sofrer alterações no decorrer do código. Os tipos que as variáveis podem assumir são:

- Numérico;
- Lógico ou booleano (verdadeiro ou falso);
- Strings, ou sequências de caracteres;
- Arrays (conhecidos como vetores em outras linguagens);

- Valor nulo;
- Valor indefinido.

### **3.3.2. Operadores**

Os operadores podem ser:

- Aritméticos – adição, subtração, multiplicação, divisão;
- De comparação – igual, diferente, maior, menor;
- Lógicos – e lógico, ou lógico, negação lógica;
- Atribuição de valores;
- Bit a Bit – and, or, xor, not.

### **3.3.3. Estruturas de Controle**

Possui tanto estruturas de testes quanto laços, como:

- if / else – testa se uma condição é verdadeira e realiza uma série de comandos, caso contrário realiza outra série de comandos.
- switch / case / break – verifica o valor de uma variável e executa determinados comandos dependendo do valor encontrado, utilizando o comando break para que o programa pare de verificar as outras possibilidades, uma vez que a certa já foi determinada.
- while – realiza uma série de comandos enquanto uma condição for verdadeira.
- do ... while – possui a mesma função do while, com a diferença de que neste caso a condição passa a ser monitorada depois da primeira iteração.
- for – realiza uma série de comandos frente a uma quantidade de vezes pré-determinada.

### **3.3.4. Funções**

Funções têm como intuito simplificar o código, fazendo com que pedaços de códigos que seriam repetidos diversas vezes sejam escritos apenas uma vez. A criação de funções permite a divisão do programa em partes menores, facilitando na hora da programação e na procura por erros.

### **3.3.5. Orientação a Objeto**

A linguagem JavaScript permite a criação de objetos, que, diferentes de simples variáveis, podem possuir diversos valores armazenados neles, chamados de atributos, e também funções próprias, conhecidas como métodos. Atributos e métodos são chamados de propriedades do objeto. Vários objetos já foram programados anteriormente e estão disponíveis para a utilização dos programadores, como é o caso de Strings e Arrays, que são suportados por diferentes tipos de browsers.

## **3.4. Reconhecimento de fala**

De maneira simplificada, tecnologias de reconhecimento de voz permitem que seres humanos possam se comunicar com máquinas por meio da fala. A fala é então interpretada pela máquina, que em seguida realiza uma série de ações.

As dificuldades encontradas no desenvolvimento de tais tecnologias se devem ao fato de que cada pessoa possui uma maneira própria de falar. As diferenças podem ser vistas no timbre da voz, na velocidade da fala, no sotaque, na linguagem. Porém, apesar das dificuldades, de um modo geral as tecnologias de reconhecimento de voz vêm se mostrando cada vez mais eficientes e presentes no mundo atual.

Um exemplo de aplicação é o aplicativo Siri, encontrado em celulares com o sistema operacional iOS, sendo utilizado para ajudar o usuário a acessar diversas funções do aparelho apenas pelo comando de voz. Outro exemplo cada vez mais comum é o de atendentes eletrônicas em empresas de serviços, como operadoras de telefone e de internet, que te guiam por uma série de menus de opções que o

usuário passa a escolher por comando de voz, em vez do tradicional método de clicar em determinadas teclas do teclado numérico.

### 3.4.1. Webspeech API

Uma API – Interface de Programação de Aplicações – funciona como uma espécie de biblioteca de funções, que podem ser utilizadas de maneira pronta por programadores sem a necessidade de uma pesquisa avançada sobre como funcionam internamente.

A Webspeech API, desenvolvida na linguagem JavaScript e implementada no Chrome 25 e superior, fornece a programadores Web ferramentas que possibilitam a interação do usuário com a máquina e vice-versa por meio de recursos de fala. Os recursos que a Webspeech API fornece no browser não são disponíveis usando reconhecimento de voz padrão ou softwares de leitura de tela.

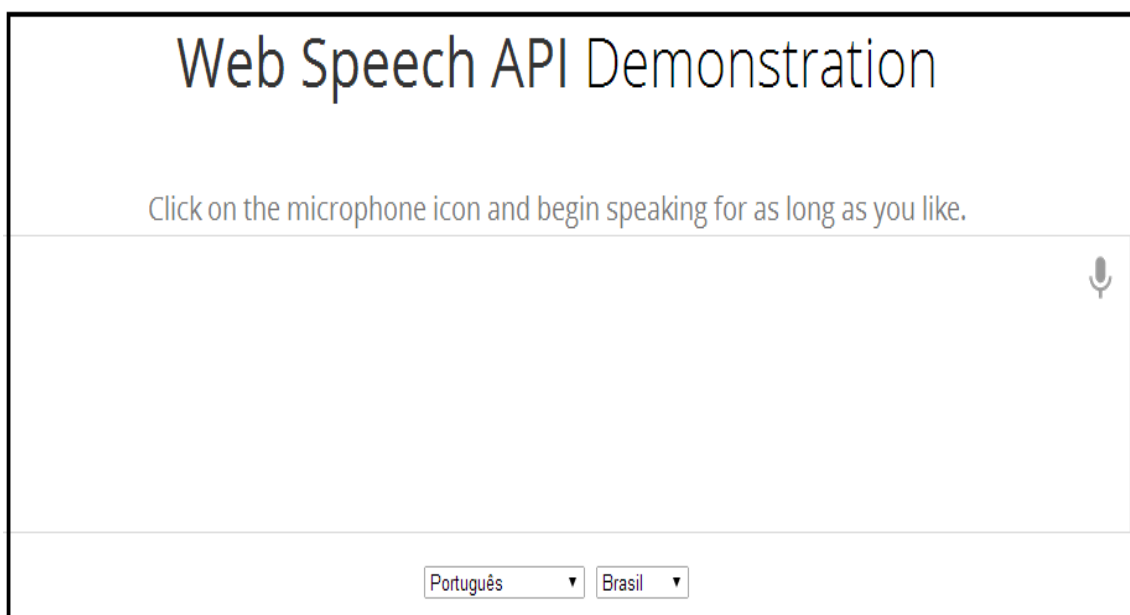


Figura 8 – Página de Demonstração da Webspeech API

Como base será utilizada a página de demonstração mostrada acima, por já possuir processos de captura de som e reconhecimento de palavras implementados. Algumas modificações deverão ser feitas para se adequar a página ao projeto, como a questão da língua e da forma de reconhecimento da fala.

### 3.5. Inteligência Artificial

Russell e Norvig (1995,p.2) apresentam no livro *Artificial Intelligence – A Modern Approach* oito diferentes definições para Inteligência Artificial, divididas em quatro grupos, classificados de 2 formas.

Pode-se primeiro dividir essas definições entre as que levam em consideração o pensamento, a forma de pensar, e as que levam em consideração o comportamento, a forma de agir. A segunda forma de se dividir é relacionada à forma de se medir o desempenho da IA criada. Enquanto alguns pesquisadores valorizam a fidelidade ao desempenho humano, outros preferem compará-la com um desempenho ideal, relativo a premissas racionais.

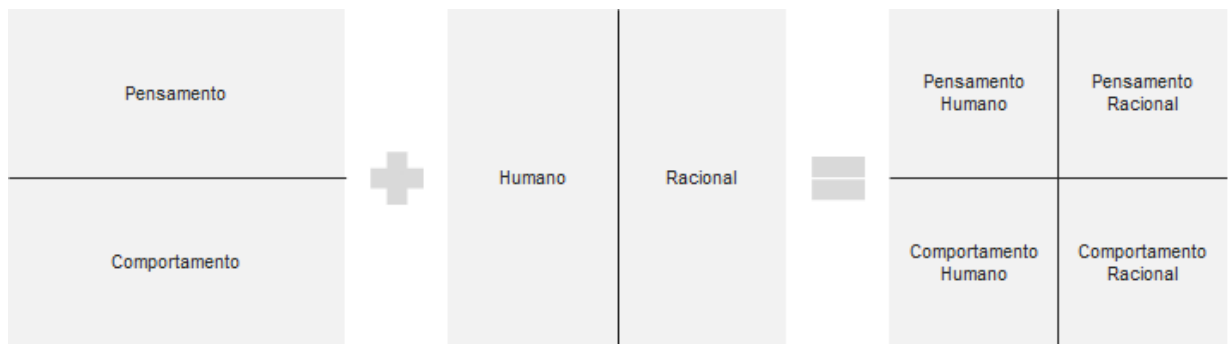


Figura 9 – Diferentes formas de se definir IA (Russell e Norvig, 1995, p.2)

Independente da escola de pensamento, a ideia geral e o objetivo da IA acaba sendo o mesmo para todas: desenvolvimento de máquinas capazes de interpretar situações, tomar decisões e realizar ações de maneira autônoma, isto é, sem a necessidade de um ser humano para controlá-la.

#### 3.5.1. Inteligência Artificial e o jogo de Xadrez

A indústria de jogos como um todo possui uma relação muito forte com os conceitos de IA, uma vez que a grande maioria destes jogos possui funções de jogo solo, no qual o adversário do jogador é a máquina. As tartarugas do Mário, os fantasmas do *Pacman*, os jogadores de futebol do jogo FIFA, todos possuem por trás uma máquina que analisa as possíveis ações e escolhe a melhor a ser tomada. Jogadores gostam de ser desafiados, portanto os desenvolvedores de jogos devem



tentar criar um balanço nos níveis de dificuldade dos jogos, de forma a atender os diferentes níveis de jogadores. Esse balanço é alcançado por intermédio de um trabalho bem feito na IA do jogo, que é responsável por controlar, ou “proporcionar inteligência” aos personagens do jogo que não são controlados pelo jogador (GALDINO, 2007).

No caso de uma partida de xadrez, a IA é utilizada para fazer o papel de adversário do usuário. Um famoso exemplo da aplicação de inteligência artificial no mundo do xadrez é o do computador Deep Blue, desenvolvido pela IBM, que em maio de 1997 se tornou o primeiro computador a derrotar um campeão mundial de xadrez numa partida com regras oficiais. O valor de mercado da IBM subiu US\$ 18 bilhões (RUSSELL E NORVIG, 1995).

De maneira simplificada, o que a IA de um software de xadrez faz é analisar todas as jogadas possíveis e determinar qual a que traz um melhor resultado para a máquina. Levando em consideração que a complexidade de um jogo de xadrez é de aproximadamente  $10^{46}$  (Herik, 2001), torna-se importante a criação de técnicas que minimizem a quantidade de jogadas a serem analisadas sem perder eficácia.

#### **3.5.1.1. Avaliação de posições**

Existem diversas formas e diversos fatores levados em consideração na avaliação de posições no tabuleiro. Os mais básicos são a quantidade de peças em jogo para cada jogador, ponderadas por suas respectivas pontuações, e a mobilidade de cada jogador (quantidade de movimentos possíveis).

Softwares mais sofisticados levarão em conta o momento do jogo (as peças passam a ter importâncias diferentes no decorrer da partida), a posição que cada peça ocupa no tabuleiro, a quantidade e disposição dos peões (uma quantidade menor de peões leva a um aumento no poder das torres, por exemplo), o domínio do centro do tabuleiro.

O objetivo das funções de avaliação é atribuir um determinado valor às diferentes posições das peças no tabuleiro, de modo a tornar possível classificá-las entre melhores e piores. A função avaliação é geralmente uma combinação linear de diferentes fatores com seus respectivos pesos.

### 3.5.1.2. Algoritmos de Busca – Minimax, Negamax, Alfa-Beta

Da mesma forma que os jogadores humanos, os algoritmos de IA para xadrez tendem a analisar mais de 1 lance no futuro. Considerando-se a máquina com as peças pretas, por exemplo, de forma a analisar a pontuação de um determinado movimento X das pretas, a máquina analisaria este movimento, todos os possíveis movimentos das brancas em resposta a este movimento X, todos os movimentos das pretas em resposta a cada uma das respostas ao movimento X, e assim sucessivamente. Na figura 10 pode-se observar parte da chamada árvore de movimentos possíveis.

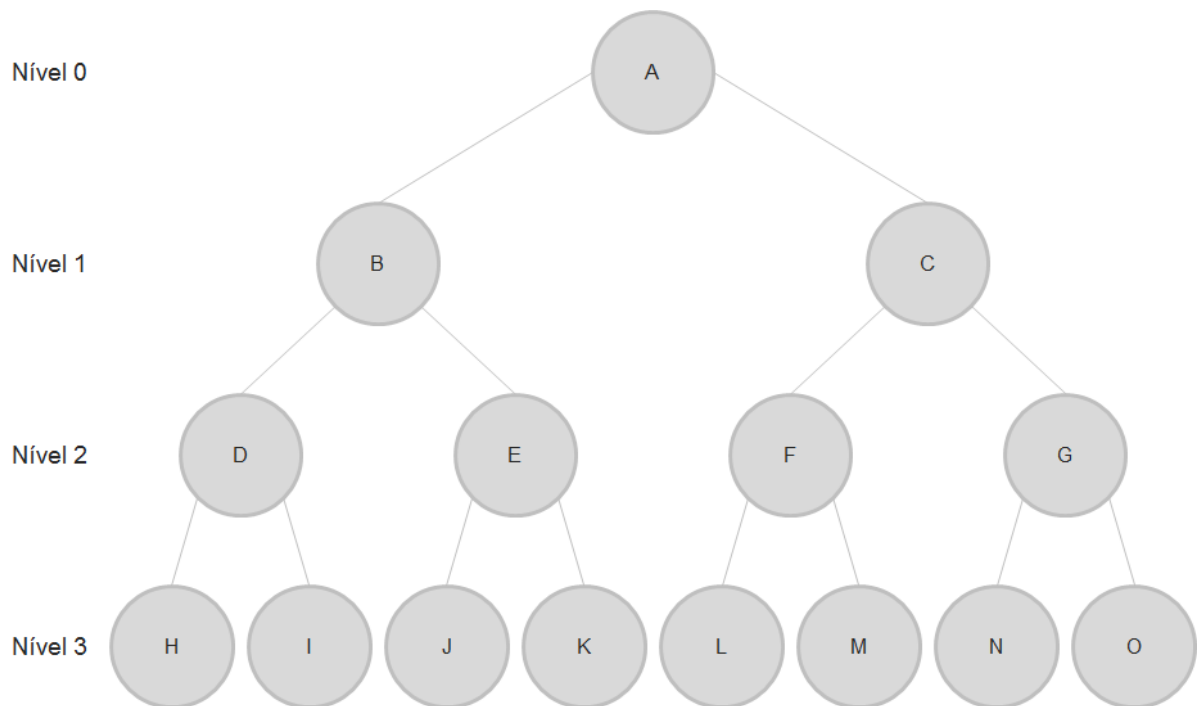
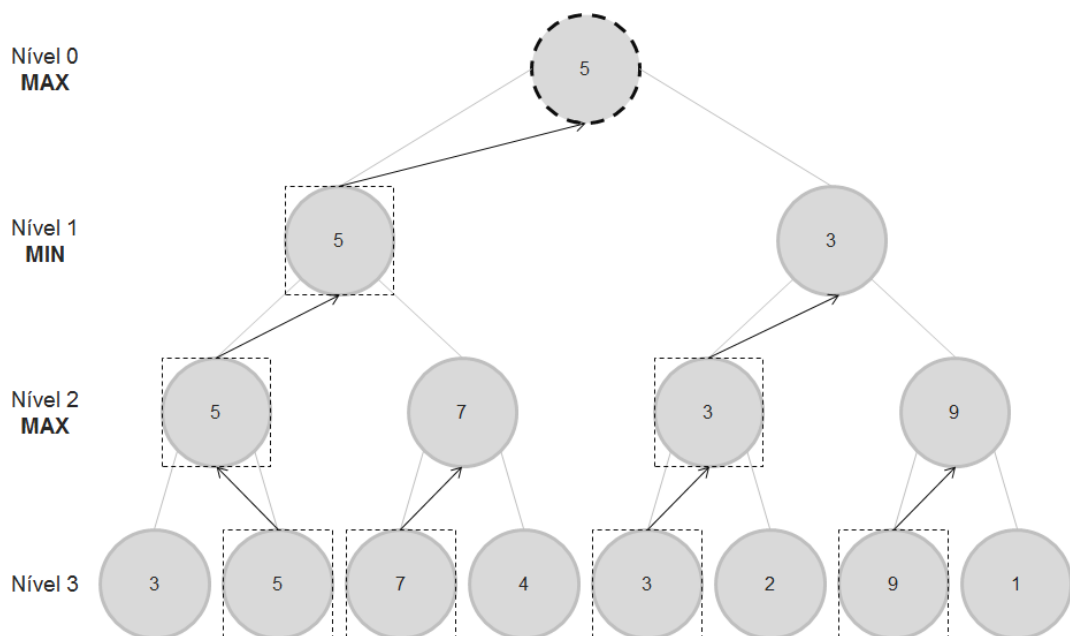


Figura 10 – Configuração da Árvore de Movimentos Possíveis

Depois de um movimento A do jogador 1, o jogador 2 possui as opções B e C como resposta. Supondo que ele se decida por responder com o movimento B, o jogador 1 poderá contra-responder com os movimentos D ou E, e assim sucessivamente. No caso do xadrez, considerando-se a posição inicial do tabuleiro, o jogador com as peças brancas possui 20 opções de movimento. Cada uma dessas opções pode ser respondida com 20 opções de movimento das peças pretas. Logo, é possível encontrar 400 configurações diferentes de tabuleiro considerando-se apenas um lance de cada jogador. À medida que se aumenta o nível da árvore de

movimentos, esse número cresce exponencialmente, dificultando bastante o trabalho de busca das inteligências artificiais.

Um algoritmo simples de busca é o Minimax, que busca minimizar a maior perda possível. O algoritmo pode ser explicado da seguinte forma: para árvores de profundidade 1 o algoritmo examina todos os movimentos possíveis e retorna o que fornece uma maior avaliação. Já para árvores de profundidade maior que 1, a máquina passa a analisar não só os próprios movimentos como também os movimentos do jogador em resposta a seus movimentos. A máquina deve considerar que o jogador também escolherá em resposta o movimento que retorne a melhor avaliação para ele, e conseqüentemente a pior avaliação para a máquina.



**Figura 11 – Algoritmo Minimax**

A figura 11 ilustra o método Minimax. Para cada nível par é selecionado o movimento de nível posterior que retorna a maior pontuação, ou seja, os níveis pares representam o momento em que o jogador realizou seu movimento e a máquina está então decidindo o movimento que lhe dê maior pontuação. Por outro lado, cada nível ímpar seleciona o movimento do nível posterior que retorna a menor pontuação, representando a melhor resposta possível que o jogador poderia realizar para um movimento da máquina.

Um jeito comum de se implementar o Minimax é por meio de sua versão simplificada, o algoritmo Negamax, no qual a avaliação de uma determinada posição

do ponto de vista de um jogador é igual a avaliação da mesma posição do ponto de vista do adversário multiplicada por -1.

$$\max(a, b) = -\min(-a, -b) \quad (1)$$

Quando se analisa apenas uma jogada na frente, um movimento que aparenta ser excelente para a máquina, pode acarretar em uma resposta por parte do jogador que leve a um cenário não tão bom do ponto de vista dela, portanto quanto mais profunda é a análise feita na árvore de possibilidades, maiores as chances de se encontrar o melhor movimento de fato. Porém, à medida que se aumenta a profundidade da árvore, a quantidade de análises a serem feitas cresce exponencialmente, o que pode tornar inviável o software, seja por questão de memória da máquina, seja por tempo de processamento. Portanto, é de extrema importância a utilização de técnicas para se “podar” esta árvore, isto é, eliminar os ramos que com certeza oferecem resultados piores do que resultados previamente encontrados, agilizando desta forma a busca.

O algoritmo Alfa-Beta aparece então como uma versão melhorada do Minimax, eliminando grandes porções da árvore de possibilidades. O algoritmo mantém duas variáveis, alfa e beta, que funcionam como limites máximo e mínimo. Suponha que a máquina esteja com as peças pretas e possua 3 movimentos possíveis, P1, P2 e P3. Para cada um destes movimentos as brancas podem responder com outros 3 movimentos, B1, B2, B3. Em um primeiro momento, o algoritmo analisa todas as possibilidades para o movimento P1 das pretas, isto é, P1-B1, B1-B2 e P1-B3, chegando no fim a uma certa avaliação. Em seguida é feita uma análise do movimento P2. O algoritmo então descobre que P2-B1 já retorna uma avaliação pior que a do movimento P1 do ponto de vista das pretas, portanto não há necessidade de se analisar P2-B2 e P2-B3, uma vez que pelo menos uma das respostas ao movimento P2 já a torna pior que P1. Não importa se B2 ou B3 levam a resultados piores ainda; uma refutação já é suficiente para mostrar que o movimento P1 é melhor.

Neste caso, a avaliação de P1 aparece como um limite mínimo. Qualquer avaliação menor que essa não serve para as peças pretas. Para análises com profundidade maior que 2 deve-se manter também um limite máximo, pois, se um movimento que parece ruim em uma certa profundidade leva a uma continuação

muito boa ao se analisar mais profundamente, o jogador adversário não permitirá que ele aconteça, pois havia algum outro movimento que ele poderia ter feito anteriormente para evitar tal situação. O limite máximo de um jogador é o limite mínimo de outro.

Para um melhor aproveitamento deste algoritmo torna-se necessária a implementação de algum algoritmo para tentar ordenar os movimentos, fazendo com que os melhores movimentos sejam testados primeiro, economizando assim cada vez mais tempo de procura.

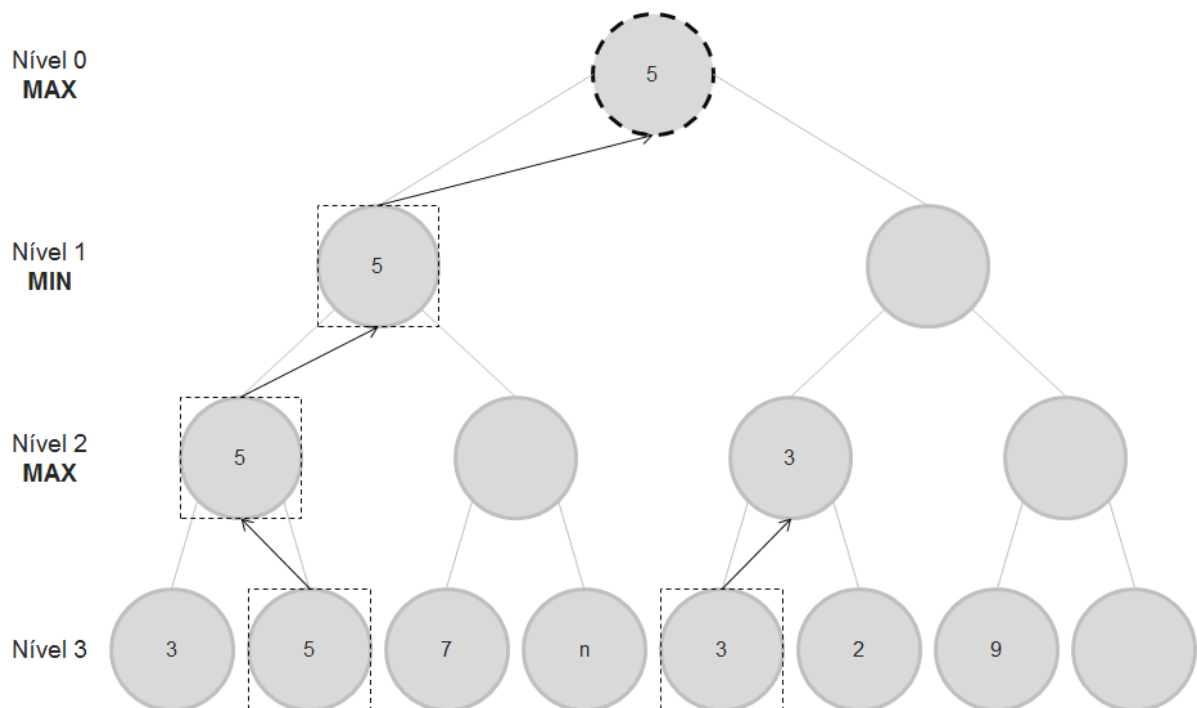


Figura 12 – Algoritmo Alfa-Beta

A figura 12 apresenta o mesmo exemplo da figura 11, porém utilizando o raciocínio Alfa-Beta. A ideia geral é a mesma: níveis pares procuram valores máximos e níveis ímpares procuram valores mínimos. A diferença está na criação de limites inferior e superior. Enquanto que na Minimax todos os nós do nível 3 seriam avaliados, no algoritmo Alfa-Beta alguns nós são vistos como sem necessidade de serem avaliados. Do lado esquerdo, por exemplo, tem-se que  $5 > 3$ , portanto sobe para o nível 2 e será comparado com o máximo valor entre os dois nós à sua direita. Descobre-se que um desses valores é 7. Neste momento dois casos são possíveis de acontecer:

i.  $n > 7$ :

Nesse caso,  $n$  subiria para o nível 2. Porém, se  $n > 7$ ,  $n > 5$ , portanto o número 5 subiria para o nível 1

ii.  $n < 7$ 

Nesse caso, o número subiria para o nível 2. Porém,  $5 < 7$ , portanto o número 5 subiria para o nível 1

Portanto, não há a necessidade de se saber qual o valor de  $n$ , pois, independente do caso, o número 5 passará para o nível 1. Dessa forma tempo de processamento é economizado, uma vez que buscas desnecessárias são eliminadas.

Para uma melhor eficácia do algoritmo Alfa-Beta é necessário que haja um poderoso algoritmo de ordenação de movimentos, fazendo com que os melhores movimentos apareçam primeiro. Quanto mais cedo estes movimentos aparecerem, mais nós serão cortados da busca. No melhor dos casos, ou seja, considerando que o melhor movimento é sempre escolhido primeiro, a quantidade de nós procurados reduz para um valor próximo à raiz quadrada da quantidade total de nós.

## 4. Implementação e Desenvolvimento do Projeto

Este trabalho pode ser dividido em 3 etapas:

- Desenvolvimento de uma página web utilizando HTML, CSS e JavaScript
- Implementação da Webspeech API para o reconhecimento de voz
- Implementação da inteligência artificial do jogo

A figura 13 mostra o nível 0 do diagrama de fluxo de dados do software *Xadrez por Reconhecimento de Voz*:



Figura 13 – Nível Zero do DFD do Software *Xadrez por Reconhecimento de Voz*

Antes de partir para a programação propriamente dita, algumas adaptações foram feitas ao jogo de xadrez:

- Optou-se por não se levarem em consideração movimentos especiais do xadrez como Roque e En Passant, uma vez que possuem regras e definições um pouco complicadas e sua ausência não afeta a dinâmica do projeto;
- Definiu-se que no movimento de Promoção o peão sempre transformará em uma dama, por esta ser geralmente a peça mais escolhida.
- Notação: as federações oficiais utilizam notações enxadrísticas específicas para se reproduzir o jogo. Tais notações podem ser facilmente entendidas por seres humanos, porém para facilitar o entendimento da máquina e evitar certas ambiguidades de lances (duas peças iguais podendo se movimentar para a mesma casa), definiu-se o seguinte padrão para notação dos movimentos:

<b>peça</b>	<b>coluna inicial</b>	<b>linha inicial</b>	<b>coluna final</b>	<b>linha final</b>
-------------	---------------------------	--------------------------	-------------------------	------------------------

#### 4.1. Desenvolvimento do jogo de xadrez em HTML, CSS e JavaScript

##### 4.1.1. Inicialização do tabuleiro e posicionamento das peças

O primeiro passo foi gerar um código para a inicialização do tabuleiro quadriculado e posicionamento das peças em suas casas iniciais, usando como base o arquivo “index.html”.

Dentro do arquivo “estilo.css” definiu-se primeiro o tamanho do tabuleiro em 450 pixels x 450 pixels:

```
#tabuleiro {  
    width: 450px;  
    height: 450px;  
    float: left;  
    margin-left: 10px;  
}
```

A propriedade *float* força o elemento em uma posição específica da tela (direita ou esquerda), permitindo que outros elementos sejam colocados a seu redor.

Em seguida, definiu-se duas classes para representar as casas do tabuleiro:

```
.casa_branca{ background-color: #fff; }  
  
.casa_preta{background-color: #ccc;}
```

No arquivo “iniciarTabuleiro.js” foi utilizado o método `append()` da linguagem JavaScript que permite a inserção e/ou alteração das propriedades dos elementos



presentes no código HTML. Um tabuleiro quadriculado então foi criado, com cada casa recebendo um id da forma  $i-j$ , no qual  $i$  corresponde à linha e  $j$  corresponde a coluna, permitindo assim a fácil identificação das casas. Em seguida, as peças vistas na figura 14 foram adicionadas em suas respectivas posições com o auxílio do seguinte comando:

```
$("#j-j").append("<img src='imagem.png'>");
```

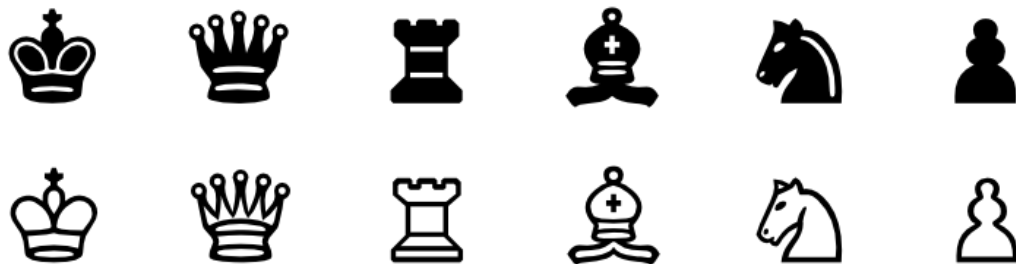
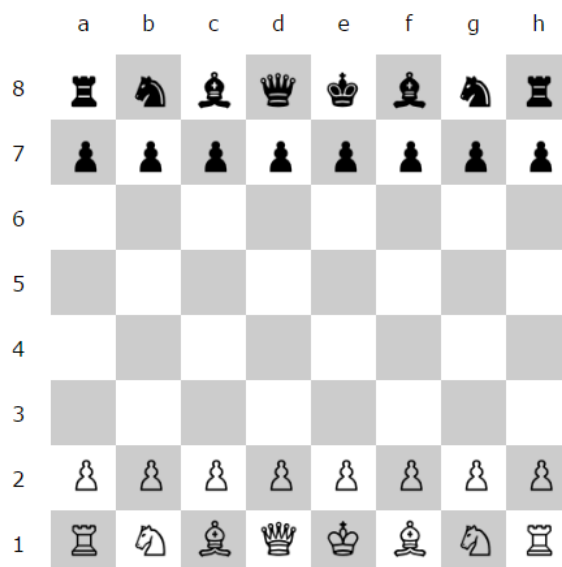


Figura 14 – Imagens utilizadas como peças, retirado do site [http://jpe.gs/wiki/Image:Chess\\_set\\_symbols\\_igor\\_k\\_01.svg](http://jpe.gs/wiki/Image:Chess_set_symbols_igor_k_01.svg)

Esse processo resulta na configuração vista na figura 15.

## Xadrez por Reconhecimento de Voz



Trabalho de Conclusão de Curso  
Pablo Henrique Alves de Oliveira - 6920435

Figura 15 – Página web com a configuração inicial do tabuleiro de xadrez

#### 4.1.2. Implementação das regras e dinâmica do jogo do Xadrez

Uma vez inicializado o tabuleiro, a logística do jogo de xadrez deve então ser implementada. Nessa parte deve-se gerar o código que fará o papel de checar se a String (conjunto de caracteres representando o movimento) recebida pode ser interpretada como um movimento; caso possa, deve-se checar então se o movimento recebido é um movimento válido de acordo com as regras do xadrez, e, caso seja, realizar o movimento acionado pelo jogador.

Na figura 16 é possível ver o nível 1 do DFD do software em questão.

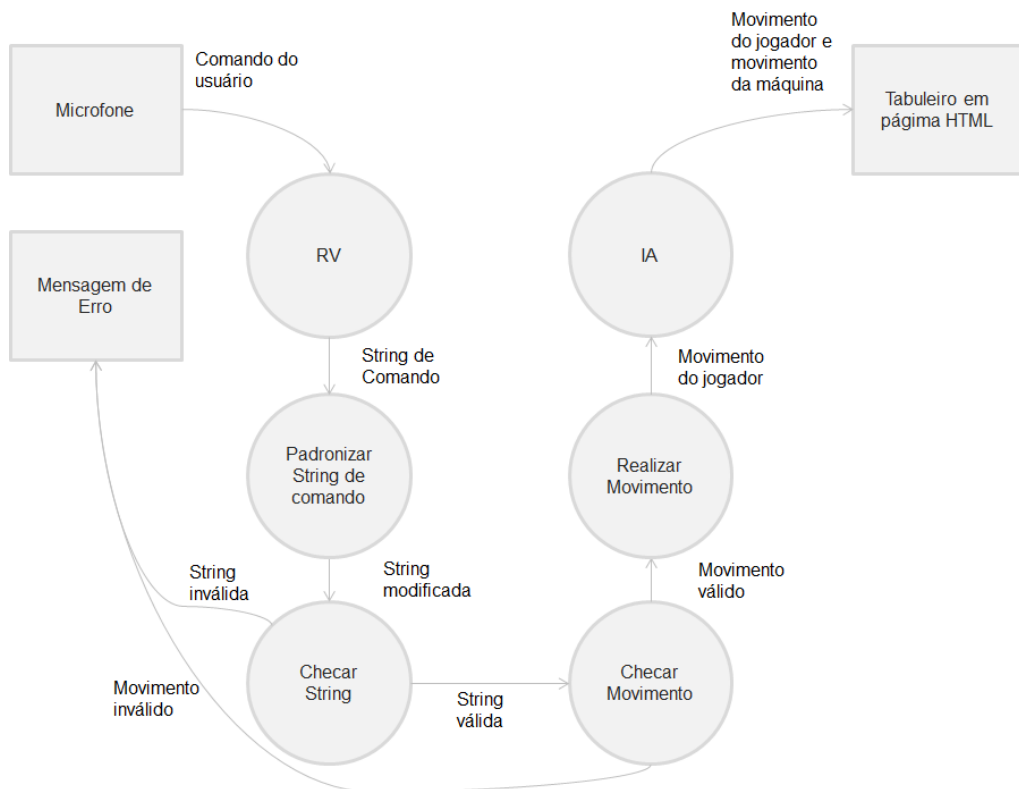


Figura 16 – Nível 1 do DFD do Software *Xadrez por Reconhecimento de Voz*

Para poder checar o comando recebido deve-se primeiro modificá-lo de forma a haver uma padronização na String passada para a função de teste. Portanto, primeiramente retira-se todos os espaços da frase recebida como comando e leva todas as letras para sua forma minúscula através dos comandos:

```

stringModificada = comandoRecebido.replace(/ /g, "");
stringModificada = stringModificada.toLowerCase();

```

Em seguida, novamente com o auxílio do comando replace, as palavras correspondentes às peças são substituídas por sua inicial maiúscula. Por exemplo: o comando “peaoe2e4” é transformado em “Pe2e4”.

Por último, as letras que correspondem às colunas inicial e final do movimento são substituídas por números seguindo a lógica a = 1, b = 2 e assim por diante.

Com isso, tem-se que comandos como “peao e2 e4”, “peão e 2 e 4”, “PeAoe2e4”, “peAO e2e 4” e todas as outras variantes possíveis serão sempre interpretados como P5254, ou seja, o peão da quinta coluna e segunda linha deve se movimentar para a casa na quinta coluna e quarta linha. Caso o resultado da string modificada ao final desse processo seja diferente deste formato, tem-se que o comando foi acionado de forma errada pelo usuário. Caso contrário, a string modificada deve então ser testada com relação à validade do movimento.

O teste de validade do movimento segue o fluxo mostrado na figura 17,

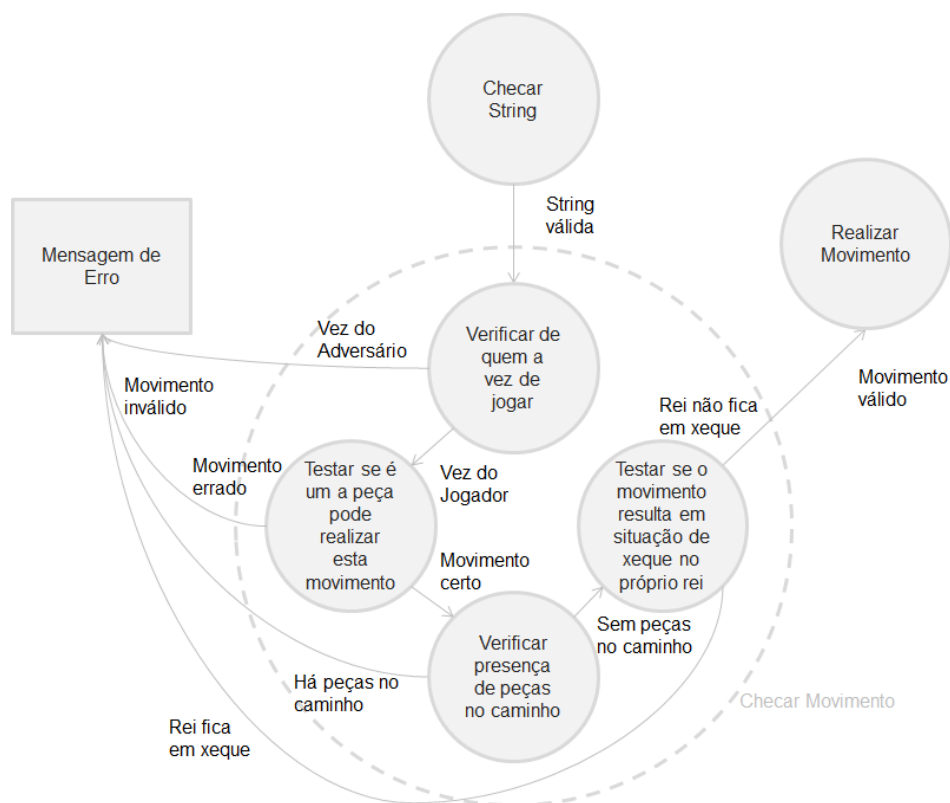


Figura 17 – Nível 2 do DFD do Software *Xadrez por Reconhecimento de Voz* (Checar Movimento)

Primeiro é verificado de quem é a vez de jogar e se a peça selecionada é deste jogador. Logo depois é testado se o movimento solicitado é possível de ser realizado pela peça em questão, isto é, se uma torre foi solicitada a se movimentar

na vertical ou horizontal, se um bispo foi solicitado a se movimentar na diagonal, e assim por diante, seguindo-se as regras de movimento e captura previamente descritas.

Em seguida, é verificado se há alguma obstrução no caminho da peça entre sua casa de destino e sua casa de origem, caso não sejam casas adjacentes. No caso do cavalo e do rei não há a necessidade de se realizar esse teste, uma vez que o cavalo não possui essa restrição em seu movimento e o rei só pode se movimentar uma casa por vez.

Por último é testado se o movimento realizado deixa de alguma forma o próprio rei em xeque, o que violaria as regras do xadrez. Para isso, o programa realiza o movimento, verifica se o rei possui alguma peça atacando-o e em seguida desfaz o movimento. Caso nenhum movimento possa ser realizado, configura-se então uma situação de xeque-mate.

Uma vez que o movimento solicitado tenha passado por todos esses testes, ele é de fato então realizado. Para isso o programa armazena a peça a ser movida – que é acessada a partir do ID “linha inicial – coluna inicial” –, apaga o que tem na casa de destino, cola a peça na casa de destino e apaga a casa de origem. Todo esse algoritmo encontra-se no arquivo “movimentarPaca.js”.

## **4.2. Implementação Reconhecimento de Voz**

### **4.2.1. Webspeech API**

Para a inclusão da Webspeech API na página, responsável pelo reconhecimento de voz, utilizou-se como base uma página de demonstração disponibilizada pelo próprio Google, que já apresenta procedimentos de captura do som e reconhecimento da sentença falada.

Foi criado, primeiramente, um novo objeto do tipo `webkitSpeechRecognition`, que possui todas as propriedades necessárias para o reconhecimento de voz, ou seja:

```
var reconhecimentoDeVoz = new webkitSpeechRecognition();
```

Em seguida, com base no projeto de Carlson e Rodrigues (2013), determinou-se quais funções eram relevantes para o programa, e quais modificações deveriam ser feitas. A primeira alteração foi na questão da língua em que o reconhecimento seria feito. Alterou-se então a propriedade “lang” do objeto reconhecimentoDeVoz para português brasileiro.

Outra propriedade alterada foi a de reconhecimento contínuo, que possui como padrão o valor “false”, o que significa que o reconhecimento acaba no momento em que o usuário para de falar. Neste trabalho optou-se por trabalhar com um reconhecimento contínuo, uma vez que alguns comandos, principalmente algumas letras, devem ser ditos de maneira pausada para uma melhor interpretação do programa.

Por último, alterou-se a propriedade “interimResults” para “true”, fazendo com que uma prévia do resultado seja constantemente mostrada.

```
reconhecimentoDeVoz.lang = 'pt-BR';  
reconhecimentoDeVoz.continuous = true;  
reconhecimentoDeVoz.interimResults = true;
```

Feito isso, foram definidas 4 funções:

- Função onstart: responsável por iniciar o reconhecimento de voz.
- Função onerror: verifica a ocorrência de algum erro, como a falta de microfone, microfone bloqueado, ou ausência de comando.
- Função onend: indica que o processo de reconhecimento de voz foi finalizado
- Função onresult: armazena o resultado obtido

A função onstart é chamada quando o usuário clica na imagem do microfone presente na tela e o reconhecimento de voz se inicia (figura 18). O reconhecimento só vai parar quando o usuário clicar novamente na imagem do microfone. No caso de uma pessoa paraplégica, pode-se programar o software para realizar um reconhecimento contínuo de voz, sendo que o reconhecimento do movimento de xadrez começaria quando a pessoa dissesse e, conseqüentemente, o software interpretasse um comando específico, como, por exemplo, “iniciar”.

## Xadrez por Reconhecimento de Voz



Trabalho de Conclusão de Curso  
Pablo Henrique Alves de Oliveira - 6920435

**Figura 18 – Processo de Reconhecimento de Voz: início**

Enquanto o segundo clique não ocorre, uma prévia do que está sendo falado aparece na caixa ao lado do tabuleiro (figura 19).

## Xadrez por Reconhecimento de Voz

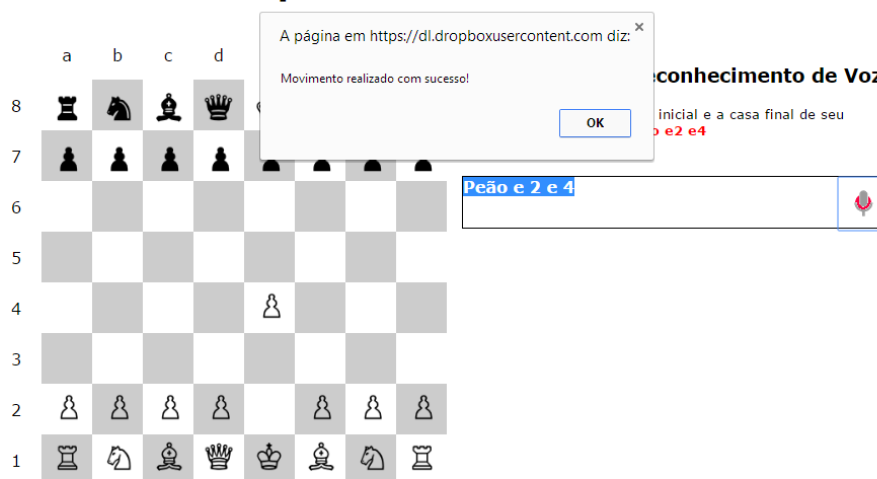


Trabalho de Conclusão de Curso  
Pablo Henrique Alves de Oliveira - 6920435

**Figura 19 – Processo de Reconhecimento de Voz: captura da fala**

Uma vez que o usuário seleciona o fim do reconhecimento de voz, o valor presente na caixa de resultado é enviado ao algoritmo de xadrez, transformando o comando falado em um movimento no tabuleiro (figura 20).

## Xadrez por Reconhecimento de Voz



Trabalho de Conclusão de Curso  
Pablo Henrique Alves de Oliveira - 6920435

Figura 20 – Processo de Reconhecimento de Voz: Realização do movimento

### 4.2.2. Hospedagem da Página Web no Dropbox

Programas em html, javascript e css podem ser abertos no navegador sem necessidade de conexão com a internet, sem necessidade de hospedagem do site em um servidor. No caso da Webspeech API, porém, para que o programa consiga ter acesso ao microfone e as caixas de som, é necessário que os arquivos estejam hospedados em algum servidor http ou https.

Por se tratar de um programa com somente elementos estáticos, uma solução encontrada para este problema foi a utilização de sites como googledrive.com e dropbox.com para a hospedagem da página web. O escolhido foi o dropbox, por se mostrar mais fácil.

O procedimento de hospedagem é bem simples:

- i. Criar uma conta no site dropbox.com;
- ii. Criar uma pasta pública;
- iii. Colocar os arquivos js, html e css dentro da pasta pública, respeitando as pastas e caminhos definidos dentro dos arquivos;
- iv. Compartilhar o link público do arquivo.

Com isso, qualquer pessoa passa a ter acesso à página web, desde que possua o link para acessá-la. É uma forma simples, rápida e barata de se hospedar sites estáticos, porém não funciona caso a página web contenha arquivos php ou ajax, por exemplo. Outra ressalva está na quantidade de arquivos que você pode armazenar na pasta da sua página, que se limita à capacidade de armazenamento da conta no dropbox; porém, para o objetivo do projeto, esse método se mostra mais que suficiente. O código para a Webspeech API encontra-se nos arquivos “index.html” e “reconhecerVoz.js”.

#### **4.2.3. Speech Synthesis API**

A Speech Synthesis API aparece para fazer o papel inverso da Webspeech API, isto é, transformar textos em áudios. Seu uso é fácil e permite alteração em parâmetros como voz, volume, velocidade da fala e idioma, isto é:

```
var mensagem = new SpeechSynthesisUtterance(texto);  
window.speechSynthesis.speak(mensagem);
```

### **4.3. Implementação da Inteligência Artificial**

Como dito anteriormente, a IA de um jogo de xadrez se baseia em 2 pontos:

- Encontrar todos os movimentos possíveis;
- Avaliar cada um dos movimentos a fim de se determinar o melhor.

Foi utilizada a linguagem JavaScript para realizar a função dos algoritmos descritos a seguir.

#### **4.3.1. Avaliação de Posições**

Para a parte de avaliação de posições, um algoritmo compacto foi criado levando-se em consideração apenas a pontuação das peças (figura 21) e a mobilidade de um jogador, isto é, a quantidade de lances que ele pode realizar naquela posição.





Figura 21 – Pontuação das peças de Xadrez

Vale ressaltar que a avaliação de uma jogada é realizada não com relação ao movimento propriamente dito, mas com relação ao resultado de seu movimento. Para isso é necessário que o algoritmo criado primeiro realize o movimento, determine um valor numérico que será interpretado como a avaliação deste movimento, e em seguida desfça o movimento.

Este valor numérico é determinado pela seguinte equação:

$valor =$

$$\sum_{todas\ as\ peças} peso_{peça} \times (quantidade\_da\_peça_{pretas} - quantidade\_da\_peça_{brancas}) + peso_{mobilidade} \times (mobilidade_{pretas} - mobilidade_{brancas}) \quad (2)$$

Outros fatores podem ser levados em consideração para se determinar o valor de uma posição. O momento do jogo influencia na importância das peças (início, meio ou fim), a estrutura de peões também influencia no poder de algumas peças, a posição das peças no tabuleiro também influencia em sua força. Por exemplo, a medida que a quantidade de peões presentes no jogo diminuem, as torres se tornam mais fortes, enquanto que em posições mais fechadas, com uma menor mobilidade em geral, os cavalos passam a possuir uma importância maior, uma vez que é a única peça que pode “saltar” sobre as outras,

Para o intuito deste projeto, porém, os dois fatores levados em consideração já são suficientes.

#### 4.3.2. Algoritmo Negamax

Para a IA do jogo de xadrez dois métodos promissores de busca foram implementados e testados. O primeiro método é o Negamax, ou Negated Minimax. O algoritmo da função Negamax pode ser escrito genericamente da seguinte forma:

```

função Negamax(profundidade)
  se profundidade = 0
    retorna o valor da função avaliação
  fim se
  valor_máximo ←  $-\infty$ 
  para (todos os movimentos possíveis)
    realiza movimento
    pontuação ← - Negamax(profundidade - 1)
    se pontuação > valor_máximo
      valor_máximo ← pontuação
    desfaz movimento
  fim para
retorna valor_máximo

```

Dessa forma, a função Negamax é chamada de maneira iterativa, fazendo com que todos os níveis da árvore de movimentos possíveis até uma determinada profundidade sejam atingidos e avaliados. Aqueles que oferecem um melhor resultado vão sendo mantidos, sempre levando em consideração que ambos os jogadores farão as melhores escolhas possíveis.

A função Negamax é inicialmente chamada logo após a realização do movimento do usuário e retorna apenas a pontuação máxima capaz de ser alcançada. É preciso então criar variáveis que armazenem também o movimento que gerou aquela pontuação. Vale ressaltar que este movimento encontra-se na primeira camada da árvore, e que os outros movimentos, das outras camadas, são apenas respostas aos movimentos realizados dessa primeira camada, bem como respostas a estas respostas, e assim por diante.

Para que o algoritmo Negamax funcione perfeitamente é necessário que a função de avaliação das posições retorne um valor referente a quem está sendo avaliado, isto é, caso a vantagem seja de quem está sendo avaliado, a função deve retornar um valor positivo, enquanto que, caso a vantagem fique do lado adversário, o número retornado será negativo.

### 4.3.3. Algoritmo Alfa-Beta

Esse algoritmo, criado de maneira independente por diferentes pesquisadores dos anos 50, aparece com o objetivo de melhorar o algoritmo Minimax. Enquanto que no Minimax todos os movimentos possíveis são testados, o algoritmo Alfa-Beta foi feito de forma a eliminar movimentos que certamente apresentam resultados piores. Tomando como base o algoritmo Negamax, pode-se escrever o Alfa-Beta da seguinte forma:

```

função AlfaBeta(alfa, beta, profundidade)
  se profundidade = 0
    retorna o valor da função avaliação
  fim se
  para (todos os movimentos possíveis)
    realiza movimento
    pontuação ← - AlfaBeta(-beta, -alfa, profundidade - 1)
    se pontuação >= beta
      retorna beta
    fim se
    senão se pontuação > alfa
      alfa ← pontuação
    fim senão se
  desfaz movimento
fim para
retorna alfa

```

Esse algoritmo é chamado na função principal, assim como o anterior. É necessário passar valores iniciais de alfa e beta, com alfa tendendo a  $-\infty$  e beta tendendo a  $+\infty$ . Assim como na Negamax, é necessário criar variáveis para armazenar os valores das linhas e colunas do movimento, uma vez que alfa-beta também retorna apenas a pontuação máxima.

## 5. Resultados e Análise

O programa feito HTML, CSS e JavaScript gerou um software capaz de receber comandos de voz, aplicar as regras de xadrez, analisar situações e determinar lances respostas. A página web criada possui a seguinte interface inicial:

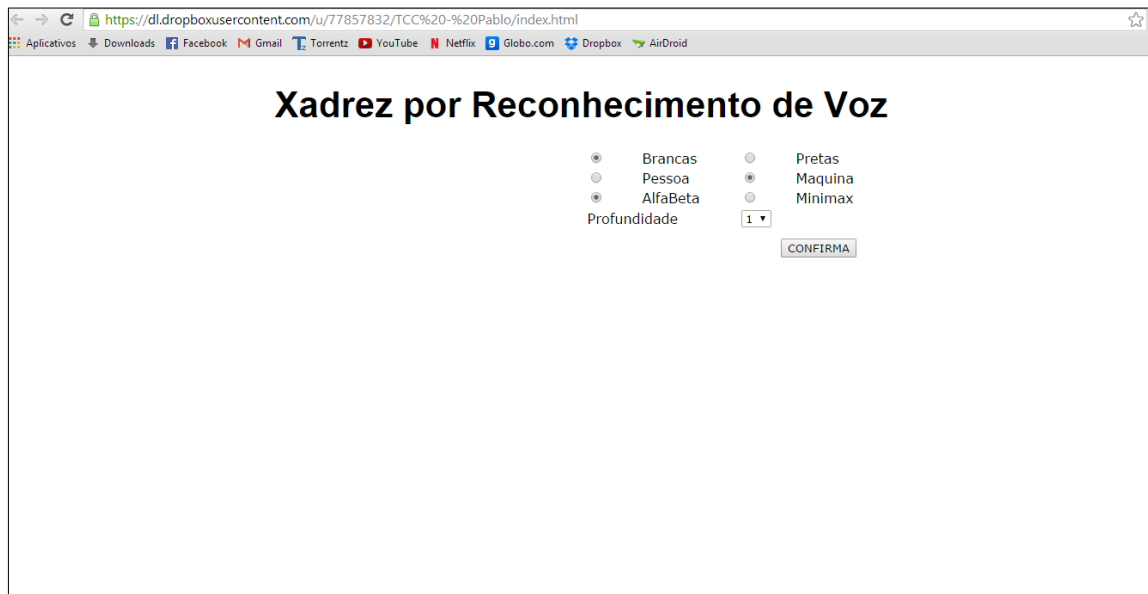


Figura 22 – Tela inicial da página Xadrez por Reconhecimento de Voz

Assim que o usuário define com que cor quer jogar e qual será seu adversário o tabuleiro é inicializado e o jogo pode então começar, como na figura 1.



Figura 23 – Tela de jogo da página Xadrez por Reconhecimento de Voz

De maneira resumida, a sequência de funcionamento é a seguinte:

- i. O usuário define com que cor pretende jogar e qual será seu adversário, tendo como opções: algoritmo alfa-beta, algoritmo minimax e ainda a opção de não jogar contra uma inteligência artificial, e sim contra outro jogador humano de maneira local.
- ii. Quando o usuário clica no microfone é iniciado o reconhecimento de voz, que é interrompido por um segundo clique no microfone.
- iii. O movimento escolhido pelo usuário passa por uma série de testes para se checar se é um movimento válido.
- iv. Caso seja, o movimento é então realizado, e o computador realiza um lance em resposta. Caso contrário, uma mensagem de erro aparece e o usuário recebe mais uma chance para dizer seu movimento.

A página web foi implementada e seu funcionamento foi verificado com sucesso, realizando de exatamente o que lhe foi inicialmente proposto. No entanto, alguns fatores observados devem ser levados em consideração.

- **Reconhecimento de Voz**

Quando o movimento era falado de forma muito rápida, a transcrição nem sempre se mostrava fiel ao que o usuário falou. Palavras bem diferentes das que foram distas eram mostradas, e outras eram esquecidas.

Com relação às peças, dificuldades foram encontradas no caso do “peão”, que por várias vezes era interpretado como “pião”. As demais peças possuem pronúncias bem distintas de outras palavras, e portanto não apresentaram muitos problemas. Já no caso das colunas, a API teve problemas para identificar movimentos na coluna “e”, que por diversas vezes era interpretada como “é” e em outras simplesmente não era reconhecido. Esses problemas se devem a dificuldade em se interpretar diferentes sotaques. Não há um jeito certo de se pronunciar a letra “e”, e tanto a pronúncia “ê” quanto a “é” podem ser interpretadas como palavras com significado real. No caso do peão, é normal as pessoas pronunciarem a letra “e” como letra “i” em certas palavras, o que nesse caso pode ser um problema, uma vez que ambas as palavras, “peão” e “pião”, existem, podendo ser igualmente interpretadas.

Outro fato interessante observado foi a tentativa API em interpretar o que o usuário estava falando. Por exemplo, quando tentando dizer “peão e2 e4”, o software interpretava que a palavra “mil” havia sido dita, retornando “peão e2004”, uma vez que “dois mil e quatro” faz mais sentido que “dois e quatro”.

Outro ponto foi a questão de o software reconhecer e transcrever palavras em diferentes idiomas mesmo tendo sido direcionado para fazer o reconhecimento apenas no idioma português. Porém, apesar de essas palavras serem interpretadas, os movimentos devem ser especificados em português para que possam ser validados pelo código do jogo de xadrez.

No geral, quando o movimento era pronunciado de maneira lenta e em alto e bom som, o resultado obtido era o resultado esperado. Erros vistos com frequência podem ser tratados dentro do código do jogo, para serem interpretados como certos.

- **Inteligência Artificial**

Com relação à avaliação das posições, o software funcionou bem, porém de forma limitada, uma vez que o algoritmo de avaliação criado era bem compacto, sendo levados em consideração poucos aspectos do jogo. Para uma aplicação básica em um nível de iniciante, o algoritmo se mostrou suficiente.

Com relação aos algoritmos de busca, notou-se que de fato há uma diferença na eficiência de ambos. Testou-se ambos os algoritmos em diferentes profundidades e para diferentes lances no jogo.

As tabelas a seguir apresentam os resultados encontrados com relação ao tempo de resposta do algoritmo, ao número de nós analisados e à pontuação máxima encontrados, considerando-se sempre a mesma sequência de movimentos para as brancas: peão e2 e4 ->cavalo g1 f3 ->bispo f1 c4

**Tabela 1 – Pontuação máxima obtida pelos algoritmos de busca**

pontuação máxima	Negamax			Alfa-Beta		
	Profundidade	Lance 1	Lance 2	Lance 3	Lance 1	Lance 2
1	2,9	2,4	0,29	2,9	2,6	0,7
2	2,7	3,3	3,3	2,7	3,3	3,3
3	2,9	1,6	1,7	2,9	1,6	1,7

A tabela 1 mostra que os dois algoritmos retornam de fato o mesmo resultado. Porém, a tabela 2 a seguir comprova a maior eficiência do algoritmo Alfa-Beta uma vez que este método precisa de analisar menos posições que o Negamax para tomar uma decisão.

**Tabela 2 – Nós analisados pelo algoritmo de busca**

nós analisados	Negamax			Alfa-Beta		
	Lance 1	Lance 2	Lance 3	Lance 1	Lance 2	Lance 3
1	20	29	33	20	21	31
2	620	865	1.208	195	313	801
3	13.780	15.213	23,603	1.810	716	3.528

O fato de analisar menos posições faz com que o algoritmo Alfa-Beta seja várias vezes mais rápido que o Negamax, como pode ser visto na tabela 3.

Para a medição de tempo foi utilizada uma função em JavaScript que retorna a data, incluindo os segundos, em que ela for chamada. A função foi então chamada duas vezes: uma imediatamente antes da aplicação do algoritmo da IA e uma imediatamente depois. A diferença entre estas duas datas representa o tempo em segundos que o algoritmo levou para chegar a um resultado.

**Tabela 3 – Tempos gastos na realização de movimentos**

tempo (s)	Negamax			Alfa-Beta		
	Lance 1	Lance 2	Lance 3	Lance 1	Lance 2	Lance 3
1	0,5	0,6	0,7	0,5	0,5	0,7
2	10,8	16,7	26,1	3,2	6,3	18,4
3	276,4	298,1	632,6	37,3	14,5	68,3

Nota-se então que para uma mesma profundidade, em um mesmo número de lances, o algoritmo Alfa-Beta consegue chegar ao mesmo resultado do algoritmo Negamax realizando uma busca por menos nós e levando menos tempo, comprovando assim sua maior eficiência em comparação com o outro.

## 6. Considerações Finais

Cada vez mais o ser humano tenta diminuir a distância entre e as máquinas. Tanto as tecnologias de reconhecimento de voz quanto os estudos sobre inteligência artificial fazem parte deste processo de aproximação. Ambos os campos já estão bem avançados, mas ainda há espaço para melhorias.

No caso do reconhecimento de voz ficou claro que a dificuldade de se implantar sistemas que realizam este reconhecimento de maneira perfeita se encontra nas diferentes formas de falar das pessoas. Diferenças nos sotaques, expressões, idiomas, timbres de voz, velocidades na fala, todos esses são fatores que muitas vezes dificultam o entendimento até mesmo entre seres humanos. As tecnologias neste ramo, no entanto, estão cada vez mais avançadas e fazendo parte do nosso dia a dia. A Webspeech API, por exemplo, usada neste trabalho, aparece como uma biblioteca de fácil entendimento, para que qualquer pessoa, mesmo sem conhecimentos muito aprofundados de programação, possa criar aplicações utilizando recursos de reconhecimento de fala.

O ramo da inteligência artificial, por sua vez, já vem sendo estudado há muito tempo, tendo visto avanços significativos. É um ramo essencial para a indústria de jogos, uma vez que na maioria das vezes o jogador não encontra algum adversário humano para jogar, e acaba tendo que jogar contra a máquina. Quanto aos algoritmos de IA utilizados neste projeto, pôde-se perceber que são algoritmos inviáveis de serem utilizados em jogos de xadrez, uma vez que a quantidade de movimentos possíveis nesse jogo cresce de forma exponencial, tendendo a um número extremamente grande.

Para se viabilizar a utilização do algoritmo Alfa-Beta é necessário que se aplique técnicas mais sofisticadas, principalmente no quesito de ordenação de movimentos, isto é, quais movimentos serão checados primeiro. Utilizando-se este algoritmo para uma busca em profundidade 2, foi possível realizar partidas longas, com jogadores de nível iniciante. Outro ponto que pode ser melhorado é o algoritmo de avaliação de posições, adicionando outros fatores importantes do jogo.

No que diz respeito a implementação, pode-se dizer que para os objetivos deste projeto as linguagens de programação utilizadas se mostraram suficientes. A sintaxe da linguagem javascript é bastante parecida com as linguagens C e Java, o



que faz com que seja fácil de ser entendida. Porém, ela apresentou dificuldades nos momentos de localização e reparo de erros por, diferentemente das linguagens compiladas, não possuir um programa com um debuggador inserido para rodar o código e determinar onde se encontram os erros. A implementação dos algoritmos de busca e de avaliação de posições não apresentou muitas dificuldades, porém, para algoritmos mais complexos é possível que outras linguagens devam ser utilizadas.

O xadrez é um jogo que há muitos anos fascina e instiga a curiosidade do ser humano, por ser simples de ser jogado e ao mesmo tempo bastante complexo com relação à estratégia, e que com utilização das tecnologias existentes hoje pode se tornar cada vez mais interessante.

### **6.1. Trabalhos Futuros**

Algumas sugestões de trabalhos futuros:

- Criação de um sistema de hardware que realize as jogadas fisicamente, ou seja, com um tabuleiro e peças reais.
- Aprimoramento dos algoritmos de inteligência artificial, utilizando conceitos como ordenação de movimentos, aprofundamento iterativo e tabelas de transposição.
- Utilização de reconhecimento de voz em outros jogos.

## 7. Referências

BERWICK, Bob. **Game trees, minimax, & alpha-beta search**. Disponível em: <<http://web.mit.edu/6.034/wwwbob/handout3-fall11.pdf>>. Acesso em: 27 set. 2014.

CHIRRAVURI, Varrun; KUO, Michael. **Voice-Controlled chess game on FPGA using dynamic time warping**. Disponível em: <[http://web.mit.edu/6.111/www/f2008/projects/mikekuo\\_Project\\_Final\\_Report.pdf](http://web.mit.edu/6.111/www/f2008/projects/mikekuo_Project_Final_Report.pdf)>. Acesso em: 13 abr. 2014.

CROCKFORD, Douglas. **Javascript: The World's Most Misunderstood Programming Language**. Disponível em: <<http://javascript.crockford.com/javascript.html>>. Acesso em: 10 set. 2014.

GALDINO, Carlos Henrique Silva. **Inteligência Artificial aplicada no desenvolvimento de jogos de computador**. 2007, Itajubá.

GRILLO, Filipe del Nero; FORTES, Renata Pontin de Mattos. **Aprendendo JavaScript**. 2008. Disponível em: <[http://www.icmc.usp.br/CMS/Arquivos/arquivos\\_enviados/BIBLIOTECA\\_113\\_ND\\_7\\_2.pdf](http://www.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND_7_2.pdf)>. Acesso em: 12 set. 2014.

HERIK, H. Jaap van Den; UITERWIJK, Jos W.h.m.; VAN RIJSWIJCK, Jack. **Games solved: Now and in the future**. Elsevier Science B.v., 2001.

KNUTH, Donald E.; MOORE, Ronald W.. **An Analysis of Alpha-Beta Pruning**. Stanford: North-holland Publishing Company, 1975.

PAGINA DE DEMONSTRAÇÃO WEBSPEECH API. Disponível em <https://www.google.com/intl/pt/chrome/demos/speech.html>.

PLAAT, Aske et al. **Exploiting Graph Properties of Game Trees**. Disponível em: <[https://webdocs.cs.ualberta.ca/~jonathan/publications/ai\\_publications/graph.pdf](https://webdocs.cs.ualberta.ca/~jonathan/publications/ai_publications/graph.pdf)>. Acesso em: 14 out. 2014.

PORTAL BAQUARA. Disponível em <http://baquara.com/xadrez/>.

PORTAL CHESSPROGRAMMING. Disponível em <http://chessprogramming.wikispaces.com/>.

PORTAL DEVMEDIA. Disponível em <http://www.devmedia.com.br/>.

PORTAL SOXADREZ. Disponível em <http://www.soxadrez.com.br/>. Acesso em 13/04/2014.

PORTAL STANFORD LOGIC GROUP. Disponível em <http://logic.stanford.edu/>.

RODRIGUES, Bernardo Araujo; CARLSON, Pedro. **Raspberry Pi + Webspeech API: Projetos em Sistemas Digitais**. 2013. 1 f. Monografia - Curso de Engenharia Elétrica ênfase em Eletrônica, Universidade de São Paulo, São Carlos, 2013.

ROSA, Aurelio de. **Introducing Web Speech API**. Disponível em: <<http://www.sitepoint.com/introducing-web-speech-api/>>. Acesso em: 13 abr. 2014.

RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. 3. ed. Upper Saddle River: Pearson Education, 2010.

Vazquez-Fernandez, E.; Coello, C.A.C. **An adaptive evolutionary algorithm based on tactical and positional chess problems to adjust the weights of a chess engine**, Evolutionary Computation (CEC), 2013 IEEE Congress

YEH, Ben. **CarlsenBot – Detailed Report (long post)**. Disponível em: <<http://benpyeh.com/2013/05/02/carlsenbot-detailed-report-long-post/>>. Acesso em: 13 abr. 2014.

## APÊNDICE A – index.html

```

<!DOCTYPE html>
<meta charset="utf-8">
<title>Webspeech</title>
<head>
  <link rel="stylesheet" href="estilo.css"
type="text/css"/>
  <script type="text/javascript"
src="https://code.jquery.com/jquery-1.7.2.min.js"></script>
  <script type="text/javascript"
src="iniciarTabuleiro.js"> </script>
  <script type="text/javascript"
src="movimentarPeca.js"> </script>
  <script type="text/javascript"
src="reconhecerVoz.js"> </script>
  <script type="text/javascript" src="aplicarIA.js">
</script>
</head>
<body>
<div id="pagina">
  <div id="title">
    <h1 id="headline"> Xadrez por
Reconhecimento de Voz </h1>
  </div>
  <div id="conteudo">
    <div id="tabuleiro"></div>
    <table id="escolha">
      <tr>
        <td><input
type="radio" name="cor" value="Brancas"
checked="checked">Brancas</td>
        <td><input
type="radio" name="cor" value="Pretas">Pretas</td>
      </tr>
      <tr>
        <td><input
type="radio" name="adv" value="Pessoa"
checked="checked" onclick='showMais(0);'>Pessoa</td>
        <td><input
type="radio" name="adv" value="Maquina"
onclick='showMais(1);'>Maquina</td>
      </tr>
      <tr id="bloco_alg">
        <td><input
type="radio" name="ia" value="AlfaBeta"
checked="checked">AlfaBeta</td>
        <td><input
type="radio" name="ia" value="Minimax">Minimax</td>
      </tr>
      <tr id="bloco_prof">
        <td>Profundidade</td>
        <td><select
id="profundidade">
          <option
value="um">1</option>
          <option
value="dois">2</option>
          <option
value="tres">3</option>
          <option
value="quatro">4</option>
        </select>
      </tr>
    </table>
    <button id="botaoConfirma"
onclick="showResult(1);">CONFIRMA</button>
    <div id="resultado">
      <p id='texto_escolha'></p>
      <button id="botaoTroca"
onclick="showResult(0);">REINICIAR</button>
      <div id="result_box">
        <div
id='move_manual'>
          <h3>
Movimento Manual</h3>
          <p>Identifique a peça, sua casa inicial e a casa
final de seu movimento. <span>Exemplo: peao e2
e4</span></p>
          <input
type="text" id="movimento" autocomplete="off">
          <button
id="botaomover"
onclick="movimentarPecaManual();">MOVER</button>
        </div>
      </div>
      <div
id="move_voz">
          <h3>Movimento por Reconhecimento de
Voz</h3>
          <p>Identifique a peça, sua casa inicial e a casa
final de seu movimento. <span>Exemplo: peao e2
e4</span></p>
          <div
id="info">
            <p id="msg_inicio">Seja bem vindo! Clique no
microfone para começar a falar.</p>
            <p id="msg_fale">Fale o seu comando.</p>
            <p id="msg_semcomando">Nenhum comando
detectado. Verifique suas configurações do microfone.</p>
            <p id="msg_semmicrofone"
style="display:none">Nenhum microfone encontrado.
Verifique que há um microfone instalado e que as
configurações do microfone estão corretas.</p>
            <p id="msg_permitir">Clique no botão "Permitir"
acima para habilitar seu microfone.</p>
            <p id="msg_negado">Permissão para acessar o
microfone foi negada.</p>
            <p id="msg_bloqueado">Permissão para usar o
microfone foi negada.</p>
            <p id="msg_atualizar">Esse navegador não
suporta a Web Speech API.</p>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

class="interim"></span>

        <button id="botaoiniciar"
onclick="botaoiniciar(event);">
                
                </button>
        </div>
        </div>
        </div>
        </div>
        </div>

<div id="rodape">
    <p>Trabalho de Conclusão
de Curso</p>
    <p>Pablo Henrique Alves de
Oliveira - 6920435</p>
</div>
</div>
</html>
<script>

$(document).keypress(function(e) {
    if ( e.which == 13 ) {
        movimentarPecaManual();
    }
});

mostrarMensagens('msg_inicio');

var final_transcript = "";
var reconhecimentoEmAndamento = false;
var ignore_onend;
var start_timestamp;

if (!('webkitSpeechRecognition' in window)) {
    upgrade();
}

else {
    botaoiniciar.style.display = 'inline-block';
    var reconhecimentoDeVoz = new
webkitSpeechRecognition();
    reconhecimentoDeVoz.lang = 'pt-BR';
    reconhecimentoDeVoz.continuous = true;
    reconhecimentoDeVoz.interimResults = true;

    reconhecimentoDeVoz.onstart = function() {
        reconhecimentoEmAndamento = true;
        mostrarMensagens('msg_fale');
        imagem_microfone.src = 'mic-
animate.gif';
    };

    reconhecimentoDeVoz.onerror = function(event)
{
        if (event.error == 'no-speech') {
            imagem_microfone.src =
'mic.gif';
            mostrarMensagens('msg_semcomando');
            ignore_onend = true;
        }
        if (event.error == 'audio-capture') {
            imagem_microfone.src =

```

```

mostrarMensagens('msg_semmicrofone');
            ignore_onend = true;
        }
        if (event.error == 'not-allowed') {
            if (event.timeStamp -
start_timestamp < 100) {
                mostrarMensagens('msg_bloqueado');
            }
            else {
                mostrarMensagens('msg_negado');
            }
            ignore_onend = true;
        }
    };

    reconhecimentoDeVoz.onend = function() {
        reconhecimentoEmAndamento = false;
        if (ignore_onend) {
            return;
        }
        imagem_microfone.src = 'mic.gif';
        if (!final_transcript) {
            mostrarMensagens('msg_inicio');
            return;
        }
        mostrarMensagens("");
        if (window.getSelection() {
            window.getSelection().removeAllRanges();
            var range =
document.createRange();
            range.selectNode(document.getElementById('final_span'));
            window.getSelection().addRange(range);
        }
    };

    reconhecimentoDeVoz.onresult =
function(event) {
        var interim_transcript = "";
        if (typeof(event.results) == 'undefined')
        {
            reconhecimentoDeVoz.onend = null;
            reconhecimentoDeVoz.onend();
            upgrade();
            return;
        }
        for (var i = event.resultIndex; i <
event.results.length; ++i) {
            if (event.results[i].isFinal) {
                final_transcript +=
event.results[i][0].transcript;
            }
            else {
                interim_transcript
+= event.results[i][0].transcript;
            }
        }
        final_transcript =
aplicarLetraMaiuscula(final_transcript);
        final_span.innerHTML =
pularLinha(final_transcript);
        interim_span.innerHTML =
pularLinha(interim_transcript);
    };
}

function showMais(mostrar){
    if (mostrar == 0){

```

```

        $("#bloco_prof").hide();
        $("#bloco_alg").hide();
    } else {
        $("#bloco_prof").show();
        $("#bloco_alg").show();
    }
}

function showResult(mostrar){
    if (mostrar == 0){
        $("#escolha").show();
        $("#botaoConfirma").show();
        $("#resultado").hide();
        $("#texto_escolha").hide();
        $("#botaoTroca").hide();
        $("#texto_escolha").html("");
        apagarTabuleiro();
    } else {
        $("#escolha").hide();
        $("#botaoConfirma").hide();
        $("#resultado").show();
        $("#texto_escolha").show();
        $("#botaoTroca").show();

        if($('#input:radio[name=adv]:checked').val() ==
"Maquina"){
            var e =
document.getElementById("profundidade");

            $("#texto_escolha").html($('#input:radio[name=cor
]:checked').val()+ " /
"+$('#input:radio[name=adv]:checked').val()+ " /
"+$('#input:radio[name=ia]:checked').val()+ " / profundidade
"+e.options[e.selectedIndex].text);
        }
        else{
            $("#texto_escolha").html($('#input:radio[name=cor
]:checked').val()+ " /
"+$('#input:radio[name=adv]:checked').val());
        }
        montaTabuleiro();
        profundidadeIA =
document.getElementById("profundidade").options[docum
ent.getElementById("profundidade").selectedIndex].text;
    }
    if ($('#input:radio[name=cor]:checked').val() ==
"Brancas"){
        vez = 'b';
    }
    else if ($('#input:radio[name=cor]:checked').val()
== "Pretas"){
        vez = 'b';
        if
        (($('input:radio[name=adv]:checked').val() == "Maquina")
&& $('input:radio[name=ia]:checked').val() == "AlfaBeta"){
            //var tempInicial = new
Date();
            var pontuacaoMaxIA =
alfaBeta(-999,999,profundidadeIA,vez);

            realizarMovimento(movimentoIA[0],
movimentoIA[1], movimentoIA[2], movimentoIA[3]);
            //alert("Movimento levou:
"+(new Date().getTime()-tempInicial)/1000+ " secs");
            //alert(contador-1);
            //alert(pontuacaoMaxIA);
            //contador = 0;
            //falarMovimento();
        }

        //var tempInicial = new Date();
        //var posicao = "";
        //(posicao,alfa,beta,cor,profundidade)
        //var pontuacaoMaxIA =
negascout(posicao,-999,999,vez,1);

        else if
        (($('input:radio[name=adv]:checked').val() == "Maquina")
&& $('input:radio[name=ia]:checked').val() == "Minimax"){
            //var tempInicial = new
Date();
            var pontuacaoMaxIA =
negamax(profundidadeIA,vez);

            realizarMovimento(movimentoIA[0],
movimentoIA[1], movimentoIA[2], movimentoIA[3]);
            //alert("Movimento levou:
"+(new Date().getTime()-tempInicial)/1000+ " secs");
            //alert(contador-1);
            //alert(pontuacaoMaxIA);
            //contador = 0;

            //falarMovimento(movimentoIAString);
        }
        vez = 'p';
    }
}
</script>
</body>

```

## APÊNDICE B – estilo.css

```

*{
    font-family: Verdana, Arial, sans-serif;
}

#pagina{
    width: 1000px;
    margin:0 auto;
}

#headline {
    text-align: center;
    color: #000;
    font: 42px 'Arial';
    font-weight: bold;
    padding: 5px;
    height: 40px;
}

#tabuleiro, #result_box{
    width: 450px;
    height: 600px;
    float: left;
}

#resultado, #bloco_prof, #bloco_alg, #texto_escolha,
#botaoTroca {
    display: none;
}

#bloco_prof td{
    padding-left: 45px;
}

#botaoConfirma, #botaoTroca{
    float: right;
    margin-top: 10px;
    margin-right:180px;
}

```

```

#tabuleiro{
    margin-left: 10px;
}

.casa{
    height: 50px;
    width: 50px;
    float: left;
    cursor: pointer;
}

.casa p{
    text-align: center;
}

.casa_branca{
    background-color: #fff;
}

.casa_preta{
    background-color: #ccc;
}

#result_box p{
    font-size: 10pt;
}

span{
    font-weight: bold;
    color: red;
}

input {
    width:100px;
}

#move_manual, #move_voz, #contagem_pecas{
    padding:15px;
}

#box{
    width: 100%;
    height: 50px;
    border: 1px solid black;
}

#rodape{
    text-align: right;
    font-size: 9pt;
    padding: 5px;
    clear: both;
}

#botaoiniciar {
    border: 0;
    background-color:transparent;
    padding: 0;
    float:right;
}

.final {
    color: black;
    padding-right: 3px;
}

.interim {
    color: gray;
}

.info {
    font-size: 14px;
    text-align: center;
    color: #777;
    display: none;
}

#info {
    font-size: 20px;
    text-align: center;
    color: #777;
}

```

## APÊNDICE C – reconhecerVoz.js

```

function upgrade() {
    botoainiciar.style.visibility = 'hidden';
    mostrarMensagens('msg_atualizar');
}

var two_line = /\n\n/g;
var one_line = /\n/g;
function pularLinha(s) {
    return s.replace(two_line,
'<p></p>').replace(one_line, '<br>');
}

var first_char = /\S/;
function aplicarLetraMaiuscula(s) {
    return s.replace(first_char, function(m) { return
m.toUpperCase(); });
}

function botoainiciar(event) {
    if (reconhecimentoEmAndamento) {
        reconhecimentoDeVoz.stop();
        movimentarPecaVoz(final_transcript);
        return;
    }

    final_transcript = "";
    reconhecimentoDeVoz.start();
    ignore_onend = false;
    final_span.innerHTML = "";
    interim_span.innerHTML = "";
    imagem_microfone.src = 'mic-slash.gif';
    mostrarMensagens('msg_permitir');
    start_timestamp = event.timeStamp;
}

function mostrarMensagens(s) {
    if (s) {
        for (var child = info.firstChild; child;
child = child.nextSibling) {
            if (child.style) {
                child.style.display
= child.id == s ? 'inline' : 'none';
            }
        }
        info.style.visibility = 'visible';
    }
    else {
        info.style.visibility = 'hidden';
    }
}

```



## APÊNDICE D – iniciarTabuleiro.js

```

var matriz = ["", "a", "b", "c", "d", "e", "f", "g", "h"];

function montaTabuleiro(){
    constroiTabuleiro();
    colocaPecasNaPosicaoInicial();
}

function constroiTabuleiro(){
    if
    ($('#input:radio[name=cor]:checked').val() == "Brancas"){
        for (var i = 0; i < 9; i++){
            $('##tabuleiro').append("<div id='linha'+i+'>");
            for (var j = 0; j < 9;
j++){
                if (i == 0
&& j > 0){
                    $('##linha'+i).append("<div id='"+i+"-"+j+"'
class='casa'></div>");
                } else if
(j == 0 && i > 0){
                    $('##linha'+i).append("<div id='"+i+"-"+j+"'
class='casa'></div>");
                } else {
                    var classe =
(i%2==0?(j%2==0?"casa_branca":"casa_preta");(j%2!=0?"
casa_branca":"casa_preta"));
                    $('##linha'+i).append("<div id='"+(9-i)+"-"+j+"'
class='casa "+classe+"'></div>");
                }
            }
            $('##tabuleiro').append("</div>");
        }
    } else{
        for (var i = 0; i < 9; i++){
            $('##tabuleiro').append("<div id='linha'+i+'>");
            for (var j = 0; j < 9;
j++){
                if (i == 0
&& j > 0){
                    $('##linha'+i).append("<div id='"+i+"-"+(9-j)+"'
class='casa'></div>");
                } else if
(j == 0 && i > 0){
                    $('##linha'+i).append("<div id='"+(9-i)+"-"+j+"'
class='casa'></div>");
                } else {
                    var classe =
(i%2==0?(j%2==0?"casa_branca":"casa_preta");(j%2!=0?"
casa_branca":"casa_preta"));
                    $('##linha'+i).append("<div id='"+i+"-"+(9-j)+"'
class='casa "+classe+"'></div>");
                }
            }
            $('##tabuleiro').append("</div>");
        }
    }
}

function colocaPecasNaPosicaoInicial(){
    //linhas e colunas
    for (var i = 1; i < 9; i++){
        $('#"#'+i+"-0").append("<div
class='casa'><p>"+(9-i)+"</p></div>");
    }
    for (var j = 1; j < 9; j++){
        $('#"#0-'+j).append("<div
class='casa'><p>"+matriz[j]+</p></div>");
    }
    //pretas
    $('#"#8-1").append("<img
src='p_torre.png'>");
    $('#"#8-2").append("<img
src='p_cavalo.png'>");
    $('#"#8-3").append("<img
src='p_bispo.png'>");
    $('#"#8-4").append("<img
src='p_dama.png'>");
    $('#"#8-5").append("<img
src='p_rei.png'>");
    $('#"#8-6").append("<img
src='p_bispo.png'>");
    $('#"#8-7").append("<img
src='p_cavalo.png'>");
    $('#"#8-8").append("<img
src='p_torre.png'>");
    for (var i = 1; i < 9; i++){
        $('#"#7-'+i).append("<img
src='p_peao.png'>");
    }
    //brancas
    $('#"#1-1").append("<img
src='b_torre.png'>");
    $('#"#1-2").append("<img
src='b_cavalo.png'>");
    $('#"#1-3").append("<img
src='b_bispo.png'>");
    $('#"#1-4").append("<img
src='b_dama.png'>");
    $('#"#1-5").append("<img
src='b_rei.png'>");
    $('#"#1-6").append("<img
src='b_bispo.png'>");
    $('#"#1-7").append("<img
src='b_cavalo.png'>");
    $('#"#1-8").append("<img
src='b_torre.png'>");
    for (var i = 1; i < 9; i++){
        $('#"#2-'+i).append("<img
src='b_peao.png'>");
    }
}

function apagarTabuleiro(){
    for (var i = 0; i < 9; i++){
        $('#"#linha'+i).remove();
        for (var j = 0; j < 9; j++){
            $('#"#'+i+"-
"+j).remove();
        }
    }
}

```

## APÊNDICE E – movimentarPeca.js

```

//Variáveis Globais

var pecasCapturadasBrancas = [];
var pecasCapturadasPretas = [];
var n_pecasCapturadasBrancas = 0;
var n_pecasCapturadasPretas = 0;
var linReiBranco = 1;
var colReiBranco = 5;
var linReiPreto = 8;
var colReiPreto = 5;
var vez;
var profundidadeIA;
var contador = 0;

function movimentarPecaManual(){
    var movimento =
document.getElementById("movimento").value;
    moverPeca(movimento);
}

function movimentarPecaVoz(transcript){
    moverPeca(transcript);
}

function movimentarPecaMouse(){
    /* 1 - Escolher a peça com mouse
    * 2 - Identifica as variáveis linhaAntes,
colunaAntes, linhaDepois, colunaDepois
    * 3 - Chama a função
realizarMovimento(linhaAntes, colunaAntes, linhaDepois,
colunaDepois);
    */
}

function moverPeca(stringMovimento){
    var jogada = stringMovimento.replace(/g,"");
    jogada = jogada.toLowerCase();

//Simplifica a string movimento
switch(jogada.substring(0,1)){
    case 'b'://Bispo
        jogada =
jogada.replace("bispo","B");
        break;
    case 'c': //Cavalo
        jogada =
jogada.replace("cavalo","C");
        break;
    case 'd': //Dama
        jogada =
jogada.replace("dama","D");
        break;
    case 'p': //Peão
        jogada =
jogada.replace("peao","P");
        jogada =
jogada.replace("peão","P");
        jogada =
jogada.replace("pião","P");
        break;
    case 'r': //Rei
        jogada =
jogada.replace("rei","R");
        break;
    case 't': //Torre
        jogada =
jogada.replace("torre","T");
        break;
}
    jogada = jogada.replace("00","e");

//
    var linhaAntes = 0, linhaDepois = 0;
    for (var i = 0; i < 9; i++){
        if (jogada.substring(1,2) == matrizz[i]){

```

```

        var colunaAntes = i;
    }
    if (jogada.substring(3,4) == matriz[i]){
        var colunaDepois = i;
    }
}

var linhaAntes = parseInt(jogada.substring(2,3));
var linhaDepois =
parseInt(jogada.substring(4,5));

if((colunaAntes > 0) && (colunaAntes < 9) &&
(colunaDepois > 0) && (colunaDepois < 9) && (linhaAntes
> 0) && (linhaAntes < 9) && (linhaDepois > 0) &&
(linhaDepois < 9)){
    if (testarMovimento(jogada,
linhaAntes, colunaAntes, linhaDepois, colunaDepois) ==
1){

//falarMovimento(stringMovimento);

        realizarMovimento(linhaAntes, colunaAntes,
linhaDepois, colunaDepois);

        if(vez == 'b'){
            vez = 'p';
        }
        else if(vez == 'p'){
            vez = 'b';
        }

        if
(($('input:radio[name=adv]:checked').val() == "Maquina")
&& $('input:radio[name=ia]:checked').val() == "AlfaBeta"){

//var tempolnicial =
new Date();

        var
pontuacaoMaxIA = alfaBeta(-999,999,profundidadeIA,vez);

        realizarMovimento(movimentoIA[0],
movimentoIA[1], movimentoIA[2], movimentoIA[3]);

//alert("Movimento
levou: "+(new Date().getTime()-tempolnicial)/1000+"
secs");

//alert(contador-1);

//alert(pontuacaoMaxIA);

//contador = 0;

//falarMovimento(movimentoIAString);

        if(vez == 'b'){
            vez = 'p';
        }
        else if(vez == 'p'){
            vez = 'b';
        }
    }
}

//var tempolnicial = new
Date();

//var posicao = ";

//(posicao,alfa,beta,cor,profundidade)

//var pontuacaoMaxIA =
negascout(posicao,-999,999,vez,1);

        else if
(($('input:radio[name=adv]:checked').val() == "Maquina")
&& $('input:radio[name=ia]:checked').val() == "Minimax"){

//var tempolnicial =
new Date();

        var
pontuacaoMaxIA = negamax(profundidadeIA,vez);

        realizarMovimento(movimentoIA[0],
movimentoIA[1], movimentoIA[2], movimentoIA[3]);

//alert("Movimento
levou: "+(new Date().getTime()-tempolnicial)/1000+"
secs");

//alert(contador-1);

//alert(pontuacaoMaxIA);

//contador = 0;

//falarMovimento(movimentoIAString);

        if(vez == 'b'){
            vez = 'p';
        }
        else if(vez == 'p'){
            vez = 'b';
        }
    }
}

```

```

        $("#"+linhaDepois+"-"+colunaDepois).html("");
    }
    else{
        alert ("Movimento inválido.
Tente novamente.");
        document.getElementById("movimento").value =
";
    }
}
else{
    alert ("Movimento inválido. Tente
novamente.");
    document.getElementById("movimento").value =
";
}
}

function
realizarMovimento(linhaAntes,colunaAntes,linhaDepois,col
unaDepois){
    //Pega peça na posição
    [linhaAntes,colunaAntes]

    var pecaID =
document.getElementById(linhaAntes+"-"+colunaAntes);

    var pecaTAG =
pecaID.getElementsByTagName('img')[0];

    var pecaSRC =
pecaTAG.src.replace("https://dl.dropboxusercontent.com/u
/77857832/TCC%20-%20Pablo/", "");

    if((pecaSRC.substring(2,3) == 'p' &&
pecaSRC.substring(0,1) == 'b' && linhaDepois == 8) ||
(pecaSRC.substring(2,3) == 'p' &&
pecaSRC.substring(0,1) == 'p' && linhaDepois == 1)){

        promoverPeao(linhaAntes,colunaAntes,pecaSR
C.substring(0,1));

        pecaID =
document.getElementById(linhaAntes+"-"+colunaAntes);

        pecaTAG =
pecaID.getElementsByTagName('img')[0];

        pecaSRC =
pecaTAG.src.replace("https://dl.dropboxusercontent.com/u
/77857832/TCC%20-%20Pablo/", "");
    }

    $("#"+linhaDepois+"-"+colunaDepois).html("");

    //Apaga o que tem na nova
casa

    $("#"+linhaDepois+"-
"+colunaDepois).append("");
    //Insere peça na nova casa

    $("#"+linhaAntes+"-"+colunaAntes).html("");

    //Deleta peça da
casa anterior
}

function
desfazerMovimento(pecaAntes,pecaDepois,linhaAntes,col
unaAntes,linhaDepois,colunaDepois){

    //Apaga ambas as casas

    $("#"+linhaAntes+"-"+colunaAntes).html("");

    $("#"+linhaDepois+"-"+colunaDepois).html("");

    //Volta as peças pro lugar

    $("#"+linhaAntes+"-
"+colunaAntes).append("");

    if(pecaDepois != "vazio"){

        $("#"+linhaDepois+"-
"+colunaDepois).append("");

    }

    //Se for rei arruma as variaveis linReiBranco e
colReiBranco ou linReiPreto e colReiPreto

    if (pecaAntes == 'b_rei.png'){

        linReiBranco = linhaAntes;

        colReiBranco = colunaAntes;

    }

    if (pecaAntes == 'p_rei.png'){

        linReiPreto = linhaAntes;

        colReiPreto = colunaAntes;

    }

}

function falarMovimento(stringMovimento){

    var msg = new
SpeechSynthesisUtterance(stringMovimento);

    msg.lang = 'pt-BR';

    msg.rate = '0.7';

    window.speechSynthesis.speak(msg);
}

```

```

}

function
testarMovimento(palavra,linAnt,colAnt,linDep,colDep){

    //Pega peça na posição [linAnt,colAnt] - Antes

    var pecaAntID =
document.getElementByld(linAnt+"-"+colAnt);

    //Verifica se há alguma peça na casa de origem

    if(pecaAntID.getElementsByTagName('img').len
gth > 0){

        var pecaAntTAG =
pecaAntID.getElementsByTagName('img')[0];

        var pecaAntSRC =
pecaAntTAG.src.replace("https://dl.dropboxusercontent.co
m/u/77857832/TCC%20-%20Pablo/", "");

    }

    else{

        return 0;

    }

    //Verifica de quem é a vez

    if(pecaAntSRC.substring(0,1) != vez){

        return 0;

    }

    //Pega peça na posição [linDep,colDep] - Depois

    var pecaDepID =
document.getElementByld(linDep+"-"+colDep);

    //Verifica se há alguma peça na casa de destino

    if(pecaDepID.getElementsByTagName('img').len
gth > 0){

        var pecaDepTAG =
pecaDepID.getElementsByTagName('img')[0];

        var pecaDepSRC =
pecaDepTAG.src.replace("https://dl.dropboxusercontent.c
om/u/77857832/TCC%20-%20Pablo/", "");

    }

    else{

        var pecaDepSRC = "vazio";

    }

    //Verifica se há uma peça da mesma cor na casa
de destino

    if(pecaDepSRC.substring(0,1) ==
pecaAntSRC.substring(0,1)){

        return 0;

    }

    //Módulo da diferença entre as linhas ou colunas

    var modLin = Math.abs(linDep - linAnt); // Linhas

    var modCol = Math.abs(colDep - colAnt); //
Colunas

    //Testa o movimento dependendo da peça

    var movimento = 0;

    switch (palavra.substring(0,1)){

        case 'B':

            movimento =
testarMovimentoBispo(linAnt,linDep,colAnt,colDep,pecaAn
tSRC,modLin,modCol);

            break;

        case 'C':

            movimento =
testarMovimentoCavalo(linAnt,linDep,colAnt,colDep,pecaA
ntSRC,modLin,modCol);

            break;

        case 'D':

            movimento =
testarMovimentoDama(linAnt,linDep,colAnt,colDep,pecaAn
tSRC,modLin,modCol);

            break;

        case 'P':

            movimento =
testarMovimentoPeao(linAnt,linDep,colAnt,colDep,pecaAnt
SRC,pecaDepSRC,modLin,modCol);

            break;

        case 'R':

            movimento =
testarMovimentoRei(linAnt,linDep,colAnt,colDep,pecaAntS
RC,modLin,modCol);

            break;

        case 'T':

```

```

        movimento =
testarMovimentoTorre(linAnt,linDep,colAnt,colDep,pecaAnt
SRC,modLin,modCol);

        break;

        default:

        break;

    }

    var testeDeMovimento;

    if(movimento == 1){

        realizarMovimento(linAnt,colAnt,linDep,colDep);

        testeDeMovimento =
testarXeque(pecaAntSRC.substring(0,1));

        desfazerMovimento(pecaAntSRC,pecaDepSRC,
linAnt,colAnt,linDep,colDep);

        return testeDeMovimento;

    }

    return 0;
}

function
testarMovimentoBispo(linAnt,linDep,colAnt,colDep,pecaAn
tSRC,modLin,modCol){

    if(pecaAntSRC.substring(2,7) == 'bispo'){

        if (modLin == modCol){

            return
verificarPresencaDePecasNoCaminho(linAnt, linDep,
colAnt, colDep, modLin, modCol);

        }

    }

    return 0;
}

function
testarMovimentoCavalo(linAnt,linDep,colAnt,colDep,pecaA
ntSRC,modLin,modCol){

    if(pecaAntSRC.substring(2,8) == 'cavalo'){

        if (((modLin == 1) && (modCol == 2)) ||
((modLin == 2) && (modCol == 1))){

            return 1;

        }

    }

}

        return 0;
    }

function
testarMovimentoDama(linAnt,linDep,colAnt,colDep,pecaAn
tSRC,modLin,modCol){

    if(pecaAntSRC.substring(2,6) == 'dama'){

        if ((modLin == modCol) || (linDep ==
linAnt) || (colDep == colAnt)){

            return
verificarPresencaDePecasNoCaminho(linAnt,linDep,colAnt
,colDep,modLin,modCol);

        }

    }

    return 0;
}

function
testarMovimentoPeao(linAnt,linDep,colAnt,colDep,pecaAnt
SRC,pecaDepSRC,modLin,modCol){

    //Peao Branco

    if(pecaAntSRC == 'b_peao.png'){

        if((modLin == 1) && (linDep > linAnt)
&& (colDep == colAnt) && (pecaDepSRC == 'vazio')){

            return 1;

        }

        else if((modLin == 2) && (linDep >
linAnt) && (colDep == colAnt) && (linAnt == 2) &&
(pecaDepSRC == 'vazio') &&
(document.getElementById("3-
"+colAnt).getElementsByTagName('img').length == 0)){

            return 1;

        }

        else if((modLin == 1) && (linDep >
linAnt) && (modCol == 1) && (pecaDepSRC != 'vazio')){

            return 1;

        }

    }

}

    //Peao Preto

    else if(pecaAntSRC == 'p_peao.png'){

        if((modLin == 1) && (linDep < linAnt)
&& (colDep == colAnt) && (pecaDepSRC == 'vazio')){

            return 1;

        }

    }

}

```

```

                else if((modLin == 2) && (linDep <
linAnt) && (colDep == colAnt) && (linAnt == 7) &&
(pecaDepSRC == 'vazio') &&
(document.getElementById("6-
"+colAnt).getElementsByTagName('img').length == 0)){
                    return 1;
                }
                else if((modLin == 1) && (linDep <
linAnt) && (modCol == 1) && (pecaDepSRC != 'vazio')){
                    return 1;
                }
            }
        }
        return 0;
    }
}

```

```

function
testarMovimentoRei(linAnt,linDep,colAnt,colDep,pecaAntSRC,modLin,modCol){
    if(pecaAntSRC.substring(2,5) == 'rei'){
        if ((modLin == 1) && ((modCol == 1) ||
(colDep == colAnt))){
            if(pecaAntSRC.substring(0,1) == 'b'){
                linReiBranco =
linDep;
                colReiBranco =
colDep;
            }
            else
if(pecaAntSRC.substring(0,1) == 'p'){
                linReiPreto =
linDep;
                colReiPreto =
colDep;
            }
            return 1;
        }
        else if ((linDep == linAnt) && (modCol
== 1)){
            if(pecaAntSRC.substring(0,1) == 'b'){
                linReiBranco =
linDep;
                colReiBranco =
colDep;
            }
        }
    }
}

```

```

                else
if(pecaAntSRC.substring(0,1) == 'p'){
                    linReiPreto =
linDep;
                    colReiPreto =
colDep;
                }
                return 1;
            }
        }
        return 0;
    }
}

function
testarMovimentoTorre(linAnt,linDep,colAnt,colDep,pecaAnt
SRC,modLin,modCol){
    if(pecaAntSRC.substring(2,7) == 'torre'){
        if ((linDep == linAnt) || (colDep ==
colAnt)){
            return
verificarPresencaDePecasNoCaminho(linAnt, linDep,
colAnt, colDep, modLin, modCol);
        }
    }
    return 0;
}

function
verificarPresencaDePecasNoCaminho(linAnt,linDep,colAnt
,colDep,modLin,modCol){
    //Passo pare teste de peças no caminho
    var passoLin = ((linDep != linAnt)?(linDep -
linAnt)/modLin:0);
    var passoCol = ((colDep != colAnt)?(colDep -
colAnt)/modCol:0);
    //Variáveis guardando linhas e colunas testadas
    var linTeste = linAnt + passoLin;
    var colTeste = colAnt + passoCol;
    while (!(linTeste == linDep) && (colTeste ==
colDep)){
        if
(document.getElementById(linTeste+"-
"+colTeste).getElementsByTagName('img').length > 0){
            return 0;
        }
    }
}

```





```

        if(document.getElementById(linha+"-
+i).getElementsByTagName('img').length != 0){

            if(document.getElementById(linha+"-
+i).getElementsByTagName('img')[0].src.replace("https://d
l.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == (corOponente+"_torre.png")){

                if
(verificarPresencaDePecasNoCaminho(linha,linha,coluna,i
,0,Math.abs(coluna-i))){

                    return 0;

                }

            }

            else
if(document.getElementById(linha+"-
+i).getElementsByTagName('img')[0].src.replace("https://d
l.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == (corOponente+"_dama.png")){

                if
(verificarPresencaDePecasNoCaminho(linha,linha,coluna,i
,0,Math.abs(coluna-i))){

                    return 0;

                }

            }

        }

    }

    //Testa se Bispo ou Dama dão Xequê -
Diagonais

    for (var i = -7; i < 8; i++){

        if((linha+i > 0) && (linha+i < 9) &&
(coluna+i > 0) && (coluna+i < 9) &&
(document.getElementById((linha+i)+"-
+(coluna+i)).getElementsByTagName('img').length != 0)){

            if(document.getElementById((linha+i)+"-
+(coluna+i)).getElementsByTagName('img')[0].src.replace
("https://dl.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == (corOponente+"_bispo.png")){

                if
(verificarPresencaDePecasNoCaminho(linha,(linha+i),colu
na,(coluna+i),Math.abs(i),Math.abs(i))){

                    return 0;

                }

            }

            else
if(document.getElementById((linha+i)+"-
+(coluna+i)).getElementsByTagName('img')[0].src.replace
("https://dl.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == (corOponente+"_dama.png")){

```

```

        if
(verificarPresencaDePecasNoCaminho(linha,(linha+i),colu
na,(coluna+i),Math.abs(i),Math.abs(i))){

            return 0;

        }

    }

    if((linha+i > 0) && (linha+i < 9) &&
(coluna-i > 0) && (coluna-i < 9) &&
(document.getElementById((linha+i)+"-
+(coluna-
i)).getElementsByTagName('img').length != 0)){

        if(document.getElementById((linha+i)+"-
+(coluna-
i)).getElementsByTagName('img')[0].src.replace("https://dl.
dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == (corOponente+"_bispo.png")){

            if
(verificarPresencaDePecasNoCaminho(linha,(linha+i),colu
na,(coluna-i),Math.abs(i),Math.abs(i))){

                return 0;

            }

        }

        else
if(document.getElementById((linha+i)+"-
+(coluna-
i)).getElementsByTagName('img')[0].src.replace("https://dl.
dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == (corOponente+"_dama.png")){

            if
(verificarPresencaDePecasNoCaminho(linha,(linha+i),colu
na,(coluna-i),Math.abs(i),Math.abs(i))){

                return 0;

            }

        }

    }

    //Testa se peão da Xequê

    if(cor == 'b'){

        if(linha+1 < 9){

            if((coluna-1 > 0) &&
(document.getElementById((linha+1)+"-
+(coluna-
1)).getElementsByTagName('img').length != 0)){

                if(document.getElementById((linha+1)+"-
+(coluna-
1)).getElementsByTagName('img')[0].src.replace("https://dl.
dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == "p_peao.png"){

                    return 0;

```

```

    }
}

else if((coluna+1 < 9) &&
(document.getElementById((linha+1)+"-
"+(coluna+1)).getElementsByTagName("img").length != 0)){

    if(document.getElementById((linha+1)+"-
"+(coluna+1)).getElementsByTagName("img")[0].src.replac
e("https://dl.dropboxusercontent.com/u/77857832/TCC%2
0-%20Pablo/", "") == "p_peao.png"){

        return 0;

    }

}

}

}

else if(cor == 'p'){

    if(linha-1 > 0){

        if((coluna-1 > 0) &&
(document.getElementById((linha-1)+"-"+(coluna-
1)).getElementsByTagName("img").length != 0)){

            if(document.getElementById((linha-1)+"-
"+(coluna-
1)).getElementsByTagName("img")[0].src.replace("https://dl
.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == "b_peao.png"){

                return 0;

            }

        }

    }

}

else if((coluna+1 < 9) &&
(document.getElementById((linha-1)+"-
"+(coluna+1)).getElementsByTagName("img").length != 0)){

    if(document.getElementById((linha-1)+"-
"+(coluna+1)).getElementsByTagName("img")[0].src.replac
e("https://dl.dropboxusercontent.com/u/77857832/TCC%2
0-%20Pablo/", "") == "b_peao.png"){

        return 0;

    }

}

}

//Testa se está do lado de outro rei

for (var i = -1; i < 2; i++){

    for (var j = -1; j < 2; j++){

        if((linha+i > 0) && (linha+i <
9) && (coluna+j > 0) && (coluna+j < 9) &&
(document.getElementById((linha+i)+"-
"+(coluna+j)).getElementsByTagName("img").length != 0)){

            if(document.getElementById((linha+i)+"-
"+(coluna+j)).getElementsByTagName("img")[0].src.replace
("https://dl.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "") == (corOponente+"_rei.png")){

                return 0;

            }

        }

    }

}

//Rei não fica em xeque

return 1;

}

```

## APÊNDICE F – aplicarIA.js

```

var movimentolA = ["0", "0", "0", "0", "0"];
//var movimentolAString = "";

function alfaBeta(alfa,beta,profundidade,cor){
    contador++;
    var pontuacao;
    if (profundidade == 0){
        if (cor == 'b'){
            return -pontuarPosicao();
        }
        else if (cor == 'p'){
            return pontuarPosicao();
        }
    }

    if(cor == 'b'){
        var corOponente = 'p';
    }
    else if(cor == 'p'){
        var corOponente = 'b';
    }

    for(var i = 1; i < 9; i++){
        for(var j = 1; j < 9; j++){

            if(document.getElementById(i+"-
"+j).getElementsByTagName('img').length > 0){
                var pecaAntes =
document.getElementById(i+"-
"+j).getElementsByTagName('img')[0].src.replace("https://d
l.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "");

                if(pecaAntes.substring(0,1) != cor){
                    continue;
                }
                else{
                    continue;
                }
                for(var k = 1; k < 9; k++){
                    for(var l = 1; l < 9;
l++){

                        if(document.getElementById(k+"-
"+l).getElementsByTagName('img').length > 0){

                            var pecaDepois =
document.getElementById(k+"-
"+l).getElementsByTagName('img')[0].src.replace("https://d
l.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "");

                            if(pecaDepois.substring(0,1) == cor){
                                continue;
                            }
                        }
                    }
                }
                var pecaDepois = "vazio";
                var
jogada =
pecaAntes.substring(2,3).toUpperCase()+"+matriz[j]"+i+"
+matriz[l]"+k;
                if
(testarMovimento(jogada, i, j, k, l) == 1){
                    realizarMovimento(i, j, k, l);
                    vez = corOponente;
                    var pontuacao = -alfaBeta(-beta,-
alfa,(profundidade-1),corOponente);
                    vez = cor;
                    if(pontuacao >= beta){
                        desfazerMovimento(pecaAntes,pecaDepois,i,j,k,l
);
                        return beta;
                    }
                    if(pontuacao > alfa){
                        alfa = pontuacao;
                        if(profundidade == profundidadeIA){
                            movimentolA[0] = i;
                            movimentolA[1] = j;
                            movimentolA[2] = k;
                            movimentolA[3] = l;
                        }
                    }
                    desfazerMovimento(pecaAntes,pecaDepois,i,j,k,l
);
                }
            }
        }
    }
    return alfa;
}

function negamax(profundidade,cor){
    contador++;
    var max = -999;
    var pontuacao;
    if (profundidade == 0){
        if (cor == 'b'){
            return -pontuarPosicao();
        }
        else if (cor == 'p'){
            return pontuarPosicao();
        }
    }
    alfa = max;
    if(cor == 'b'){
        var corOponente = 'p';
    }
    else if(cor == 'p'){
        var corOponente = 'b';
    }
    for(var i = 1; i < 9; i++){

```

```

        for(var j = 1; j < 9; j++){
            if(document.getElementById(i+"-
"+j).getElementsByTagName('img').length > 0){
                var pecaAntes =
document.getElementById(i+"-
"+j).getElementsByTagName('img')[0].src.replace("https://d
l.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "");

                if(pecaAntes.substring(0,1) != cor){
                    continue;
                }
                else{
                    continue;
                }
                for(var k = 1; k < 9; k++){
                    for(var l = 1; l < 9;
l++){
                        if(document.getElementById(k+"-
"+l).getElementsByTagName('img').length > 0){
                            var pecaDepois =
document.getElementById(k+"-
"+l).getElementsByTagName('img')[0].src.replace("https://d
l.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "");

                            if(pecaDepois.substring(0,1) == cor){
                                continue;
                            }
                        }
                        else{
                            var pecaDepois = "vazio";
                        }
                        var
jogada =
pecaAntes.substring(2,3).toUpperCase()+"matriz[j]"+i+"
+matriz[l]"+k;
                        if
(testarMovimento(jogada, i, j, k, l) == 1){
                            realizarMovimento(i, j, k, l);
                            if(pecaAntes.substring(0,1) == 'b'){
                                vez = corOponente;
                                var pontuacao = -
negamax((profundidade-1), corOponente);
                                vez = cor;
                            }
                            else if(pecaAntes.substring(0,1) == 'p'){
                                vez = corOponente;
                                var pontuacao = -
negamax((profundidade-1), corOponente);
                                vez = cor;
                            }
                        }
                        if(pontuacao >= max){
                            max = pontuacao;
                        }
                    }
                }
            }
        }
    }
    return max;
}

function pontuarPosicao(){
    //Numero pecas brancas
    var n_bisposB = 0;
    var n_cavalosB = 0;
    var n_damasB = 0;
    var n_peoesB = 0;
    var n_reisB = 0;
    var n_torresB = 0;
    //Numero pecas pretas
    var n_bisposP = 0;
    var n_cavalosP = 0;
    var n_damasP = 0;
    var n_peoesP = 0;
    var n_reisP = 0;
    var n_torresP = 0;
    //Peso das pecas
    var peso_bispo = 3;
    var peso_cavalo = 3;
    var peso_dama = 9;
    var peso_peao = 1;
    var peso_rei = 100;
    var peso_torre = 5;
    //Mobilidade
    var peso_mobilidade = 0.1;
    var n_movPossiveisB = 0;
    var n_movPossiveisP = 0;
    for(var i = 1; i < 9; i++){
        for(var j = 1; j < 9; j++){
            if(document.getElementById(i+"-
"+j).getElementsByTagName('img').length > 0){
                var peca =
document.getElementById(i+"-
"+j).getElementsByTagName('img')[0].src.replace("https://d
l.dropboxusercontent.com/u/77857832/TCC%20-
%20Pablo/", "");
            }
            else{
                var peca = "vazio";
            }
            //Determina que tipo de peça
            switch (peca.substring(0,3)){
                case 'b_b':
                    n_bisposB++;
                    break;
                case 'b_c':
                    n_cavalosB++;
                    break;
            }
        }
    }
}

```

```

        case 'b_d':
            n_damasB++;
            break;
        case 'b_p':
            n_peoesB++;
            break;
        case 'b_r':
            n_reisB++;
            break;
        case 'b_t':
            n_torresB++;
            break;
        case 'p_b':
            n_bisposP++;
            break;
        case 'p_c':
            n_cavalosP++;
            break;
        case 'p_d':
            n_damasP++;
            break;
        case 'p_p':
            n_peoesP++;
            break;
        case 'p_r':
            n_reisP++;
            break;
        case 'p_t':
            n_torresP++;
            break;
        default:
            break;
    }
    for(var k = 1; k < 9; k++){
        for(var l = 1; l < 9;
l++){
            var
jogada =
peca.substring(2,3).toUpperCase()+"matriz[j]"+i+"matriz
[l]"+k;

            //Determina quantos movimentos possíveis ela
pode ter
            if
((peca.substring(0,1) == 'b') && (testarMovimento(jogada,
i, j, k, l) == 1)){
                n_movPossiveisB++;
            }
            else if
((peca.substring(0,1) == 'p') && (testarMovimento(jogada,
i, j, k, l) == 1)){
                n_movPossiveisP++;
            }
        }
    }
}
var pontuacaoMaterial = peso_bispo*(n_bisposP
- n_bisposB) + peso_cavalo*(n_cavalosP - n_cavalosB) +
peso_dama*(n_damasP - n_damasB) +
peso_peao*(n_peoesP - n_peoesB) + peso_rei*(n_reisP -
n_reisB) + peso_torre*(n_torresP - n_torresB);
var pontuacaoMobilidade =
peso_mobilidade*(n_movPossiveisP - n_movPossiveisB);
        var pontuacaoTotal = pontuacaoMaterial +
pontuacaoMobilidade;
        return pontuacaoTotal;
    }

```