

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Programa de Pós-Graduação em Ciência da Computação

Uma Abordagem de Monitoramento dos Sinais  
Motores da Doença de Parkinson Baseada em Jogos  
Eletrônicos

Leonardo Melo de Medeiros

Tese submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação  
Linha de Pesquisa: Engenharia de Software

Leandro Dias da Silva (Orientador)  
Hyggo Oliveira de Almeida (Orientador)

Campina Grande, Paraíba, Brasil  
©Leonardo Melo de Medeiros, Julho - 2016

## Resumo

Os Sistemas de Monitoramento da Saúde (SMS) possibilitam aos médicos obterem informações sobre o estado de saúde de seus pacientes. Além disso, a identificação dos sintomas das doenças podem auxiliar no diagnóstico precoce e prevenir a ocorrência de situações críticas.

Para acompanhar e avaliar a saúde motora de um paciente, é necessário realizar uma avaliação motora por meio de movimentos específicos. Isto dificulta a concepção de um SMS de dados motores não-invasivo e engajados na rotina diária dos pacientes. A abordagem apresentada nesta tese, utiliza os jogos eletrônicos como fator motivacional para o fornecimento dos dados motores. Durante o jogo, o usuário é induzido a executar movimentos relevantes, de modo que um sensor de movimento possa adquiri-los e quantificá-los. Este ambiente lúdico, de jogo eletrônico, abstrai o usuário do contexto de tratamento da saúde e incentiva a execução dos movimentos de um maneira mais natural do que a imposta por um exame clínico.

Para avaliar esta abordagem, foi desenvolvido um jogo com a arquitetura proposta para identificar sintomas motores relacionadas com a Doença de Parkinson. Num estudo de caso-controle, foram avaliados os movimentos angulares dos braços para quantificar as habilidades motoras desses grupos. Os dados coletados foram processados e aplicados a uma Máquina de Vetor de Suporte (SVM) para classificar a ocorrência do sintoma da bradicinesia do Parkinson. Obteve-se uma classificação com uma acurácia de 86,67% e falsos positivos de 6,67%. Além disso, em uma experimento para avaliação da aceitação dos usuários, 90% ficaram motivados com o jogo desenvolvido e afirmaram que integrariam o SMS em sua rotina diária. Estes resultados demonstram que a abordagem de monitoramento baseado em jogos, apresentada nesta tese, tem potencial para ser um SMS para monitoramentos dos sintomas motores.

## **Abstract**

Health Monitoring Systems (HMS) allow doctors to gain a better picture of their patient's health status. An early identification of symptoms can support the disease's diagnostic and prevent critical situations.

In order to monitor a patient's motor abilities, it is necessary to record and evaluate specific movements. This makes it difficult to design a HMS that non-obtrusively integrates into the patient's daily routine. The approach presented in this thesis makes use of the motivational power of electronic games. While playing the game, the user is incited to make the relevant movements, so that an optical sensor can detect and measure them. The playful situation distracts the user from thinking about health issues and therefore encourages more natural movements with improved validity for a health examination.

To evaluate this approach, a game has been developed and employed to detect Parkinson related motor symptoms. In a study with patients diagnosed as affected by the Parkinson Disease and a healthy control group, the angular movements of the arms were used to measure motor abilities. The data was then processed and applied to a Support Vector Machine (SVM) to predict, based on the detected movements, whether a subject shows Parkinson related symptoms or should be classified as healthy. The system classified the subjects with an accuracy of 86.67% and a rate of 6.67% false positives. Furthermore, the user acceptance of the game-based approach was studied and showed that 90% of the users felt motivated to play the game as part of their daily routine. These results demonstrate that the game-based approach presented in this thesis has the potential to become a base for HMS that monitor motor symptoms.

# Conteúdo

<b>1</b>	<b>Avaliação Experimental</b>	<b>1</b>
1.1	Entrevista Semiestruturada com Profissionais de Saúde . . . . .	1
1.1.1	Objetivo da Entrevista Semiestruturada . . . . .	2
1.1.2	Perfil dos Participantes . . . . .	2
1.1.3	Análise . . . . .	3
1.1.4	Requisitos Identificados . . . . .	9
1.1.5	Considerações Finais Sobre a Entrevista SemiEstruturada . . . . .	11
1.2	Máquina de Vetor de Suporte para Estudo Analítico de Caso Controle Por Intermédio de Sensor de Movimento Usado em Jogos Eletrônicos . . . . .	13
1.2.1	Estudo analítico de caso-controle . . . . .	13
1.2.2	Aplicação do Método . . . . .	17
1.2.3	Resultados . . . . .	17
1.2.4	Aprendizagem de Máquina (SVM) . . . . .	18
1.3	Avaliação Da Aceitação Da Abordagem Junto aos Pacientes com Parkinson Utilizando <i>Goal Question Metric</i> . . . . .	23
1.3.1	Aplicação do Método . . . . .	24
1.3.2	Resultados . . . . .	25

# Lista de Símbolos

**Parkinson** Doença de Parkinson

**GQM** *Goal-Question-Metric*

**OMS** Organização Mundial de Saúde

**SMS** Sistemas de Monitoramento da Saúde

**SVM** Máquina de Vetor de Suporte

**JOGUE-ME** Jogo com Monitoramento de Saúde Embutido

# Lista de Figuras

1.1	Teste de um jogo usando acelerômetro para quantificação do sinal de tremor do Parkinson . . . . .	10
1.2	Movimentos de abdução e adução . . . . .	15
1.3	Exemplo do gráfico dos ângulos de adução e abdução dos braços em função do tempo . . . . .	16
1.4	Vetor médio do movimento de abdução e adução . . . . .	18
1.5	<i>Grid-search</i> - acurácia da classificação . . . . .	20
1.6	<i>Grid-search</i> - <i>fpRate</i> . . . . .	21
1.7	O paradigma GQM . . . . .	24
1.8	Resultado da pergunta 1 . . . . .	26
1.9	Resultado da pergunta 3 . . . . .	27
1.10	Resultado da pergunta 10 . . . . .	27

# Lista de Tabelas

1.1	Perfil dos participantes . . . . .	3
1.2	Matriz de Rastreabilidade: fragmento x requisitos . . . . .	12
1.3	Requisitos implementados . . . . .	12
1.4	Descrição do vetor de características extraído da coleta de dados. . . . .	16
1.5	Descrição da matriz de confusão . . . . .	19
1.6	Resultado da matriz de confusão usando SVM . . . . .	20
1.7	Média da amplitude do movimento de abdução do braço . . . . .	22
1.8	Métricas da matriz de confusão . . . . .	23
1.9	Métricas avaliadas pelo <i>GQM</i> . . . . .	26

# **Lista de Códigos Fonte**



# Capítulo 1

## Abordagem *JOGUE-ME*

Para desenvolver um Sistema de Monitoramento da Saúde (SMS) dos sinais motores, usando jogos eletrônicos como interface de entrada de dados, é necessário analisar que movimentos e ações o usuário deve executar para que seja possível identificar os sinais motores, a partir de suas ações. Estes movimentos devem ser testados junto a indivíduos portadores da deficiência a ser monitorada e indivíduos como grupo de controle para avaliar a viabilidade de detecção do sinal.

### 1.1 Definição de Requisitos da Solução

Com base no levantamento bibliográfico e nas entrevistas semiestruturadas [6] com profissionais de saúde, identificamos e enumeramos os seguintes requisitos funcionais, os quais devem ser desenvolvidos para uma solução *Jogo com Monitoramento de Saúde Embutido (JOGUE-ME)*:

**REQ-JOGUE-ME-01 - Pontuação e Taxa de Acerto:** O jogador percebe os objetivos e visualiza o sucesso ou o fracasso alcançado. O jogo pontua o jogador de acordo com seus erros e acertos [15; 14].

**REQ-JOGUE-ME-02 - Progresso e Evolução do Jogador e dos Desafios:** O jogador percebe seu progresso e sua evolução no jogo. Os desafios tornam-se mais complexos no decorrer do tempo [15].

**REQ-JOGUE-ME-03 - Estado de Fluxo :** Um dos grandes desafios de um jogo eletrônico é levar o usuário a um “Estado de Fluxo” ou escapismo, passando a executar a atividade proposta pelo jogo de uma forma autotélica, ou seja, o usuário não vislumbra um benefício imediato ou futuro [16].

**REQ-JOGUE-ME-04 - Preocupação com Integridade Física do Jogador:** Promover atividades físicas ou ações que venham a trazer injúria ao jogador,

como: movimentos de equilíbrio, movimentos repetitivos ou bruscos [2; 14].

**REQ-JOGUE-ME-05 - Aquisição e Armazenamento de Sinais Motores:** O jogo deve realizar a aquisição dos sinais motores do usuário usando sensores de movimento. Os dados capturados são enviados a um servidor para tornar possível o acompanhamento da saúde motora.

**REQ-JOGUE-ME-06 - Mecanismo de Identificação de Sinais Motores:** Baseados em algoritmos de aprendizagem de máquina, o servidor acompanha todos os usuários do sistema e identifica qual deles está com distúrbio motor; em caso afirmativo, envia-se a informação ao profissional de saúde.

**REQ-JOGUE-ME-07 - Mecanismo de Visualização dos Parâmetros Motores do Usuário:** O profissional de saúde poderá visualizar os dados identificados pela máquina de aprendizagem, para realizar a tomada de decisão sobre o estado de saúde do usuário.

## 1.2 Visão geral da solução

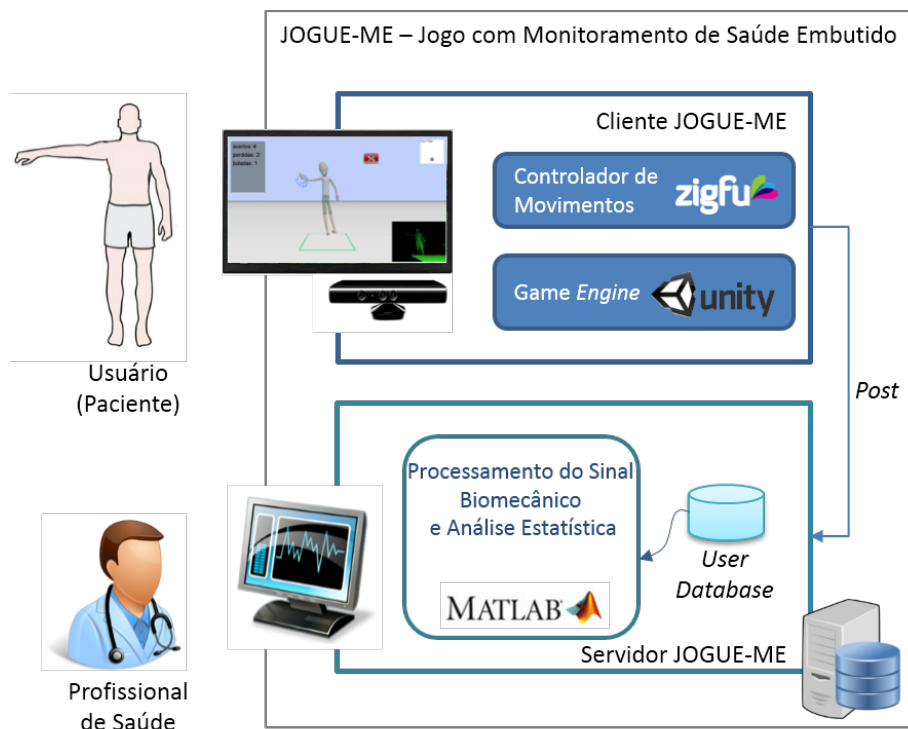


Figura 1.1: Visão geral da abordagem JOGUE-ME

A abordagem JOGUE-ME faz uso de jogos eletrônicos como interface de aquisição de sinais, tornando os usuários mais motivados a fornecer seus sinais motores, em comparação

ao uso dos dispositivos vestíveis [1]. Então, com o uso da presente abordagem, um paciente portador de uma doença motora, no conforto do seu lar, poderá fornecer sinais motores de uma maneira colaborativa e não invasiva. Por outro lado, o profissional de saúde poderá visualizar os sinais motores de seus pacientes com uma frequência muito maior, em comparação às avaliações clínicas realizadas durante o período de consulta.

Nesse estudo, ao utilizar jogos eletrônicos como mecanismo para entrada de dados, é possível alcançar os requisitos de não invasividade identificados nesta tese, pois, através dos dispositivos de sensores de movimento usados nesses ambientes, é possível desenvolver um jogo que motive o usuário a executar ações específicas para permitir o monitoramento de sinais motores. A partir de uma interface com o usuário, que permite enviar os dados capturados a um servidor, e armazenar estes dados para o acompanhamento da saúde motora por parte do profissional de saúde.

Para esta solução, propõe-se usar técnicas de processamento de sinais para reconhecer os padrões de movimento e identificar os sinais motores (Figura ??). Para tornar isso factível é necessário identificar ciclos de movimento, filtrá-los e extrair características deste movimento. Após a extração das características, os dados são repassados para máquinas de aprendizagem, as quais são responsáveis por classificar os dados, baseadas em evidências estatísticas. Caso a máquina identifique algum usuário com distúrbio motor, ela poderá notificar o profissional de saúde para que este visualize os dados e tenha um melhor suporte para a tomada de decisão em relação ao tratamento.

O funcionamento da abordagem pode ser descrito como uma composição de quatro passos: aquisição dos sinais por meio de sensores, processamento de sinais biomecânicos, classificação dos dados e visualização. Estes passos são detalhados nas seções seguintes.

## 1.3 Aquisição dos Sinais Por Meio de Sensores

O cliente JOGUE-ME é um jogo com funcionalidades de aquisição de dados motores de movimentos específicos. Logo, ele realiza a captura e o envio de dados para um servidor, que recebe requisições para efetuar o recebimento e o armazenamento das informações, o que torna possível armazenar o histórico do usuário para um acompanhamento dos sinais motores por um longo período. Com o uso do JOGUE-ME é possível adquirir os movimentos do paciente para identificar a evolução dos sinais do Doença de Parkinson (Parkinson), e consequentemente, quantificar sua saúde motora.

Atualmente, a análise dos sintomas motores é feita de forma subjetiva pelo cuidador ou esporádica pelo neurologista quando o paciente está em atendimento clínico, visto que, atualmente não existem mecanismos disponíveis em larga escala que permitam quantificar os sintomas motores ou acompanhar o tratamento a distância. Este projeto pretende atender a esta demanda e auxiliar a prática dos profissionais de saúde melhorando a qualidade de

vida dos pacientes com Parkinson.

Um sensor de movimentos como o MS-Kinect [11], por exemplo, possui uma câmera infravermelho capaz de reconhecer os movimentos de todo o corpo humano e identificar as posições das articulações anatômicas [7], para análise da cinemática do movimento humano [9].

Para mensurar os movimentos do paciente, utilizamos diretrizes médicas para avaliação motora do Parkinson como a UPDRS [5], a qual, permite diagnosticar e acompanhar o progresso da doença. Logo, os dados adquiridos pelos sensores podem ser mensurados com o processamento dos sinais e reconhecimento de padrões para identificar a ocorrência de sintomas motores.

Desta maneira, a aquisição dos sinais motores e a quantificação do movimento de seus usuários permitirá uma participação mais precisa do profissional de saúde.

## 1.4 Processamento de Dados Biomecânicos

O módulo de Processamento de Dados Biomecânicos é responsável por filtrar, remover ruídos e identificar ciclos de movimento para uma posterior extração dos vetores de características, como pode ser visto na Figura ?? . A partir dos sinais processados, aplicam-se técnicas de aprendizagem de máquina para obter a classificação dos sinais e, consequentemente, validar este trabalho.

### 1.4.1 Identificação de Ciclos de Movimento

Os sinais adquiridos por sensores de movimento possuem bastante ruído, o que dificulta a identificação dos ciclos de movimento, pois eles possuem uma posição que inicia o ciclo de movimento, como na Figura ?? , e o ruído existente pode cruzar por essa linha e consequentemente gerar falsas identificações.

Em casos de análise de sinais biomecânicos da amplitude do movimento, é possível aplicar a técnica de detecção de picos e vales do sinal [3]. Esta técnica consiste em usar um valor de referência,  $\delta$  (*delta*), para identificação dos picos, e descartar valores menores que são considerados ruídos. O pico é o ponto mais alto entre os 2 pontos mais baixos, que são considerados os vales do ciclo. A técnica é aplicada no sinal, com um  $\delta$  de 500, obtendo-se como resultado os picos e os vales identificados como pode ser visto na Figura ?? .

O processo de Identificação de Ciclos de Movimento é realizado em 3 etapas distintas:

- identificar ciclos de movimentos;
- calcular movimento angular realizado durante o ciclo de movimento;
- remover ciclos de movimentos incompletos.

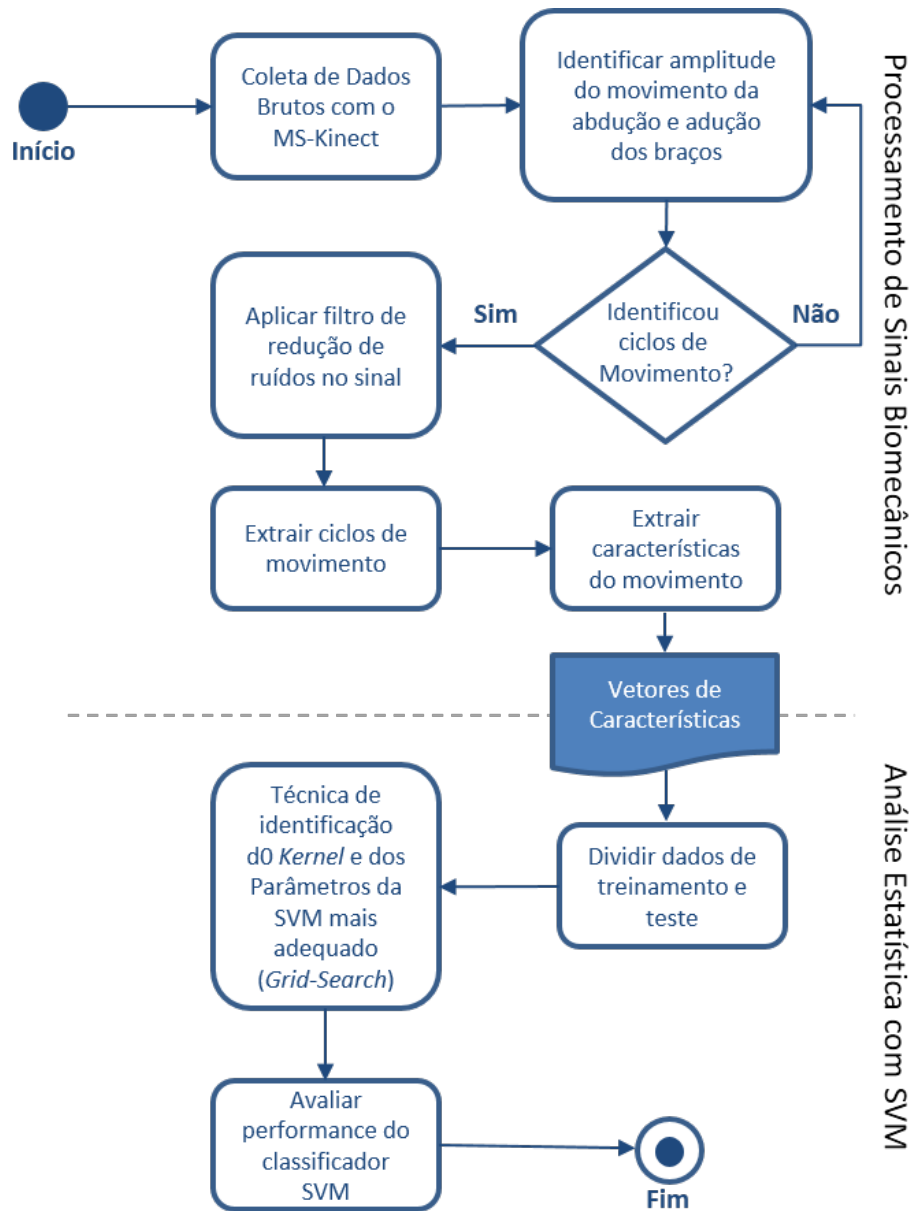


Figura 1.2: Processamento de sinais biomecânicos.

Para identificar os ciclos de movimento de adução e abdução dos braços, é necessário utilizar uma das articulações como referência. Neste movimento, a articulação do punho (Figura ??) é a que possui o sinal com maior amplitude entre as demais; por esse motivo, esta é a escolhida para identificar os ciclos. Realiza-se a técnica de picos e vales no sinal do *punho* para identificar o início e o fim do movimento de adução e abdução dos braços. Depois de identificado onde começa e termina o movimento, calcula-se o deslocamento angular através do produto escalar entre as articulações do punho, do ombro e da bacia (Seção ??). Neste momento, o sinal irá conter ciclos de movimentos angulares, então realiza-se uma nova eliminação de ruídos, ao extrair os ciclos de movimento identificados no sinal. Essa

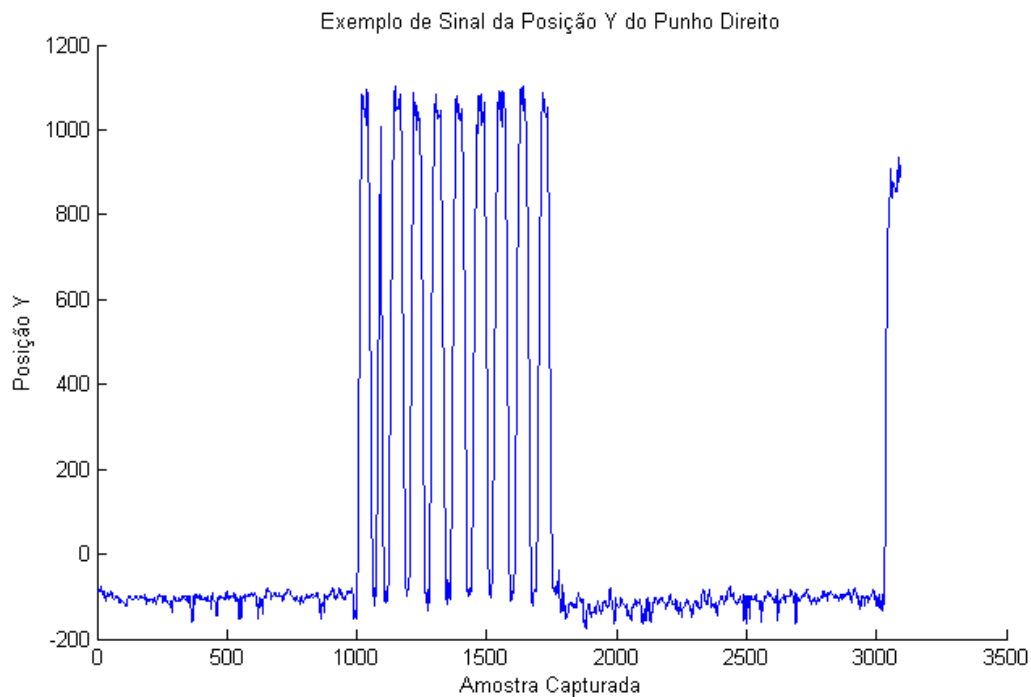


Figura 1.3: Exemplo de sinal capturado da articulação do punho direito usando MS-Kinnect na posição Y

é a primeira etapa da filtragem dos dados, a qual seleciona o início e o fim dos ciclos de movimentos. Depois desta etapa, realiza-se a extração de cada ciclo e identifica-se sua completude, para que as características extraídas dos ciclos de movimento sejam semelhantes para cada indivíduo e torne possível a classificação dos dados.

### 1.4.2 Extração das Características do Movimento

As características do sinal a ser obtido são baseadas na cinemática do movimento angular. Logo, é necessário um estudo da biomecânica do movimento humano nos ciclos de movimento [7]. De posse do tempo de ocorrência de cada ciclo e das articulações do **punho**, da **bacia** e do **ombro**, deve-se calcular o ângulo relativo do movimento de abdução e adução do braço através da aplicação do teorema do produto escalar, que encontra o ângulo entre dois vetores dentro do intervalo de  $0 \leq \theta \leq 180^\circ$ .

#### Cálculo do Ângulo Relativo do Movimento de Abdução e Adução

O produto escalar é uma operação entre dois vetores cujo resultado é um escalar [10]. Então, o ângulo entre dois vetores é definido como “o menor” ângulo entre eles. Dessa forma, este ângulo está dentro do intervalo de  $0 \leq \theta \leq 180^\circ$ . O produto escalar é o ângulo de  $\theta$  formado entre os vetores  $v$  e  $w$ .

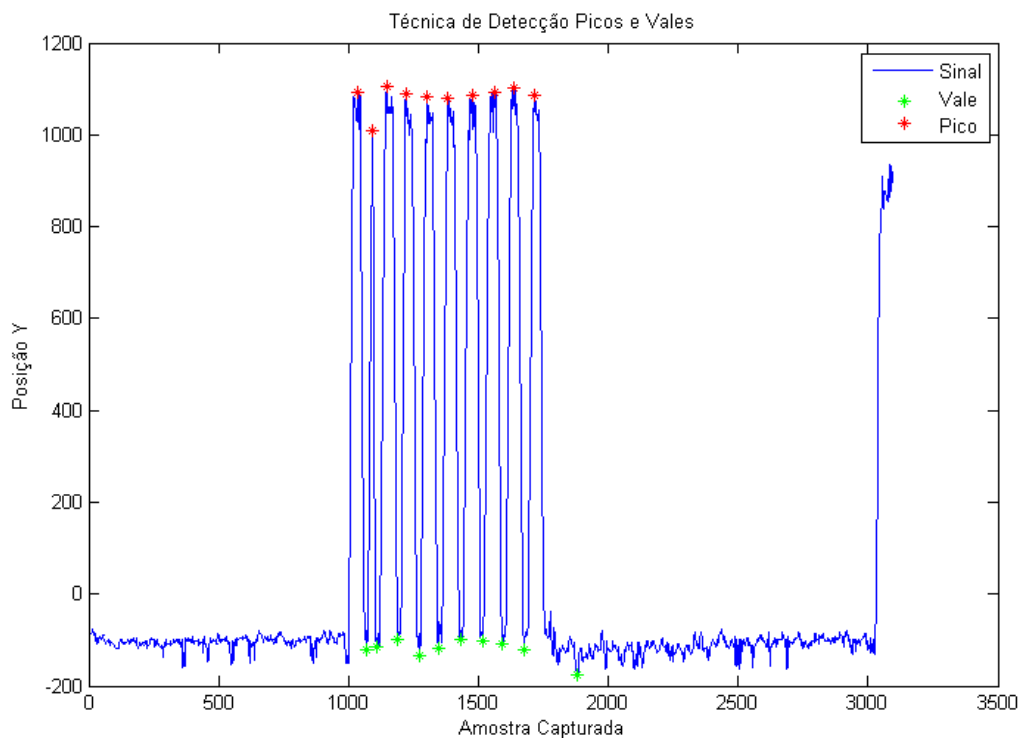


Figura 1.4: Exemplo da aplicação da técnica de detecção de picos e vales no sinal

No movimento de abdução e adução do braço, o ângulo relativo pode ser calculado com as posições ( $x$ ,  $y$ ,  $z$ ) das articulações (*quadril*, *ombro* e *punho*). Utilizando o produto escalar entre esses pontos, extraem-se as características do movimento, como amplitude do movimento, e, quando relacionamos com o tempo, conseguimos extrair a velocidade angular deste movimento, como pode ser visto na Figura ??, quantificando o movimento da adução e abdução do braço em relação ao tempo.

### Cálculo da Velocidade Angular do Movimento de Abdução e Adução

O pico da amplitude do movimento irá conter a amplitude máxima desse movimento. O tempo gasto entre o 1º vale e o pico em cada ciclo de movimento, será o tempo gasto para a abdução do braço, e o tempo gasto entre o pico e o 2º vale de cada ciclo, será o tempo gasto para a adução do braço. Portanto, com a amplitude máxima e o tempo gasto nesses movimentos, podem ser calculadas as velocidades angulares de abdução e adução dos braços, como pode ser visto na Figura ??.

### 1.4.3 Filtragem de Dados

A filtragem dos dados consiste na realização das seguintes etapas nos ciclos de movimento:

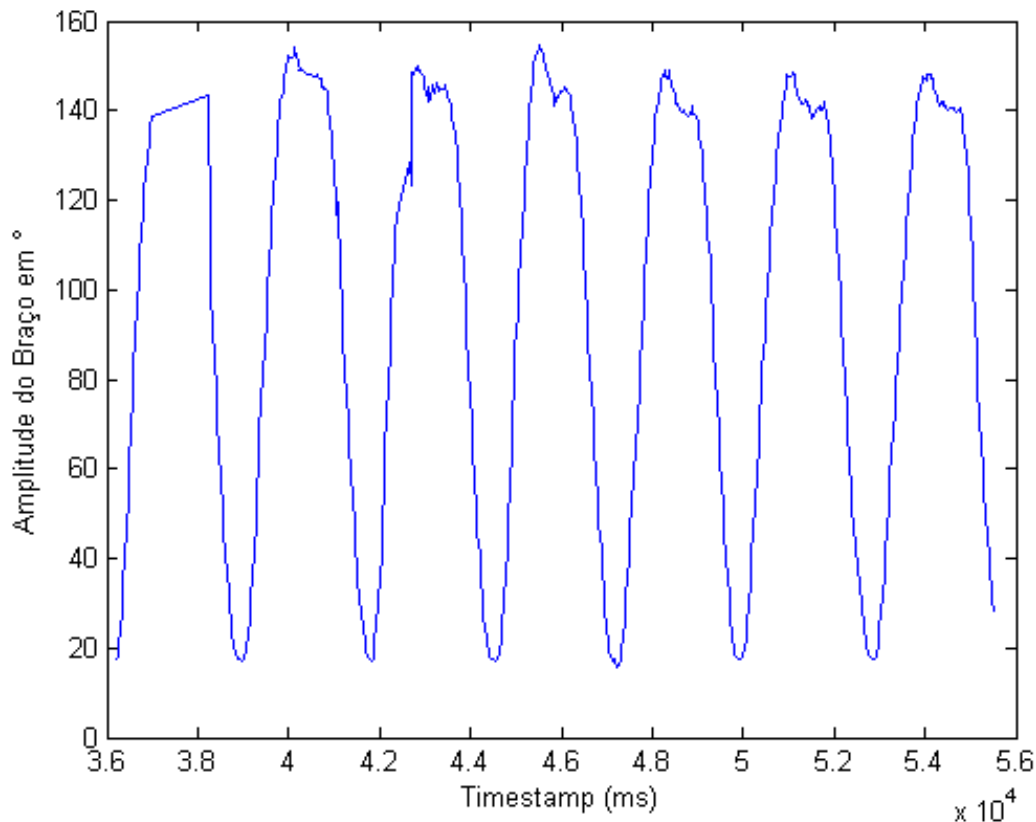


Figura 1.5: Amplitude do movimento de abdução e adução

**Escalonamento dos ciclos** : O conjunto de dados deve possuir a distribuição de  $M$  amostras de vetores de dimensão  $n$ . Como os dados a serem analisados são sinais, deve-se então escalonar o sinal para uma dimensão  $n$  para poder realizar o cálculo matricial quadrático de  $(M \times n)$ .

**Normalização dos ciclos** : Em estatística, o termo normalização possui diferentes significados [4]. Neste trabalho, a normalização consiste no ajuste dos valores dos dados em torno do valor máximo. Ou seja, o máximo valor obtido dos dados terá o valor 1, e os demais serão obtidos a partir da divisão do valor máximo. A normalização se faz necessária para que a variação dos dados seja mantida, além de facilitar a identificação de similaridades [18].

**Cálculo do Vetor Médio dos Ciclos** : Para definir a completude de um ciclo de movimento, deve-se calcular a média entre todos os ciclos de movimento, que é o vetor médio dos ciclos escalonados e normalizados. O **vetor médio**, Equação (??), chamado de  $\bar{X}$ , consiste na média aritmética de todos os ciclos de movimento, ou seja, ele define a



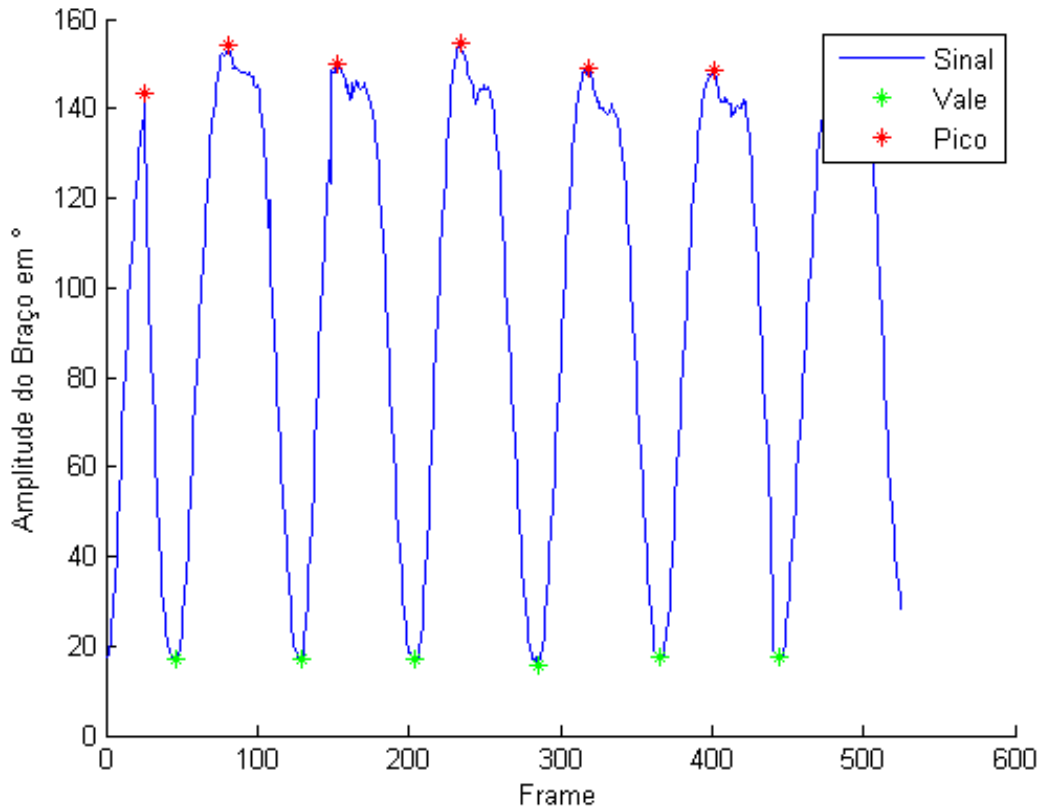


Figura 1.6: Detecção de picos e vales na amplitude do movimento de abdução e adução do braço

centralização dos dados [19].

$$\bar{X} = \frac{\sum_{i=1}^n (X_i)}{(n)} \quad (1.1)$$

**Calculo da Variância de Cada Ciclo ao Vetor Médio :** A variância é uma medida de dispersão estatística, que indica o quão longe os dados estão de um valor esperado [19]. Neste caso, o valor esperado é o vetor médio dos ciclos ( $\bar{X}$ ), e a variância, Equação (??), irá nos informar o quão distante cada ciclo ( $C$ ) está em relação a média.

$$var(C) = (C - \bar{X})^2 \quad (1.2)$$

**Definição do limiar para remoção de ciclos :** Essa etapa do processo de filtragem não é trivial, pois deve-se definir uma constante, *filtro* que será comparada à variância do ciclo. Se esta for menor, será aceita; caso contrário, removida. Contudo, balancear entre o limiar de dispersão do ciclo de movimento e a média é complexo, pois existe uma grande variabilidade de movimento. Logo, um limiar muito alto pode acarretar na remoção de uma grande quantidade de ciclos. Por outro lado, um limiar baixo colocaria ruídos nos dados e, conseqüentemente, impactaria no resultado da classificação.

Como exemplo, temos um ciclo de movimento removido (Figura ??), com *valor do filtro* = 1 e o *valor da variância* = 2,3078.

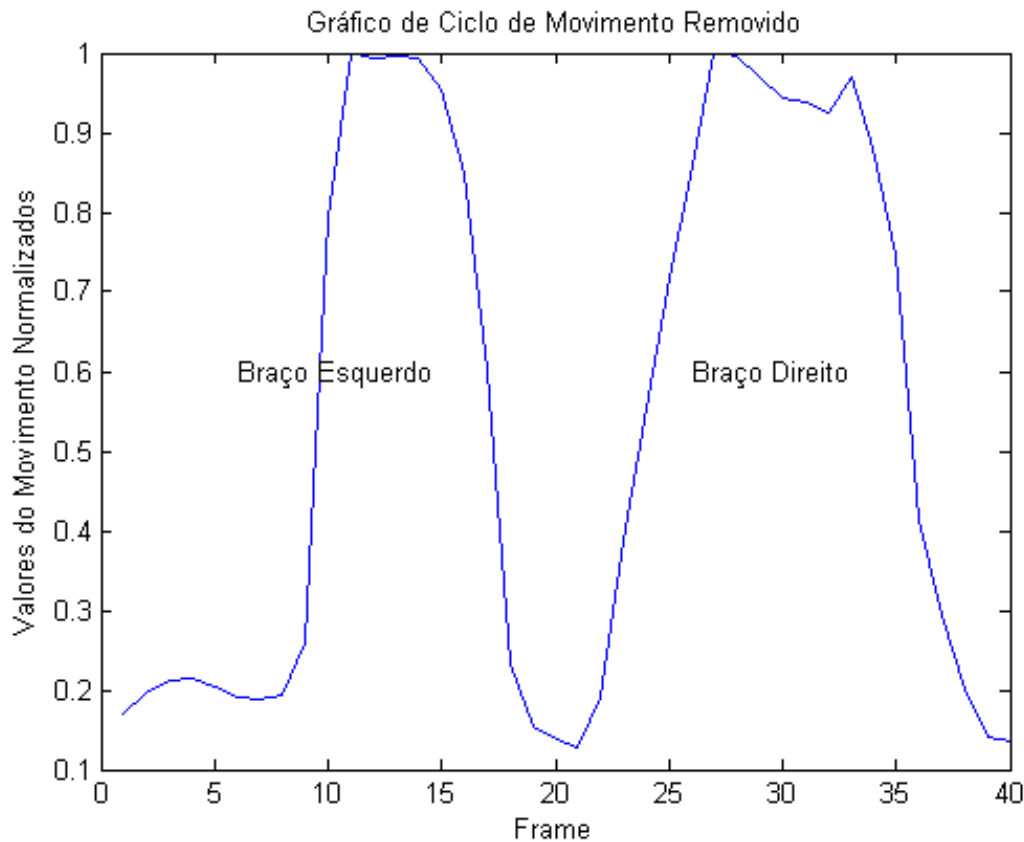


Figura 1.7: Ciclo de movimento removido

## 1.5 Classificação de Dados por Máquina de Aprendizagem

O objetivo de todo esse processo de identificação de ciclos, extração de características e filtragem é justamente facilitar a separação dos dados por máquinas de aprendizagem. A normalização dos ciclos ficou como o resultado do cálculo do produto escalar, que nos retorna valores entre 0 a 180 do movimento de abdução e adução. O escalonamento de cada ciclo de movimento ficou com 20 *frames*. Como temos o movimento do braço esquerdo e depois o do direito, temos um total de 40 *frames* por ciclo. O motivo pelo qual decidimos juntar os ciclos dos braços esquerdo e direito foi justamente para facilitar a identificação da assimetria do movimento existente nos estágios iniciais do Parkinson. Portanto, o classificador será responsável por identificar os indivíduos diagnosticados com Parkinson, por meio das diferenças de movimento existente entre estes e os indivíduos sem o diagnóstico da doença.

O vetor de características é composto dos ciclos de movimento e das características extraídas de cada ciclo, conforme explicado na Seção ???. Ou seja, terá, além do ciclo de movimento, os valores da velocidade angular de abdução e adução do braço esquerdo e direito. De posse desse vetor de características e do rótulo sobre a classe do ciclo de movimento (indivíduo diagnosticado com Parkinson e indivíduo sem o diagnóstico estabelecido), esses dados serão repassados como entrada-saída para o classificador de dados, que irá dividir entre grupos de treinamento e teste para realizar sua classificação.

Nesta abordagem, o classificador de dados será usado para identificar usuários com problemas motores. Dessa forma, irá auxiliar o profissional de saúde no acompanhamento de seus pacientes. Supondo que um profissional de saúde detém um grande número de pacientes, e que estes fazem uso da abordagem JOGUE-ME para monitorar seus dados, caso fosse identificada alguma anormalidade motora, o profissional de saúde seria notificado e poderia visualizar as informações que poderiam auxiliar na tomada de decisão.

## 1.6 Visualização dos Dados

O acompanhamento dos sinais motores é necessário, principalmente, para doenças crônicas de impacto motor e que tenham melhoria nos sinais, pois dessa maneira auxilia o médico no acompanhamento motor e, conseqüentemente, permite tratar o paciente de acordo com a resposta ao tratamento.

Como exemplo da abordagem, o profissional de saúde poderia visualizar as características dos movimentos, que serviram como dados de entrada para a máquina de aprendizagem. Nesse caso, podemos ver duas tabelas em que é possível identificar as diferenças motoras de uma pessoa diagnosticada com Parkinson (Tabela ??) e um indivíduo sem o diagnóstico da doença (Tabela ??).

Velocidades °/S				Amplitudes	
<b>Abdução Esquerda</b>	<b>Abdução Direita</b>	<b>Adução Esquerda</b>	<b>Adução Direita</b>	<b>Esquerda</b>	<b>Direita</b>
78,95	77,82	83,06	106,42	130,00	124,72
79,94	34,68	104,69	39,98	131,50	132,44
81,05	47,05	107,38	56,52	132,22	123,66
74,73	47,09	109,05	47,75	132,33	122,20
72,01	56,02	102,36	76,00	131,40	119,75

Tabela 1.1: Extração das características do indivíduo com diagnóstico do Parkinson

Como pode ser visto nesses dados, a amplitude de um indivíduo diagnosticado com Parkinson está bem menor do que em um indivíduo sem o diagnóstico estabelecido.

Velocidades °/S				Amplitudes	
<b>Abdução Esquerda</b>	<b>Abdução Direita</b>	<b>Adução Esquerda</b>	<b>Adução Direita</b>	<b>Esquerda</b>	<b>Amplitude</b>
129,35	61,59	78,74	176,30	159,39	143,50
115,67	118,15	71,72	79,46	156,37	153,97
120,96	135,27	66,70	78,17	154,30	149,91
125,96	137,43	64,75	81,57	153,18	154,58
139,99	117,60	69,96	84,08	151,68	148,90
120,51	111,92	75,85	75,18	152,58	148,35

Tabela 1.2: Extração das características do indivíduo sem diagnóstico do Parkinson

Um valor importante também pode ser identificado na velocidade de adução esquerda do indivíduo com Parkinson, pois este possui uma velocidade muito maior do que o indivíduo sem o diagnóstico. Possivelmente, porque um paciente com Parkinson perde um pouco o controle sobre o membro, fazendo-o descer abruptamente [13]. Dessa maneira, a abordagem pretende auxiliar o profissional de saúde com o fornecimento dessa informação, para que este efetue o acompanhamento e perceba a evolução do quadro clínico do paciente.

## 1.7 Conclusão

Neste capítulo, foram apresentados os requisitos que definiram a visão geral da abordagem JOGUE-ME. Uma seção relevante deste capítulo é como será o processamento dos dados biomecânicos, nesta seção é demonstrada como será a extração de características a partir dos sinais adquiridos por sensores de movimento. Posteriormente, foi apresentada a importância da aprendizagem de máquina neste trabalho para identificar os usuários que possuem problemas motores.

Do ponto de vista clínico, foram citados os parâmetros cinemáticos quantificados que irá auxiliar o profissional de saúde na sua tomada de decisão.

No capítulo seguinte, será apresentada a arquitetura de software que define a implementação da abordagem JOGUE-ME.

## Capítulo 2

### Arquitetura de Software do *JOGUE-ME*

Na concepção da arquitetura desenvolvida para implementar o JOGUE-ME (Figura ??), foram definidos 3 elementos básicos: sensor de movimentos para adquirir os sinais motores e enviar ao cliente; cliente JOGUE-ME, que é um jogo que interage com o usuário, induzindo a execução de movimentos para avaliação motora de uma maneira não-invasiva, e envia os dados a um serviço JOGUE-ME; serviço JOGUE-ME, que fica num estado de recebimento de dados e posterior repasse ao processador de dados biomecânicos, que processa e identifica a ocorrência dos sintomas motores por meio de uma máquina de aprendizagem.

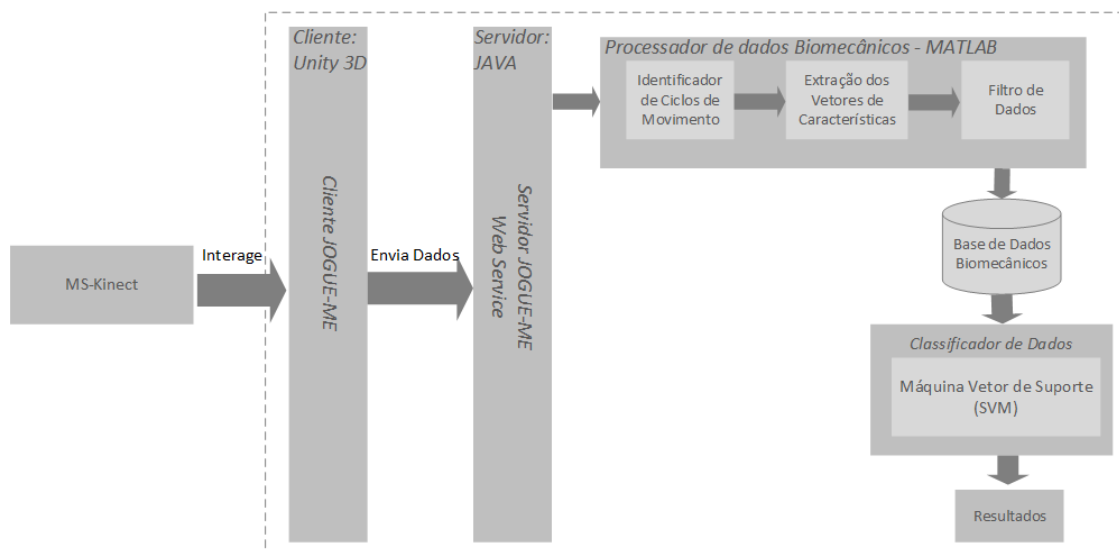


Figura 2.1: Arquitetura de Software

#### 2.1 Arquitetura do *JOGUE-ME*

A arquitetura do JOGUE-ME para o desenvolvimento de jogos utiliza uma *engine* de jogos (Unity3D [17]), que é um ambiente de desenvolvimento de jogos multi-plataforma. Esta

*engine* possibilita que os desenvolvedores abstraia-se dos aspectos de hardware, plataforma e complexidade do desenvolvimento de jogos e habilita o desenvolvedor a se ater somente às atividades referentes ao desenvolvimento do jogo.

Atualmente, desenvolvedores independentes de jogos utilizam Unity3D [17] como ferramenta de desenvolvimento. Esse ambiente facilita a criação de cenários, terrenos e interação com os objetos dos jogos usando uma linguagem de *script*. No entanto, desenvolver jogos com propósito de monitorar sinais motores possui desafios que não precisam ser de responsabilidade dos desenvolvedores de jogos. Por esse motivo, criamos uma arquitetura de software para o cliente JOGUE-ME (Figura ??) que abstrai a complexidade do desenvolvimento de um jogo de monitoramento da saúde.

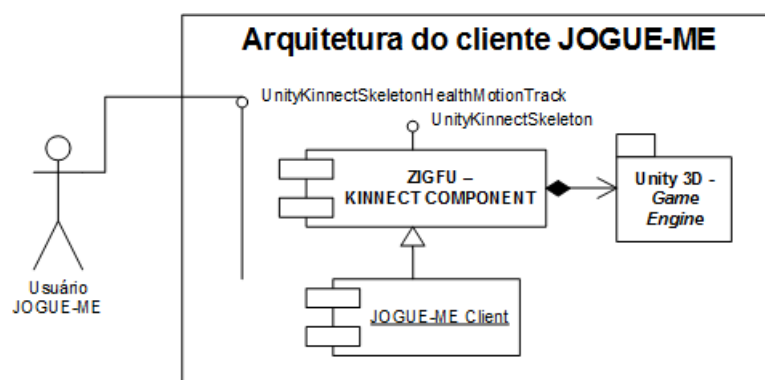


Figura 2.2: Arquitetura JOGUE-ME: módulo cliente de aquisição de sinais motores

Para possibilitar a aquisição de sinais motores, este trabalho herdou do componente (Zigfu [20]) para integrar o Ms-Kinnect [11] como controlador do jogo à *engine* de jogos Unity3D. O Ms-Kinnect [11] é um sensor de captura de movimentos utilizado tanto para o console MS-XBOX 360 quanto para *PCs*. Ele permite a aquisição dos sinais relativos ao movimento humano e identifica as articulações por meio da posição anatômica do corpo humano [7].

O Zigfu [20] é um componente de software que permite integrar o Ms-Kinnect ao Unity3D. O Zigfu faz um mapeamento das articulações adquiridas pelo Ms-Kinnect, para uma classe chamada *ZigSkeleton*, com todas as articulações, como podemos ver no Diagrama de Classe (Figura ??). No entanto, para adquirir os sinais motores, é necessário armazenar os valores das posições das articulações durante as ações dos usuários. Por esse motivo, o Zigfu foi estendido na classe *ZigSkeletonHealth* para armazenar as posições das articulações, além de um mecanismo para habilitar ou desabilitar o monitoramento dos sinais (métodos *logOn()* e *logOff()*).

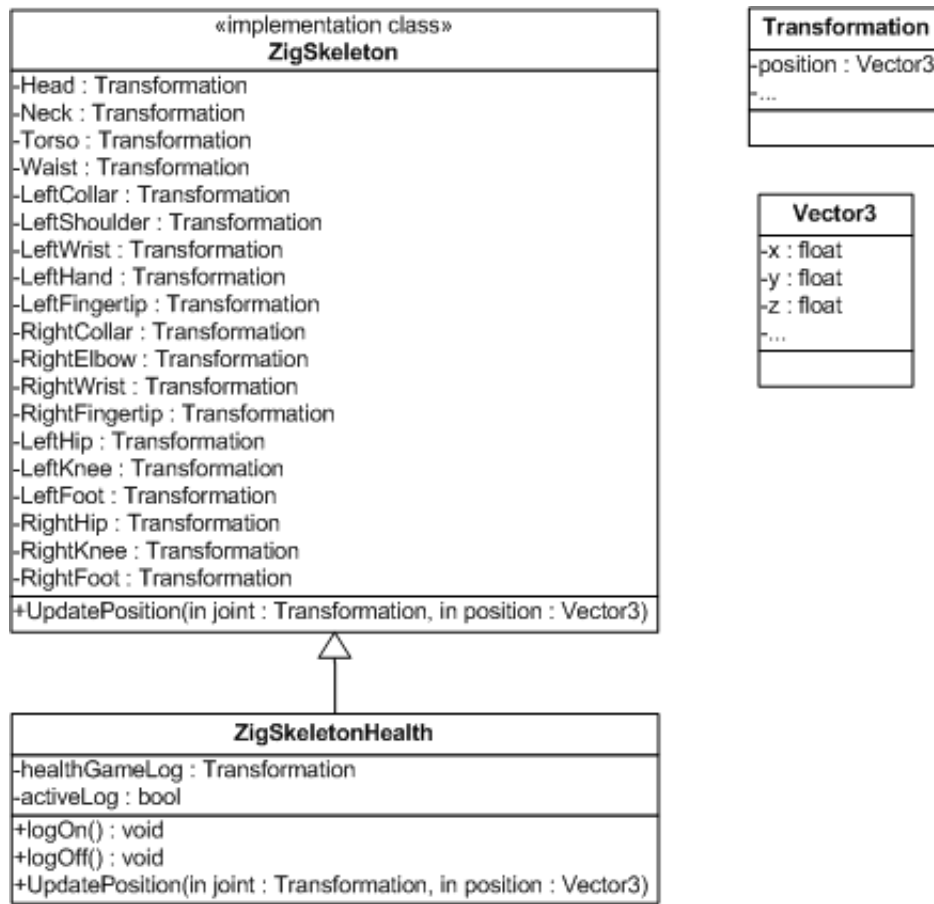


Figura 2.3: Diagrama de classe do ZigSkeleton e ZigSkeletonHealth

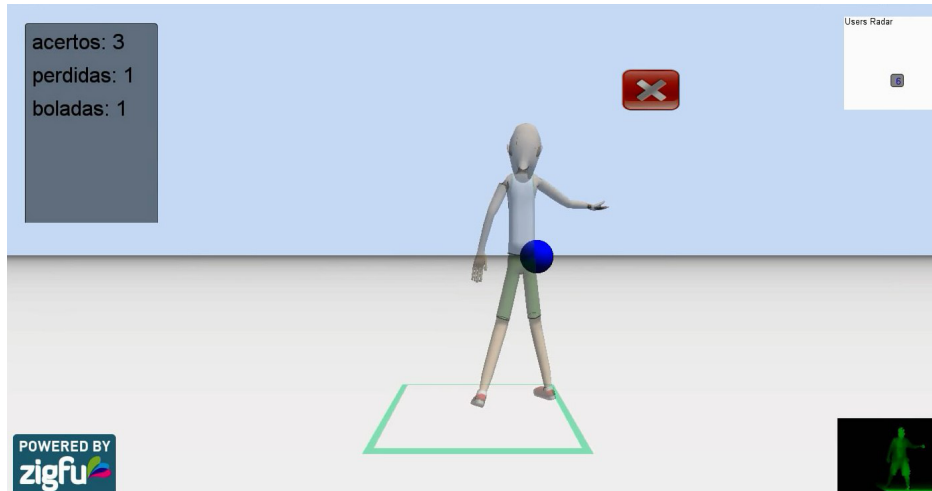
### Jogo: *Catch the Spheres*

Para testar a abordagem JOGUE-ME, foi criado o jogo *Catch the Spheres*, de acordo com os requisitos identificados na Seção ??.

O *Catch the Spheres* é um jogo em terceira pessoa, em que o jogador, por meio de sua personagem, deve tocar ou desviar das bolas que vêm em sua direção. Se o jogador tocar as bolas azuis, receberá uma pontuação por isso; caso seja atingido pelas bolas vermelhas, haverá uma penalização([REQ-JOGUE-ME-01]). Com o progresso do usuário, as bolas tornam-se mais rápidas, exigindo uma maior agilidade nos movimentos ([REQ-JOGUE-ME-02]). Este é o principal mecanismo de fluxo do jogo [16], que tem o intuito de atrair a atenção do jogador, baseado nos desafios propostos ([REQ-JOGUE-ME-03]).

Houve uma preocupação com a integridade física do jogador ([REQ-JOGUE-ME-04]). Por este motivo, baseado nos relatos dos usuários (Seção 1.3), o mecanismo de desvio de bolas foi removido por ter sido considerado inseguro.

O mecanismo de aquisição e armazenamento dos sinais motores ([REQ-JOGUE-ME-05]) torna possível o envio de sinais motores de maneira colaborativa, usando um serviço responsável por receber e armazenar esses sinais. Na abordagem JOGUE-ME, o servidor irá

Figura 2.4: O jogo *Catch the Spheres*

processar os sinais e transformá-los em informação para o profissional de saúde responsável pelo paciente.

### 2.1.1 Arquitetura do *JOGUE-ME Webservice*

O mecanismo de aquisição e armazenamento dos sinais motores ([REQ-JOGUE-ME-05]) torna possível a análise dos dados motores do usuário no qual o jogo armazena as informações e as envia para o servidor de dados.

O *JOGUE-ME Webservice* é responsável por: criar usuário, receber dados motores, gerenciar arquivos exportá-los para o MATLAB [8] conforme a arquitetura descrita do Diagrama de Classes (Figura ??).

O processo inicia com a aquisição dos dados dos sensores, que podem ser enviados para o *webservice* e processados pela classe `ReadingResource` ou enviados por arquivos e processados pela classe `FileManager`, acessada através do `DataManager`. O `ReadingResource` envia os dados recebidos para o `DatabaseManager`, também acessado através do `DataManager`, para armazená-los no *banco de dados* [12]. Na Tabela ??, ilustram-se as operações disponibilizadas pelo *webservice* e um exemplo de como os dados devem ser estruturados para cada operação.

O envio dos dados dos usuários coletados com os dispositivos é feito através de uma requisição POST para o *web service*. Os dados devem ser coletados durante uma sessão completa do jogo, que dura de alguns segundos a alguns minutos, para depois serem estruturados e enviados para o *webservice*. O formato aceito pelas operações é o JSON (JavaScript Object Notation).



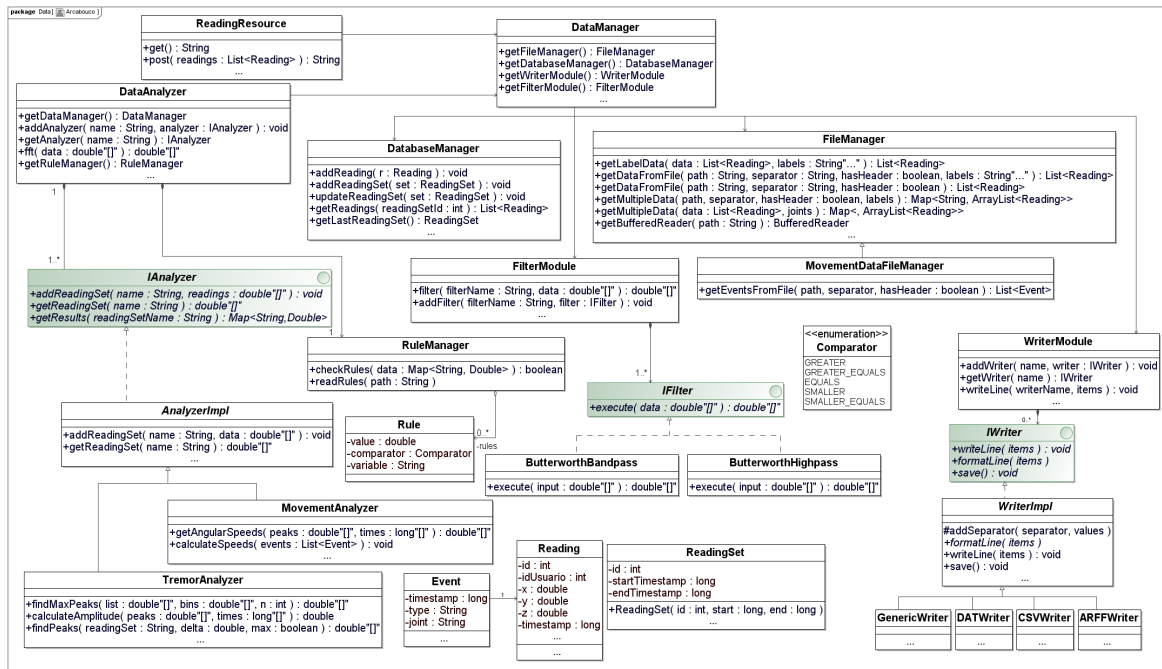


Figura 2.5: Diagrama de Classes do do Serviço JOGUE-ME

## Gerenciador de Dados

O *Gerenciador de Dados* possui submódulos responsáveis por ler, separa e filtrar os dados, além do gerenciá-los usando um Banco de Dados para o armazenamento das informações motoras. A classe `DataManager` implementa as funcionalidades do *Gerenciador de Dados*, referenciando os quatro módulos: *Gerenciador de Arquivos*, *Módulo de Escrita*, *Módulo de Filtragem* e *Gerenciador do Banco de Dados*. Estes módulos serão explicados nas subseções a seguir. A classe `DataManager` possui um construtor `DataManager(DatabaseManager, FileManager, WriterModule, FilterModule)`, que recebe como parâmetros os quatro módulos. Dessa forma, é possível aumentar a funcionalidade de cada um dos módulos estendendo suas respectivas classes por herança e adicionando a elas novos métodos. A classe `MovementDataFileManager`, tratada mais adiante, é um exemplo de extensão do `FileManager`.

O *webservice*, implementado utilizou a biblioteca Jersey<sup>1</sup>, que facilita o desenvolvimento de *RESTful webservices*. As requisições são enviadas para serem processadas pela classe `ReadingResource`, que é um *web resource*, uma entidade que recebe requisições HTTP e envia respostas. Esta classe possui dois métodos, o `get()` que trata requisições *GET*, retornando o identificador do último conjunto de leituras para controle do armazenamento no banco de dados; e o método `post(List<Reading> readings)` processa os dados das

<sup>1</sup>Disponível em: <http://jersey.java.net/>

Tabela 2.1: Operações disponibilizadas pelo *web service*

Operação	Método	Exemplo
cadastrarUsuario	POST	<pre>{ "id":2, "nome":"Ana",   "masculino":false,   "nascimento":"2012-11-28" }</pre>
obterToken	GET	-
enviarDados	POST	<pre>{ "leitura":[{"id":0,   "idUsuario":1, "x":2.9097333,   "y":6.770132, "z":2.0355952,   "timestamp":1336134935706},   {"id":0, "idUsuario":1,   "x":4.5565815, "y":4.9461093,   "z":1.4911331,   "timestamp":1336134935706}] }</pre>

leituras enviados através de requisições *POST*, e convertidos de JSON para objetos Java pela biblioteca Jersey. A classe `ReadingResource` está acoplada à classe `DataManager` e, através dela, tem acesso ao *Gerenciador do Banco de Dados*. O *webservice* pode ser instalado em qualquer *web container*, como o Apache Tomcat<sup>2</sup> e o GlassFish<sup>3</sup>.

### Gerenciador de Arquivos

A classe `FileManager` implementa o módulo *Gerenciador de Arquivos*, que processa as operações de abertura de arquivos de dados delegadas pelo *Gerenciador de Dados*. Esse módulo processa os dados recebidos, armazenando-os em dados estruturados para serem processado posteriormente pelo *Analizador de Dados*. O dado estruturado aceito pelo *Analizador de Dados* é composto por um rótulo identificador do dado, uma marca de tempo com precisão de milissegundos, e coordenadas x, y e z, cujo significado depende do tipo de sensor que as gera.

Os métodos da classe `FileManager` são:

1. `getLabelData(List<Reading> data, String... labels)` filtra os dados da lista de leituras `data`, retornando uma nova lista `List<Reading>` contendo apenas os dados com os rótulos definidos em `labels`.

<sup>2</sup>Disponível em: <http://tomcat.apache.org/>

<sup>3</sup>Disponível em: <http://glassfish.java.net/>

2. `getDataFromFile(String path, String separator, boolean hasHeader)` lê os dados de um arquivo localizado no caminho `path`, cujos dados estão separados pelo separador `separator` e definidos linha a linha. O parâmetro `hasHeader` indica se o método deve procurar por uma linha de cabeçalho na primeira linha do arquivo. Retorna uma `List<Reading>` com os dados.
3. `getDataFromFile(String path, String separator, boolean hasHeader, String... labels)` estende a funcionalidade do método anterior, retornando uma `List<Reading>` com os dados que possuem os rótulos definidos em `labels`.
4. `getMultipleData(String path, String separator, boolean hasHeader, String... labels)` possui a mesma função que o método ??, mas, diferente deste, retorna um `Map<String, List<Reading>` onde cada chave do mapa é um rótulo e indexa uma lista de eventos identificados pelo rótulo.
5. `getBufferedReader(String path)` retorna um `BufferedReader` para manipular o arquivo cujo caminho é especificado em `path`.

A classe `MovementDataFileManager` estende as funcionalidades do `FileManager`, adicionando um método para leitura de eventos oriundos de jogos. Os eventos marcam o início ou fim de um momento específico do jogo no qual o jogador estará executando um movimento que será enviado para análise.

### 2.1.2 Módulo de Escrita

O *Módulo de Escrita* é implementado pela classe `WriterModule`, que é responsável pela saída dos dados processados pelo *Analizador de Dados*. Os dados podem ser estruturados para serem mostrados em um programa de plotagem de gráficos, como o `GNUPlot`<sup>4</sup>, ou para servirem como entrada para mecanismos de aprendizado de máquina. Os dados são escritos em CSV (*Comma-separated Values*) ou em qualquer outro formato definido pelo usuário do arcabouço. O módulo de escrita também suporta a escrita de arquivos ARFF, para serem processados pelo `Weka`<sup>5</sup>. O *Módulo de Escrita* é extensível para permitir a geração de um formato de arquivo específico. A criação de um novo arquivo de dados é feita através da extensão da classe `WriterImpl` pela classe que se está criando.

A interface `IWriter` define três métodos para manipular arquivos de dados:

1. `formatLine(Object... items)` formata os itens `items` adicionando separadores ou qualquer outra formatação adicional definida na classe específica de escrita que implementa `IWriter` ou estende `WriterImpl`.

---

<sup>4</sup>Disponível em: <http://www.gnuplot.info/>

<sup>5</sup>Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>

2. `writeLine(Object... items)` escreve uma nova linha no arquivo, seguindo a formatação definida pelo método ??.
3. `save()` fecha a *stream* de escrita dedicada ao arquivo e salva o arquivo em disco.

A classe `WriterImpl` implementa os métodos comuns a todas as classes de escrita, definidos pela interface `IWriter`, fornecendo um método adicional para incluir separadores entre os elementos de uma linha. Para definir um comportamento diferente daquele implementado por `WriterImpl`, deve-se implementar diretamente a interface `IWriter`.

## 2.2 Processador de Dados Biomecânicos

Para transformar os sinais em informação, tanto para o profissional de saúde, quanto para máquinas de aprendizagem, é necessário fazer o processamento desse sinal. Neste trabalho, foi implementado o *Processador de Dados Biomecânicos* em MATLAB 2011 [8]. Este processador consiste de três passos: Identificação dos Ciclos, Extração de Características e Filtragem de Dados.

### 2.2.1 Identificação dos Ciclos de Movimento

A identificação dos ciclos de movimento foi baseada na identificação de picos e vales do sinal motor, como explicado na Seção ??.

Para implementar o mecanismo de detecção de ciclos, fez-se o uso da biblioteca *Peak Detection in Matlab* [3]. Essa biblioteca possui uma função chamada `peakdet()`, que recebe como parâmetros um vetor contendo o sinal a ser processado e um valor de limiar para remoção do ruído do sinal. A função retorna dois vetores: um possui os valores das máximas (picos) e o outro retorna os valores das mínimas (vales).

Usando a função `peakdet()`, criou-se a função `cycleperiodic()`, que tem o objetivo de identificar os ciclos periódicos de um sinal. Foram adicionados dois parâmetros a essa função, para justamente levar em consideração as amplitudes máximas e mínimas permitidas por este sinal.

De posse dos ciclos, pôde ser identificado quando começam e terminam os movimentos periódicos (Código Fonte ??), como, por exemplo, os movimentos sucessivos de adução e abdução do braço (Seção ??).

Código Fonte 2.1: Identificar início e término do movimento periódico

```

1 function [WindowBeginLeft, WindowLengthLeft, WindowBeginRight,
   WindowLengthRight] = identifyCycles(leftWristJoint, rightWristJoint)
2     signalLeft = leftWristJoint(:,3);
3     signalRight = rightWristJoint(:,3);
4 
```

```

5     cycleIndexLeft = cycleperiodic(signalLeft , 500, 200, 40);
6     cycleIndexRight = cycleperiodic(signalRight , 500, 200, 40);
7
8     WindowBeginLeft = cycleIndexLeft(1);
9     WindowLengthLeft = cycleIndexLeft(size(cycleIndexLeft ,2));
10    WindowBeginRight = cycleIndexRight(1);
11    WindowLengthRight = cycleIndexRight(size(cycleIndexRight ,2));

```

## 2.2.2 Extração das Características do Movimento

Supondo que os ciclos de movimento foram identificados através da posição do punho, é necessário extrair as características do movimento. Para isso, o primeiro passo é calcular os ângulos relativos do movimento angular, usando os pontos das articulações, como pode ser visto no Código Fonte ???. Então, a função *ArmRelativeAngleTorso()* realiza o cálculo do produto escalar entre as três articulações.

Código Fonte 2.2: Calcular ângulos relativos do movimento

```

1 leftShoulderJoint = leftShoulderJoint(WindowBeginLeft:WindowLengthLeft ,:)
   ;
2 leftWristJoint = leftWristJoint(WindowBeginLeft:WindowLengthLeft ,:) ;
3 leftHipJoint = leftHipJoint(WindowBeginLeft:WindowLengthLeft ,:) ;
4
5 for (j=1:size(leftHipJoint ,1))
6 leftArmAngle(j ,1) = leftHipJoint(j ,1);
7
8
9 leftArmAngle(j ,2) = ArmRelativeAngleTorso(leftHipJoint ,leftShoulderJoint ,
   leftWristJoint , j);
10 end

```

De posse do sinal dos ângulos relativos do movimento, são extraídos os picos e os vales desse sinal para calcularmos a velocidade angular do movimento de abdução e adução do braço (Código Fonte: ???).

Código Fonte 2.3: Calcular velocidade angular da adução e abdução

```

1 distanceup = cycle(peak) - cycle(1);
2 amplitude(identifiedCycles ,1) = cycle(peak);
3
4 timestampupsec = (abs(timestampcycle(1) - timestampcycle(peak)))/1000;
5 velocityUp(identifiedCycles ,1) = distanceup /timestampupsec;
6
7 distancedown = abs(cycle(end) - cycle(peak));
8 timestampdownsec = (abs(timestampcycle(peak) - timestampcycle(end)))
   /1000;

```

```
9 velocityDown(identifiedCycles,1) = distancedown/timestampdownsec;
```

### 2.2.3 Filtro de Dados

O filtro de dados remove os ciclos de movimento incompletos ou com problemas na aquisição dos dados, como explicado na Seção ???. Nessa etapa, os ciclos são normalizados, escalonados e rotulados por usuário. De posse de todos os dados, é calculado um vetor médio dos ciclos normalizados, para definir um limiar (*threshold*) de remoção dos ciclos (Código Fonte: ??).

Código Fonte 2.4: Filtro para remoção de ciclos

```
1 function [ KinectData , processedCycles , labels ] = filterCyclesAndLabels
   (T, labels , otherFeatures , scaledLength)
2
3     normalization = T;
4     for i=1:size(T,1)
5         normalization(i,1:scaledLength) = T(i,1:scaledLength) ./ max(T(i,1:
           scaledLength));
6         normalization(i,scaledLength+1:scaledLength*2) = T(i,scaledLength
           +1:scaledLength*2) ./ max(T(i,scaledLength+1:scaledLength*2));
7         normalization(isnan(normalization(i,1:scaledLength*2))) = min(
           normalization(i,1:scaledLength*2));
8     end
9
10    normalization(isnan(normalization)) = 0;
11
12    if(size(T,2) > scaledLength*2)
13        normalization(:,scaledLength*2 + 1:end) = T(:,scaledLength*2+1 :
           end) ./ max(T(:,scaledLength*2 + 1:end));
14    end
15
16    threshold = 1;
17    meanOfNormalization = mean(normalization);
18    u = ones(size(normalization,1),1);
19    filterTestVector = sum((normalization - (u*meanOfNormalization))
           .^2,2);
20    filterVector = filterTestVector < threshold;
21
22
23    KinectData = [T(filterVector,:) otherFeatures(filterVector,:)];
24    processedCycles = T(filterVector,:);
25    labels = labels(filterVector,:);
26 end
```

## 2.3 Classificador de Dados

Para avaliar os requisitos de identificação dos sinais motores ([REQ-JOGUE-ME-06]), é necessário um teste com seres humanos, para avaliar a aquisição e a classificação dos sinais. A abordagem de classificação dos dados é baseada em máquinas de aprendizagem, como explicado na Seção ???. O Código Fonte ?? demonstra como fazer a classificação dos dados utilizando o *Matlab Statistics Toolbox* [8], que possui um Máquina de Vetor de Suporte (SVM) disponível em sua biblioteca.

Primeiramente, separa-se o grupo de treinamento para realizar a aprendizagem da máquina, utilizando o método *svmtrain()*; depois utiliza-se o método *svmclassify()* para predizer os valores usando esta máquina de aprendizagem; por fim, calculam-se as diferenças entre os valores reais. Então, é calculada a taxa de erro para avaliar o resultado do classificador.

Código Fonte 2.5: Uso da SVM para classificação dos dados

```
1 realValues; %Classe Atual
2 SVMStruct = svmtrain(trainingData ,trainningClassification , '
    Kernel_Function ', 'linear ', 'BoxConstraint ', 0.10);
3 class = svmclassify(SVMStruct, testData , 'showplot ', true); %Classe
    Preditiva
4 classificationRate = sum(class~=realValues);
5 errorRate = classificationRate / size(classreal ,2);
```

## 2.4 Conclusão

Neste capítulo foram apresentados os passos da implementação de um JOGUE-ME para o monitoramento do sinal da bradicinesia presente no Parkinson.

Foi apresentado o design da arquitetura tanto do cliente quanto do serviço de recebimento de requisições, processamento do sinal e identificação do sintoma da bradicinesia [13] presente no Parkinson. Demonstrou-se neste capítulo detalhes da implementação para que este trabalho possa ser replicado para outras análises motoras.

No capítulo seguinte, é apresentado os experimentos realizados para avaliar este trabalho referentes a: perspectiva do profissional de saúde e suas necessidades, resultado da identificação do sintoma da bradicinesia no estudo analítico de caso-controle e a avaliação dos pacientes com Parkinson quanto a abordagem de monitoramento dos sinais motores usando jogos eletrônicos.

# Bibliografia

- [1] Dias-Tosta, Elza, Carlos Roberto de Mello Rieder, Vanderci Borges e Ylmar Correa Neto: *Doença de Parkinson - Recomendações da Academia Brasileira de Neurologia*. 1ª edição, 2010, ISBN 978-85-62477-05-8.
- [2] Do, Stewart e William Weiner: *Parkinson's Disease: Diagnosis Clinical Management*. Demos Medical Publishing, 2ª edição, 2007, ISBN 193-386-400-1.
- [3] Fish, Jessica: *Unified Parkinson's Disease Rating Scale*. Springer, 2011, ISBN 978-0-387-79948-3.
- [4] Flick, U.: *Uma Introdução à Pesquisa Qualitativa*. Bookman, 2ª edição, 2004, ISBN 978-853630-414-4.
- [5] Hamill, Joseph e Kathleen Knutzen: *Bases Biomecânicas do Movimento Humano*. Manole, 3ª edição, 2012, ISBN 978-852042-356-1.
- [6] Health, National Institute for e Clinical Excellence: *Parkinson's Disease: Diagnosis and Management in Primary and Secondary Care*. NICE Clinical Guideline, 1ª edição, 2006, ISBN 1-86016-283-5.
- [7] Kantardzic, Mehmed: *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons, 3ª edição, 2011, ISBN 978-111802-913-8.
- [8] Keijsers, Noël, Martin Horstink e Stan Gielen: *Ambulatory Motor Assessment In Parkinson's Disease*. Movement Disorders, 21, 2006, ISSN 1531-8257.
- [9] LeMoyne, Robert, Timothy Mastroianni, Michael Cozza, Cristian Coroian e Warren Grundfest: *Implementation of an iPhone for characterizing Parkinson's disease tremor through a wireless accelerometer application*. Em *International Conference of the IEEE Engineering in Medicine and Biology*, 2010, ISBN 978-142447-927-6.
- [10] Li, Cheng Hsua, Chin Teng Lin, Bor Chen Kuo e Hsin Hua Ho: *An Automatic Method for Selecting the Parameter of the Normalized Kernel Function to Support Vector Machines*. Em *International Conference on Technologies and Applications of Artificial Intelligence*, 2010, ISBN 978-3-319-13986-9.



- [11] McGinnis, Peter: *Biomechanics of Sport and Exercise*. Human Kinetics, 3ª edição, 2013, ISBN 978-073607-966-2.
- [12] Microsoft: *Ms-Kinect*. Disponível em: <http://www.xbox.com/pt-BR/kinect>, 2012. Último acesso: 3 de Junho de 2016.
- [13] Nuseibeh, Bashar e Steve Easterbrook: *Requirements Engineering: A Roadmap*. Em *Proceedings of the Conference on The Future of Software Engineering*. ACM, 2000, ISBN 1-58113-253-0.
- [14] Oliveira Bastos Junior, Paulo Roberto de: *Elicitação de Requisitos de Software Através da Utilização de Questionários*. Dissertação de Mestrado, PUC-Rio, 2005.
- [15] Patel, Shyamal, Konrad Lorincz, Richard Hughes, Nancy Huggins, John Growdon, David Standaert, Metin Akay, Jennifer Dy, Matt Welsh e Paolo Bonato: *Monitoring Motor Fluctuations in Patients With Parkinson's Disease Using Wearable Sensors*. IEEE Transactions on Information Technology in Biomedicine, 13(6), 2009, ISSN 1089-7771.
- [16] Research, Provalis: *QDA MINER Lite v1.2.2. Software*, Disponível em: <http://provalisresearch.com/products/qualitative-data-analysis-software/>. Último acesso: 3 de Junho de 2016.
- [17] Sano, Yuko, Akihiko Kandori, Toshinori Miyoshi, Toshio Tsuji, Keisuke Shima, Masaru Yokoe e Saburo Sakoda: *Severity Estimation Of Finger-tapping Caused By Parkinson's Disease By Using Linear Discriminant Regression Analysis*. Em *International Conference of Engineering in Medicine and Biology Society*. EMBC, 2012, ISBN 9781424479276.
- [18] Saraiva, Alessandra Vilches: *Utilização da Abordagem Goal-Question-Metrics (GQM) Na Elaboração e Execução de Planos de Avaliação de Usabilidade de Software: Um Estudo Empírico Sobre Um Software Agropecuário*. Dissertação de Mestrado, Universidade Metodista De Piracicaba, 2006.
- [19] Saúde, Ministério da: *Protocolo Clínico E Diretrizes Terapêuticas - Doença De Parkinson - Portaria SAS/MS Nº 228*. Relatório Técnico, Sistema Único de Saúde, 2010.
- [20] Solingen, Rini e Egon Berghout: *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill, 1999, ISBN 978-007709-553-6.
- [21] Sommerville, Ian: *Engenharia de Software*. Addison Wesley Bra, 2011, ISBN 978-858863-928-7.

- 
- [22] Tao, Weijun, Tao Liu, Rencheng Zheng e Hutian Feng: *Gait Analysis Using Wearable Sensors*. Sensors, 12(12), 2012, ISSN 1424-8220.
- [23] Tolosa, Eduardo, Gregor Wenning e Werner Poewe: *The Diagnosis Of Parkinson's Disease*. The Lancet Neurology, 5(1), 2006, ISSN 1474-4422.
- [24] Zowghi, Didar e Chad Coulin: *Requirements Elicitation: A Survey of Techniques, Approaches, and Tools*. Em *Engineering and Managing Software Requirements*. Springer Berlin Heidelberg, 2005, ISBN 978-3-540-28244-0.